Modern Applications of Game-Theoretic Principles

Catuscia Palamidessi, Marco Romanelli

Modern Applications of Game-Theoretic Principles

Catuscia Palamidessi
Inria, France
LIX, Ecole Polytechnique, Institut Polytechnique de Paris, France
https://www.lix.polytechnique.fr/~catuscia/

Marco Romanelli
Inria, France
LIX, Ecole Polytechnique, Institut Polytechnique de Paris, France
University of Siena, Italy
http://www.lix.polytechnique.fr/Labo/Marco.Romanelli/

Abstract

Game theory is the study of the strategic behavior of rational decision makers who are aware that their decisions affect one another. Its simple but universal principles have found applications in the most diverse disciplines, including economics, social sciences, evolutionary biology, as well as logic, system science and computer science. Despite its long-standing tradition and its many advances, game theory is still a young and developing science. In this paper, we describe some recent and exciting applications in the fields of machine learning and privacy.

2012 ACM Subject Classification Security and privacy → Formal security models; Security and privacy → Privacy-preserving protocols; Security and privacy → Information flow control

Keywords and phrases Game theory, machine learning, privacy, security

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2020.4

Category Invited Talk

Funding This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme. Grant agreement № 835294.

1 Introduction

Game theory is concerned with situations in which one agent's best action depends on the expectation about what one or more other agents will do, and viceversa. Long before being investigated as a mathematical discipline, game-theoretic insights go back to ancient times, and influenced the strategies of political and military leaders. For example, when landing in Mexico with a small force, and surrounded by a far more numerous Atzecs army, Cortez burnt the ships on which he and his soldiers had landed, and took care of doing so very visibly, so that the Atzecs would see it. His action had a discouraging effect on the Aztecs, who must have thought: Any commander who is so confident as to willfully destroy his own option to save himself must have good reasons to be so optimist; it wouldn't be wise to attack an opponent who is so sure (whatever, exactly, the reason might be) that he can't lose. Thus the Atzecs retreated and missed the occasion – perhaps the best they ever had, to stop the Spanish invasion.

The initial formulation of games in mathematical terms is usually attributed to Emile Borel. However, it was John von Neumann who first solved the problem of the two-player zero-sum games, with the minimax theorem [14]. Formally, this theorem states that, if $f : X \times Y \to \mathbb{R}$ is a continuous function that is concave in the first argument and convex in

the second one, then we have that:

$$\max_{x \in X} \min_{y \in Y} f(x, y) = \min_{y \in Y} \max_{x \in X} f(x, y).$$

The importance of this theorem consists in the fact that it states the existence of a *saddle point*, which in a sense represents the best possible pair of choices of the two adversaries – having to take each other strategy into account.

Von Neumann's work culminated in the fundamental book on game theory titled *Theory of Games and Economic Behavior*, written in collaboration with Oskar Morgenstern [20]. The mathematical foundations established in this book, and later on in the work of Nash, who generalized the notion of saddle point with that of Nash equilibrium [10], found a wide range of applications, especially in economics, social sciences, and biology.

In the past several decades, the framework has been deepened and generalized, and new variants and refinements are still being made. In logic, the principles of game theory have inspired the *game semantics*, with the purpose of giving an interpretation to intuitionistic logic [13] and to linear logic [5]. These ideas were further developed by Samson Abramsky, Radhakrishnan Jagadeesan, Pasquale Malacaria and independently Martin Hyland and Luke Ong, who focused in particular on defining strategies compositionally and inductively on the syntax. In this way, they succeeded to define a fully abstract semantics for PCF, which had been a long-standing problem [3, 9]. Following this line, game semantics has been applied to a variety of programming languages, and has led to semantic-based methods of software verification by model checking [2],

In concurrency theory, games were used for the first time by Colin Stirling to give an elegant characterization of *strong bisimulation* [19]. These are infinite-duration games played by two adversaries, called *Spoiler* and *Duplicator*. The game is played on pairs of states (s, t) in a labelled transition system, where Duplicator tries to prove that states s and t are strongly bisimilar, whereas Spoiler tries to disprove this. Spoiler plays first, and challenges Duplicator with a move from either s or t, while Duplicator is required to match this move. At the next turn, the game continues from the pair of the new states reached after the transitions. If the game is infinite (i.e., the play continues forever), Duplicator wins the game. If on the other hand the game is finite, namely one of the players gets stuck, then the game is won by the other player. Note that if it is Duplicator that gets stuck, it means that Spoiler found a way to distinguish the states. Viceversa, if it is Spoiler that gets stuck, or if the games goes on indefinitely, then it means that Duplicator is able to match all Spoiler's challenges. Hence, the two states are deemed equivalent if and only if Duplicator wins.

In recent times, game theory has found additional exciting applications in new fields: machine learning (ML), privacy, and computer security. We are going to give an overview of the first two applications in the next sections.

2 Game theory and machine learning

The principles of game theory have inspired Ian Goodfellow and other researchers at the University of Montreal (including Yoshua Bengio) to propose the so-called *generative adversarial networks* (GANs) [8], which has been called by Facebook's AI research director Yann LeCun "the most interesting idea in the last 10 years in machine learning".

To fully understand the power of this idea, we should explain the difference between two main categories of learning algorithms: the *discriminative* and the *generative* ones. The first category aims at building classifiers. For example, a machine that takes in input a picture of an animal and outputs a label representing the kind of animal: a cat, a dog, etc. To

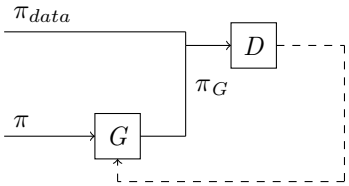

Figure 1 Scheme of a GANs, where π_{data} is the distribution of the real data, and π is the random number generator.

construct such model, typically the learning phase consists in feeding the machine with a huge amount of pairs of the form (animal, label), called training examples. Processing these examples brings to the tuning of the nodes of the net in such a way that, later on, when the machine is given a new picture (possibly never seen in training phase), it will be able to associate to it the most likely label.

In contrast, a generative algorithm should be able to *generate new examples*. For instance, the generative counterpart of the model above, should be able to generate a picture of a new dog (never seen in training phase) each time it is requested.

In 2014, while the research on discriminative algorithms had made enormous progress in the last decade and the corresponding models were being deployed successfully, generative algorithms were still an open challenge. In his seminal paper, Goodfellow and his colleagues had the idea of using an architecture consisting of two neural networks, pitting one against the other (thus the "adversarial"). The two nets are called respectively *generator* G and *discriminator* D. The purpose of G is to learn to transform a latent space of random numbers into data of the target domain, trying to imitate the (unknown) distribution of the data (the random generator in input is necessary because neural nets are essentially deterministic). The discriminator, on the other hand, tries to distinguish between real data sampled from the distribution and the synthetic data generated by the generator. The output of the discriminator is fed back to the generator, that in this way can learn if it has been successful or not in fooling the adversary. The goal of the generative network, indeed, is to trick the discriminator into thinking that the novel data produced is coming from true data distribution; this way, it increases the discriminator's error rate. The goal of the discriminator, of course, is to minimize its own error rate. Thus this game is in fact a two-player zero-sum game. The GANs architecture is represented in Figure 1. The interplay between the two networks is regulated by the following minimax problem:

$$\min_D \max_G V(D,G) \quad \text{where} \quad V(D,G) = \mathbb{E}_{x\sim\pi_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim\pi(z)}[\log(1 - D(G(z)))]$$

GANs are used widely in image generation, video generation and voice generation, and their results are impressive. As an example, Figure 2 shows the outcome of a GANs model developed in [11]. GANs are also used to crete artistic works according to a given style, see for instance Figure 3.

3 Game theory and privacy

The huge amount of data collected by digital service providers and the development of powerful technologies to perform analytics, in particular ML, have exacerbated the risks for privacy. See for example the model inversion attacks [7] and the membership inference attacks [17].

Figure 2 Synthetic images representing human faces (1024 × 1024 pixels) generated using the CELEBA-HQ dataset [11].

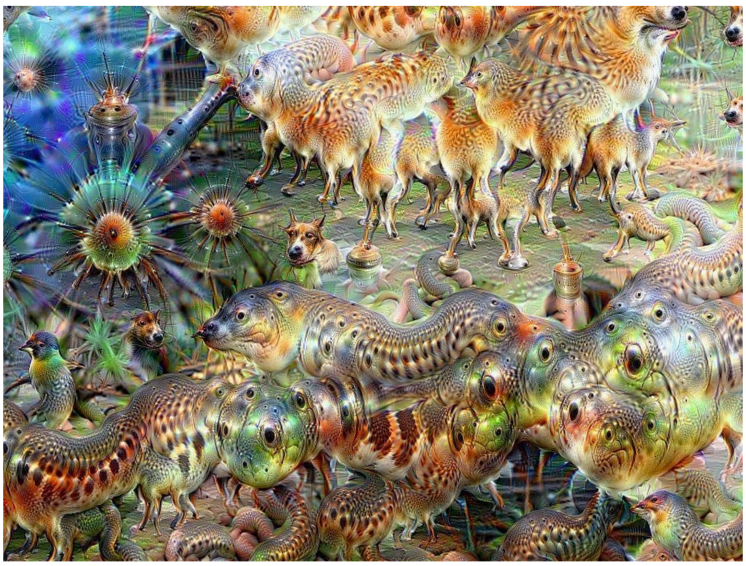

Figure 3 An artwork created using DeepDream, which researchers at Google developed in 2015.

However, if ML can be a threat to privacy, it can also be exploited as a powerful means to build "good" privacy-protection mechanisms, i.e. mechanisms that are robust and which preserve a good utility. In particular, we consider mechanisms that achieve privacy by adding controlled noise to the original data. Following the approach of [18], we aim at maximizing the privacy protection while ensuring the desired quality of service (QoS) that the user receives when providing obfuscated data. In this paper we focus on *location privacy* and on the risk of the *re-identification* of a user from his location, but the framework we develop is general and can be applied to any scenario in which there is sensitive information that we wish to hide, correlated to information that we need to make public.

In the context of location privacy, the QoS constraint (aka *utility* constraint) is typically expressed as a bound on the expected distance between the real location and the obfuscated one (cfr. [18, 4]). This is a linear constraint. If also privacy is measured by a linear expression, then in principle it is possible to use linear programming to compute a mechanism that provides the optimal privacy-utility trade-off (cfr. [18, 6, 15]). The limitation of this approach, however, is that it does not scale to large datasets. The problem is that the linear program

needs one variable for every pair (w, z) of real and obfuscated locations. Such variables represent the probability of producing the obfuscated location z when the real one is w. For a 50×50 grid this is more than six million variables, which is already at the limit of what modern solvers can do. For a 260×260 grid, the program has 4.5 billion variables, making it completely intractable (we could not even launch such a program due to huge memory requirements).

In order to overcome this limitation, in [16] we have explored an approach based on ML. Inspired by the GANs paradigm [8], we constructed a system of two adversarial neural networks: a *generator* G and *classifier* C. G generates noise so to confuse the adversary as much as possible, within the boundaries of the utility constraints, while C takes the noisy locations produced by G as inputs and tries to re-identify (classify) the corresponding user. Note that there are considerable differences with the standard GANs paradigm: in the latter the generator tries to reproduce a known target data distribution, while in our setting the target distribution that optimizes the data obfuscation is unknown and G has to "invent" it.

The interplay between G and C can be seen as an instance of a zero-sum Stackelberg game [18], where G is the *leader*, and C is the *follower*, and the payoff function f is the privacy loss. Finding the optimal point of equilibrium between G and C corresponds to solving a minimax problem on f with G being the minimizer and C the maximizer.

The definition of f was particularly critical in our setting. A first idea would be to measure the privacy in terms of C's capability of re-identifying users given their locations (i.e. classification accuracy). However this would be a poor choice. In fact, consider the following scenario: two users, A and B, are respectively in two locations a and b. C is trained on this dataset and learns that a corresponds to A and b to B. At the next iteration, G will maximize C's misclassification error by swapping the locations so that a corresponds now to B and b to A. But at the next iteration C is trained again and learns the new correct classification. G would then keep swapping the locations without ever reaching the equilibrium point. A similar example was independently pointed out in [1].

In order to address this issue we adopted a different payoff function f, one which is less sensitive to the particular labeling strategy of C and takes into account not just the *precision* of the classification, but, rather, the *information* contained in it. A way to formalize this intuition was to use the *mutual information* $I(X;Y)$, where X, Y are respectively the random variables associated to the *true ids*, and to the ids resulting from the classification (*predicted ids*). We recall that $I(X;Y) = H(X) - H(X|Y)$, where $H(X)$ is the entropy of X and $H(X|Y)$ is the residual entropy of X given Y.

The nets G and C tries to minimize/maximize $I(X;Y)$ according to the following minimax game formulation:

$$\min_G \max_C I(X;Y).$$

where the minimization by G is on the mechanisms producing the noisy points Z and which satisfy the utility constraint, while the maximization by C is on the classifications that map Z into Y.

The adversarial networks are organized as follows: G takes in input a pair $(x, w) \sim (X, W)$ consisting of an identifier x and an associated location w, and two random seeds. (The latter are necessary because a neural network by itself is deterministic.) The goal of the classifier it to transform the random inputs in "clever" probabilistic noise to add to the two coordinates of the location, taking into account the utility consraint. The noisy location produced by G is given in input to C, which tries to re-identify it. Note that C is helped in this by the fact that X and W are correlated, and that Z is also correlated to W and therefore to X, because of the utility constraint. The result of C is then fed back to G, which uses it to

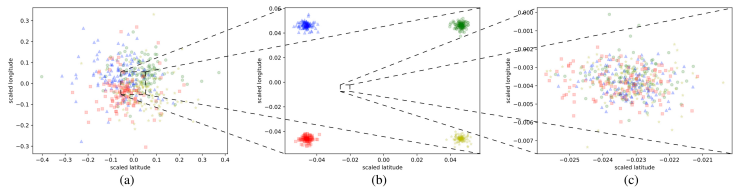

Figure 4 Synthetic testing data. From left to right: Laplace noise, no noise, our noise. $L = 270$m.

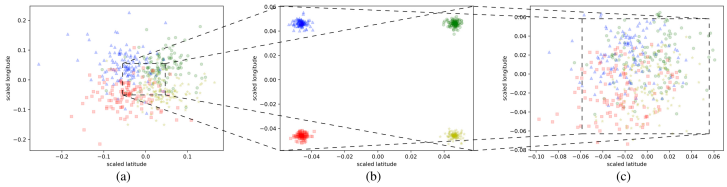

Figure 5 Synthetic testing data. From left to right: Laplace noise, no noise, our noise. $L = 173$m.

refine the mechanism.

In the next section we evaluate the performance of our mechanism with respect to others used in location privacy. We fix the level of utility and we compare the resulting Bayes error of X given Z, which is defined as $B(X \mid Z) = \sum_z P_Z(z)(1 - \max_x P_{X|Z}(x \mid z))$. The use of this metric is justified by the fact that it can be proved that $B(X \mid Z)$ is a lower bound for $B(X \mid Y)$ for any possible classifier that derives Y from Z. It is indeed the error of the "ideal" Bayesian classifier, which is optimal, and represents, therefore, the strongest possible adversary for the given Z.

The other two mechanisms under comparison are the (planar) Laplace [4] and the optimal mechanism, namely the mechanism that induces the highest $B(X \mid Z)$ for the given utility loss.

4 Experiments

We perform four different experiments, over synthetic and Gowalla [12] data, and with two different utility bounds.

The first experiment is on synthetic data with a loose utility constraint. We set the domain of W and Z to be a square of side $6500m$. We create a dataset of 600 synthetic check-ins (locations) for each one of 4 users (classes). We use 480 check-ins for training and validation, and 120 for testing. We place the check-ins randomly in clouds of $50m$ max around the four vertices of a square of side 300m (each user corresponds to a vertex). We set the bound on utility, namely the maximum expected distance L between Z and W (aka *distortion*), to be $270m$. Note that such L is enough for each check-in to be remapped into the center of the square, that is what the optimal mechanism would do. In fact, if all noisy locations coincide then they give no information about the identity of the corresponding user.

We first apply a Laplace mechanism to the test data (cfr. Figure 4(a)). For the sake of the comparison, we have set the parameters of the Laplace to be such that the average distortion is at least $270m$. In fact, it results to be $\approx 298.40m$. Then, we train our mechanism and we apply it to the test data (cfr. Figure 4(c)), obtaining an average distortion of $\approx 219.60m$. Note that, while the Laplace tends to "spread out" the obfuscated check-ins, our method tends to concentrate them into a small area in the center, thus approximating the optimal

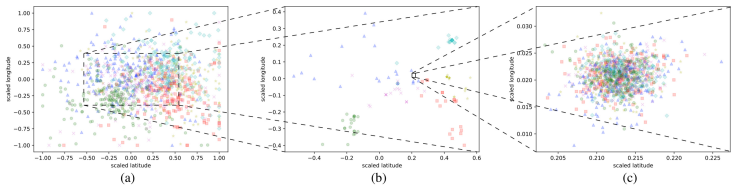

Figure 6 Gowalla testing data. From left to right: Laplace noise, no noise, our noise. $L = 1150$m.

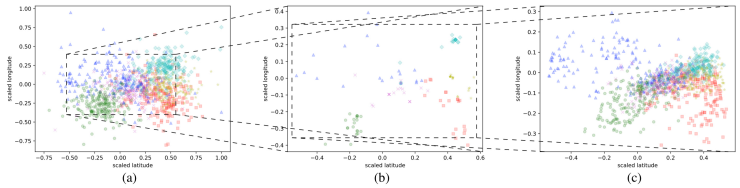

Figure 7 Gowalla testing data. From left to right: Laplace noise, no noise, our noise. $L = 518$m.

mechanism. Remapping all check-ins into the same location may not be possible in general, as it depends on the utility constraint. Nevertheless, we can expect that our mechanism will tend to overlap the check-ins of different classes as much as allowed by the utility constraint. With the Laplace, on the contrary, the zones of the various classes tend to remain separated and the noise is injected in a symmetrical fashion and remaps the check-ins into zones that remain separate and hence do not contribute to increase privacy.

We now evaluate $B(X \mid Z)$. To this aim we need to discretize Z, and we do so by using a 260×260 cells grid where each cell has $25m$ long side, i.e. $6500/260$. We apply the mechanisms on each check-in in the test set for 500 times with different random seeds thus obtaining $4 \times 120 \times 500$ obfuscated check-ins. In a real situation this would correspond to a scenario in which a user is in a certain location multiple times and each time reports different noisy locations. Figure 8(a) represents the Bayes errors for the three mechanisms. Note that, since the optimal mechanism in this case leaks no information about X (identifiers), its Bayes error is the same as that of randomly guessing the identifier. Since there are 4 identifiers, and as classes they are balanced, $B(X \mid Z) = 1 - 1/4 = 0.75$.

The second experiment is the same as the first, except now the utility bound is $173m$. This bound is not enough to map all check-ins to the center of the square, but it is enough to map the check-ins of two adjacent vertices into the same point in the middle of the corresponding side. Hence one of the possible optimal mechanisms is the one that induces a partition of the identifiers in two sets, where each set contains two indistinguishable identifiers. The Bayes error of an optimal mechanism is therefore $B(X \mid Z) = 1 - 1/2 = 0.50$. The Laplace method and ours produce the distribution in Figures 5(a) and 5(c). The Bayes error is reported in Figure 8(b).

The third experiment is on data extracted from the *Gowalla* dataset [12]. We set the domain of W and Z to be a squared region of with side 4500m and centered in 5, Boulevard de Sébastopol, Paris. We select the 6 users who checked in the region most frequently. (We limit the classes to 6 for visualization sake, but our method can handle an unbound number of classes.) For each location we create 10 repetitions with different random seeds so to increase the dataset. Again for the sake of visualization, we filter the check-ins so obtained to reduce the overlapping of those belonging to different classes. We divide the resulting check-ins into 4920 for training and validation, and 1200 for testing. Fig. 6(b) shows the test

Synthetic data, low utility		
Laplace	Ours	Optimal
0.39	0.74	0.75

(a)

Synthetic data, high utility		
Laplace	Ours	Optimal
0.23	0.42	0.50

(b)

Figure 8 Bayes error on synthetic data, for the Laplace mechanism, our mechanism, and the optimal mechanism, on a grid of 260×260 cells.

Gowalla data, low utility		
Laplace	Ours	Optimal
0.33	0.80	0.83

(a)

Gowalla data, high utility		
Laplace	Ours	Optimal
0.28	0.38	?

(b)

Figure 9 Bayes error on Gowalla data, for the Laplace mechanism, our mechanism, and the optimal mechanism, on a grid of 260×260 cells. In the last table the Bayes error of the optimal mechanism is unknown: the linear program contains 4.5 billion variables, making it intractable in practice.

data. We set the utility threshold to $L = 1150m$, which is enough to map all check-ins into a unique location. Therefore the Bayes error of the optimal mechanisms is the same as that of the random guess, i.e., $B(X \mid Z) = 1 - 1/6 \approx 0.83$. The distribution for Laplace and ours are in Figure 6(a) and 6(c) respectively, and the corresponding Bayes errors are reported in Figure 9(a).

Finally, the setting of the fourth experiment is like the third one, except we now set a stricter threshold, $L = 518m$. We obtain the data distribution in Figure 7(a) for Laplace and Figure 7(c) for our mechanism. The corresponding Bayes errors are reported in Figure 9(b). In this case we do not know the optimal mechanism: we cannot figure it theoretically and we cannot compute it due to the large number of variables.

References

1 Martín Abadi and David G. Andersen. Learning to protect communications with adversarial neural cryptography. *CoRR*, abs/1610.06918, 2016. URL: `http://arxiv.org/abs/1610.06918`, `arXiv:1610.06918`.

2 Samson Abramsky, Dan R. Ghica, Andrzej S. Murawski, and C. H. Luke Ong. Applying game semantics to compositional software modeling and verification. In Kurt Jensen and Andreas Podelski, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 421–435, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

3 Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, 2000.

4 Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: differential privacy for location-based systems. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS 2013)*, pages 901–914. ACM, 2013. URL: `http://doi.acm.org/10.1145/2508859.2516735`, `doi:10.1145/2508859.2516735`.

5 Andreas Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56(1–3):183–220, 29 April 1992.

6 Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 21th ACM Conference on Computer and Communications Security (CCS 2014)*, 2014.

7 Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22Nd ACM SIGSAC*

Conference on Computer and Communications Security, CCS '15, pages 1322–1333, New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2810103.2813677, doi:10.1145/2810103.2813677.

8 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

9 J. M. E. Hyland and C.-H. Luke Ong. On full abstraction for PCF: i, ii, and III. *Information and Computation*, 163(2):285–408, 2000.

10 John Forbes Nash Jr. Non-cooperative games. *Annals of Mathematics*, 2(54):286–295, 1951.

11 Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, (ICLR)*. OpenReview.net, 2018.

12 Jure Leskovec and Andrej Krevl. The Gowalla dataset (Part of the SNAP collection). https://snap.stanford.edu/data/loc-gowalla.html.

13 Paul Lorenzen and Kuno Lorenz. *Dialogische Logik*. Kurztitelaufnahme der Deutschen Bibliothek. Wissenschaftliche Buchgesellschaft, [Abt. Verlag], 1978.

14 John Von Neumann. Zur theorie der gesellshaftsspiele. *Mathematical Annals*, 100:295–320, 1928.

15 Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1959–1972. ACM, 2017. URL: http://doi.acm.org/10.1145/3133956.3134004, doi:10.1145/3133956.3134004.

16 Marco Romanelli, Catuscia Palamidessi, and Konstantinos Chatzikokolakis. Generating optimal privacy-protection mechanisms via machine learning. In *Proceedings of the IEEE International Symposium on Computer Security Foundations (CSF)*, 2020. URL: http://arxiv.org/abs/1904.01059, arXiv:1904.01059.

17 Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18. IEEE Computer Society, 2017. URL: https://doi.org/10.1109/SP.2017.41, doi:10.1109/SP.2017.41.

18 Reza Shokri, George Theodorakopoulos, and Carmela Troncoso. Privacy games along location traces: A game-theoretic framework for optimizing location privacy. *ACM Transactions on Privacy and Security*, 19(4):11:1–11:31, 2017. URL: http://doi.acm.org/10.1145/3009908, doi:10.1145/3009908.

19 Colin Stirling. Bisimulation, modal logic and model checking games. *Logic Journal of the IGPL*, 7(1):103–124, 1999.

20 John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, Princeton, 1944.