Adversarial Link Prediction in Spatial Networks

Michał Tomasz Godziszewski
University of Warsaw; IDEAS NCBR
Warsaw, Poland
mtgodziszewski@uw.edu.pl

Yevgeniy Vorobeychik
Washington University in St. Louis
St. Louis, MO, USA
yvorobeychik@wustl.edu

Tomasz Michalak
University of Warsaw; IDEAS NCBR
Warsaw, Poland
tpm@mimuw.edu.pl

ABSTRACT

Social networks arise as a result of complex interactions among people, and homophily plays an important role in this process. If we view homophily as a dominant force in network formation and associate each node with a collection of features, this process gives rise to spatial networks, with likelihood of an edge an increasing function of feature similarity among its incident nodes. A link prediction problem in such spatial networks then amounts to determining whether the pair of nodes are sufficiently close according to this edge likelihood function. We undertake the first algorithmic study of the adversarial side of this problem in which the adversary manipulates features of a subset of nodes on the network to prevent predicting target edges. We show that this problem is NP-hard, even if the edge likelihood function is convex. On the other hand, if this function is convex, we show that the problem can be solved with convex programming when the set of nodes that the adversary needs to manipulate is fixed. Furthermore, if the edge likelihood function is linear, we present approximation algorithms for the case when the features are binary, and we wish to hide only a single edge, and for the case when the features are real-valued but we need to hide an arbitrary collection of edges.

KEYWORDS

Social network analysis; Adversarial learning; Computational complexity

ACM Reference Format:
Michał Tomasz Godziszewski, Yevgeniy Vorobeychik, and Tomasz Michalak. 2023. Adversarial Link Prediction in Spatial Networks. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Homophily [16]—the property that relationships are more likely among more similar individuals—is an important social driver of network emergence. In the abstract, a natural way to capture homophily is by embedding nodes in a metric space. Distance between nodes so embedded, and associated links, can then capture a variety of phenomena, such as spatial proximity (with friendships more likely to emerge through frequent contact), the similarity of opinions (where individuals who are in greater overall agreement more likely to establish long-term relationships), cultural similarity, and so on.

Such similarities are not considered by traditional network analysis that typically focuses on the graph structure representing observed relationships. However, when we have access not only to the network structure but also node features, this additional information can be further leveraged for network analysis tasks [1, 11]. Let us take as an example the task of link prediction, in which we are particularly interested in this paper. The aim of this fundamental problem in social network analysis is to anticipate the existence of connections that are missing from the data or that are yet to be created [14]. Various applications, such as detecting concealed relationships in organized crime, have motivated consideration of adversarial vulnerabilities of link prediction algorithms [5, 8, 26, 27]. This line of research is focused predominantly on vulnerabilities of similarity metrics that solely use the network topology, such as the set of common neighbors. However, if link prediction leverages node attributes rather than, or in addition to, observed edges, perturbing these attributes becomes an essential means of an attack that has scarcely been analysed in the literature.

Against this background, we propose to study *adversarial link prediction in spatial networks (Adversarial LPSN)* where the probability of a link is a function of the attributes (features) of the nodes incident to it. In our model, the adversary aims to hide a collection of target links by selecting a subset of nodes on the graph and perturbing their features subject to a perturbation budget constraint. We begin by considering real-valued features. We first show that this problem is NP-Hard and, indeed, hard to approximate to arbitrary precision, even if the edge likelihood function is convex. On the other hand, we give an algorithm for this problem that yields a 2-approximation. Furthermore, we show that if we first *fix* the set of nodes that the adversary may manipulate, the problem can be solved using convex programming. Next, we consider binary features and show that the problem is NP-hard, even if we wish to hide a single edge. However, we also give a 2-approximation algorithm for this problem.

In summary, we make the following contributions:

(1) We present a novel model of adversarial perturbations to link prediction in spatial networks.
(2) We show that the adversarial problem we define is inapproximable in general. Moreover, it remains NP-hard if features are binary, even if the link likelihood function is convex.
(3) We give a convex programming solution to a special case when the set of nodes that can be adversarially perturbed is fixed, the link likelihood function is convex, and features are real-valued.
(4) We provide a polynomial algorithm with a 2-approximation guarantee if features are binary, only a single node is perturbed, and the link likelihood function is linear.
(5) Finally, we provide a polynomial time algorithm with a 2-approximation guarantee if features are real-valued.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

2 RELATED WORK

Various evasion techniques against several social network analysis tools have been investigated in the literature with a typical goal of privacy protection. Unfortunately, it was shown that social network analysis tools could be used to infer even undisclosed information about social media users, including their sexual orientation or other sensitive characteristics [17].Given this, Waniek *et al.* [24] studied how to avoid identification by community detection algorithms. Another sample research line of this literature are the papers that analyse manipulating centrality measures [4, 23, 25].

Our model is also closely related to decision-time (or adversarial perturbation) attacks in adversarial machine learning (see, e.g., [2, 9, 22]), particularly if we view link prediction as binary classification. A decision-time attack on a binary classifier involves an adversary adding a (typically norm-bounded) perturbation to inputs in order to cause these to be misclassified by the model. However, there are several crucial differences between our model and this literature. First, in our model, features of nodes enter the predictions *indirectly* via the weighted distance calculation; for example, even if the classifier is linear in inputs, it is non-linear in perturbations. Second, and more significant, is the fact that in our model, the adversary may select and perturb a subset of nodes on the network, and the goal is to impact predictions for a *collection* of links, adding an important combinatorial dimension to the problem.

Models, in which an underlying geometry is present, are actually natural for applications in data science and network analysis (we can think of network nodes as being represented by a feature vector in some d-dimensional vector space, with the nodes that share similar features being the neighbors of a given node). The spatial constraints imposed on the graph are also relevant for the geometric graph theory, especially for the studies of random geometric graphs. Our model is related to the recent investigations from [12], where the authors investigate the problem of distinguishing an Erdös-Renyi (standard) random graph $G(n, p)$ from a random geometric graph $Geo_d(n, p)$, where n vertices of the graph are identified with an independently and uniformly sampled vector from the d-dimensional unit sphere, and the pairs of vertices are connected by an edge if the vectors are *sufficiently close* to each other, so that that the marginal probability of an edge existence is equal to p. As the authors note, geometric random graphs, both in the high-dimensional setting as studied in the aforementioned paper, as well as in the more familiar setting of low-dimensional geometric graphs (as studied e.g. in the classic monograph [19]) could be a widely applicable benchmark for some computational methods used in the so-called semi-random setting. Here, we focus on spatial networks representable in a finite-dimensional normed space that are generated stochastically, which may be seen as a working case of the geometric random graph model described.

3 BACKGROUND

We consider link prediction in the so-called spatial networks, represented by graphs where nodes correspond to points in $\mathbb{R}^D$. Formally, a spatial network is a graph $G = (V_G, E_G)$ where the nodes $V_G \subset \mathbb{R}^D$ correspond to a finite subset of $\mathbb{R}^D$ and each edge $(x, y) \in E$ connecting a pair of nodes $x, y \in V_G$ is generated stochastically as follows. Let

$$z(x, y) := (|x_1 - y_1|^p, \ldots, |x_D - y_D|^p)$$

for a fixed $p \geq 1$. We will refer to this function as a p-vector function. We assume that the probability of an edge is determined by a function $f(z(x, y))$ for any pair of nodes (x, y). We refer to f as the *edge likelihood function.*

In a link prediction setting, we observe a graph with the entire set of nodes V, but only a subset of existing edges $E \subseteq \tilde{E}$, and our goal is to identify whether particular other edges in the graph exist or not. A natural link prediction algorithm in such spatial networks would first learn f (e.g., f can be a logistic regression) based on the observed edges in the network (using attributes of their incident nodes), and then use f to predict unobserved links. Note that this process makes use of the observed network structure in learning the function f, but once f is given, this structure is no longer required in link prediction.

A special type of spatial networks that we will consider here and an example that allows illustrating the idea behind them are the two-dimensional Euclidean graphs. In these graphs, the nodes are identified with points in the Euclidean two-dimensional plane, and to each link between any two points, we assign lengths equal to the Euclidean distance between those points. It is perhaps worth noting that the class of spatial graphs we investigate in this paper is not identical with the class of the so-called planar networks, where it is required that the nodes are embedded in the two-dimensional plane in such a way that the links do not intersect each other.

Formally, a two-dimensional Euclidean network is a graph $G = (V, E)$ such that each node in V is associated with a point in the Euclidean two-dimensional plane, i.e., each $v \in V$ is represented by a point $x_v \in \mathbb{R}^2$. Furthermore, with each link $e = \{v, w\} \in E$, we associate a distance $\delta(v, w)$ that is equal to the Euclidean metric.

Clearly, all norms and the metrics induced by these norms can be considered in their weighted versions, where computing the norm involves, additionally, a multiplication by a particular function (or constant) referred to as the weight.

4 ATTACK MODEL

Suppose that an Attacker is a malicious agent who aims to hide some of the links (for example, relationships among particular malicious nodes). In our context, hiding a link entails ensuring that the edge likelihood function f is erroneously perceived as small for the target pair of nodes $\{x, y\}$. We suppose that the Attacker can modify the perceived value of f by adding *adversarial perturbations* δ to a subset of nodes on the network $A \subseteq V_G$ (potentially including nodes x and y). We assume that both the network G observed by the Analyst and the edge likelihood function f (including the function $z(x, y)$) are known to the Attacker, but the Attacker is limited in the extent to which the features of any node can be perturbed, for example, to avoid becoming highly suspicious. We capture this constraint formally as follows. Let $u \in A$ be a node in the spatial network for which original features u are changed by the adversary into u', and define $\delta = u' - u$. We constrain that any perturbation δ to any node $u \in A$ satisfies $\|\delta\| \leq \epsilon$ where $\epsilon > 0$ is an exogenously specified limit on how much the Attacker can perturb any node, and $\| \cdot \|$ is an ℓ_p norm with $p \geq 1$.

We begin by considering a simplified version of the Attacker problem in which the Attacker aims to hide a link between a single pair of nodes x and y and can only add perturbation δ to one of them—say, y, without loss of generality (i.e., the subset of nodes being attacked is $A = \{y\}$). In this special setting, we formally define the following adversarial optimization problem in which the goal of the adversary is to modify a node y adjacent to a target link $\{x, y\}$ so as to minimize the perceived likelihood of this link by the Analyst, that is, to minimize $f(z(x, y + \delta))$ over feasible perturbations δ to y.

Problem 1. *Given a positive real number ϵ, a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$ and a pair of nodes $x, y \in V_G$ such that $\{x, y\} \notin E_G$ find*

$$\min_{\delta \in \mathbb{R}^D} f(z(x, y + \delta)), \quad s.t. \quad ||\delta|| \leq \epsilon.$$

A natural variation of this problem is the following formulation, in which we instead minimize the magnitude of the adversarial perturbation that enables us to hide the target link by ensuring that the resulting perceived link likelihood $f(z(x, y + \delta))$ is below an exogenously specified threshold θ.

Problem 2. *Given a real number $\theta \in [0, 1]$, a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$ and a pair of nodes $x, y \in V_G$ such that $\{x, y\} \notin E_G$ find*

$$\min ||\delta||, \quad s.t. \quad f(z(x, y + \delta)) \leq \theta.$$

In this formulation, the threshold parameter θ represents the threshold used by the Analyst in link prediction, so that whenever f is below the threshold, the Analyst is expected to predict that the target pair of nodes are not connected. This variation, therefore, requires the Attacker to also know the threshold θ used by the Analyst or at least have a conservative bound on this threshold.

Both problem formulations above are conceptually similar to the general class of decision-time attacks on classifiers in the adversarial machine learning literature [22]. The principal difference is that the impact of modifications on the target function f is indirect, mediated by $z(x, y)$. This is inconsequential if features are real-valued and we use gradient-based methods (such as PGD [15]) to heuristically solve Problems 1 and 2, but it becomes important if features are binary and we wish to take advantage of special structure of f, such as linearity: even if f is linear in z, it is non-linear in y.

A decision-theoretic version of the problems above, which we refer to as *Adversarial Link Prediction in Spatial Networks* or ALPSN, simply imposes both the constraint that perturbations δ are bounded and that the attack succeeds.

Problem 3 (Adversarial Link Prediction in Spatial Networks (ALPSN))**.** *Given real numbers $\theta \in [0, 1]$ and $\epsilon > 0$, a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$, and a pair of nodes $x, y \in V_G$ such that $\{x, y\} \notin E_G$, decide if there exists a D-vector $y' = (y'_1, \ldots, y'_D)$ with $||y - y'|| \leq \epsilon$ such that $f(z(x, y')) \leq \theta$.*

One might ask: why should the information induced by the (weighted) distances, even if implicit in the p-vector function z, be taken into account while analyzing the mere link prediction task? The answer is that the relations of the (tuples of) features of the nodes in the network are important from the learning-theoretic point of view. Consider the likelihood function f and the problem of learning it. Access to the metric structure of the features of the nodes in the network (features that correspond to coordinates of the nodes in this space) can enable learning a more fine-grained and precise link prediction function f. Then, if the algorithms realizing this learning task can use the information contained in the geometric structure of the network (information available thanks to the spatiality of the network, i.e., of its embedding into some metric space), i.e., if learning the function f can be dependent on the z-function (induced by the distances in the abstract metric space the network is embedded into), then can become a point of attack of the adversary. Therefore, attacking the learning process of the spatial likelihood function used for link prediction might consist in introducing perturbation directly to the features of the nodes, i.e., to their locations in the abstract metric space the network is embedded into.

Next, we consider a natural generalization of the problem in which the adversary *chooses* a subset of nodes A, as well as modifies the features of all of these nodes as above. The key additional aspect of this problem is that we impose a cardinality constraint that $|A| \leq k$ for an exogenously specified k. Note, however, that the problem remains non-trivial even if $k = |V_G|$. This general variant is a substantive departure from conventional adversarial machine learning approaches.

Problem 4 (Set Adversarial Link Prediction in Spatial Networks (SALPSN))**.** *Given real numbers $\theta \in [0, 1]$ and $\epsilon > 0$, a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$, a positive integer $k < |V_G|$, a subset $S \subseteq V_G$ of vectors from the set of nodes of G, and a target set $H \subseteq V_G \times V_G$ of pairs of nodes such that for each $\{x, y\} \in H$ it holds that $\{x, y\} \notin E_G$, decide if there is a set $A \subseteq S$ of at most k vectors $v' = (v'_1, \ldots, v'_D)$ with $||v - v'|| \leq \epsilon$ such that in the graph $G' = (V'_G, E_G)$, where $V'_G = V_G \setminus A \cup \{v' : v \in A\}$, for each $\{x, y\} \in H$ it holds that $f(z(x, y)) \leq \theta$. An instance of the problem is a tuple $(G, S, \theta, \epsilon, H, k)$.*

The two distinguished special cases of the problem are for the set S being (a) V_G, or (b) $V_G \setminus dom(H)$, where for $H \subseteq V_G \times V_G$:

$$dom(H) = \{v \in V_G : \exists w \in V_G : (v, w) \in H \ \vee \ (w, v) \in H\}.$$

The set A in the definition of SALPSN will be referred to as the displacement set of an instance of the problem. Observe that, in general, nothing prevents the nodes x or y from being in the set A. This is an inconsequential issue: whether x, y are allowed to be elements of A does not alter the results below.

The final variant of the problem we consider fixes the set A of nodes that can be displaced.

Problem 5 (Fixed-SALPSN)**.** *Given a tuple $(G, \theta, \epsilon, H, A)$, where $F \subseteq V_G$ is a fixed subset of nodes, decide if there exists a set F' of vectors $F' = \{v' : v \in A\}$ with $||v - v'|| \leq \epsilon$ such that in the graph $G' = (V'_G, E_G)$, where $V'_G = V_G \setminus A \cup \{v' : v \in A\}$, for each $\{x, y\} \in H$ it holds that $f(z(x, y)) \leq \theta$.*

5 COMPLEXITY ANALYSIS

For the complexity analysis of these problems, we use specific geometric tools from algorithmic graph theory. In particular, we will employ the so-called penny graphs, machinery that very recently

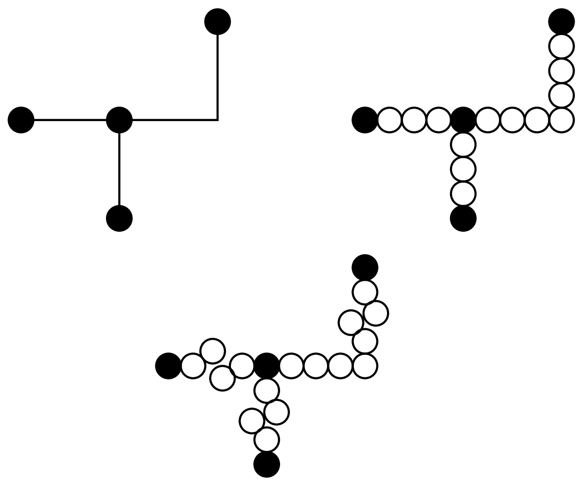

Figure 1: Construction of a penny graph from a planar graph of degree at most three, so that each link is replaced by intermediate nodes whose number is a multiple of four.

has been successfully applied for complexity analysis in computational social choice theory [7]. A penny graph is defined by a set of unit disks, i.e., balls of diameter one in $\mathbb{R}^2$, such that no two disks overlap (but they can touch). Each disk corresponds to a node, and two nodes are connected by a link if their disks touch (i.e., if their centers are precisely at a distance of 1). A graph is a penny graph if it has such a representation by unit disks (the name comes from the analogy between the disks and pennies laying on a flat surface).

Obviously, all penny graphs are planar. In our construction, we are using penny graphs of degree three obtained via the following result of [21].

Lemma 1. *[21]. There is a polynomial-time algorithm that, given a planar graph with maximum degree of at most 4, computes its embedding on the two-dimensional Euclidean real space so that its nodes are at integer coordinates, and its links are represented by vertical and horizontal line segments.*

Recall that in the Vertex Cover problem (VC) we are given a graph $G = (X, E)$ and a positive integer r. We ask if there exists a vertex cover of G—i.e., a subset of nodes $U \subseteq X$ of size at most r such that each link $\{x, y\} \in E$ has an end in U, i.e., at least one of the nodes x, y is in U. It is known that the problem is NP-hard for cubic planar graphs [18, Theorem 4.1(a)]. Given an instance (G, r) of VC, where G is a cubic planar graph, we can construct an instance of VC for penny graphs as follows (we use the construction of [3, Theorem 1.2]; we repeat it here as we need its specific properties).

First, we use Lemma 1 to obtain a planar representation of G, where the nodes are at integer coordinates and the links consist of vertical and horizontal line segments (see the left-hand side of Figure 1; note that in this figure the nodes have degrees at most three, and not exactly three). Second, we multiply node coordinates by four, ensuring that the lengths of the line segments forming the links also are multiples of four. Third, for each node v, we put a unit disk centered at the position of v, and we replace all the line segments forming the links by sequences of consecutive unit disks (located on the integral points within these lines; see the center of Figure 1). This way, each link becomes a sequence of $4t - 1$ disks, where t is an integer (possibly different for each link). Finally, for each link we introduce a single *local displacement*, which consists of replacing the second disc that lies on the link with two tangent disks (it does not matter from which end we start counting the disks); these two disks are also tangent to the disks on the two sides of the disk that we replaced (see the right-hand side of Figure 1). Local displacements ensure that disks on the links come in multiples of four. All in all, we obtain a penny graph.

Let G' be the penny graph that we constructed. Each node of G' has either two or three adjacent nodes. The nodes with two neighbors correspond to disks put on the links, and we refer to them as *intermediate*. We call a node locally displaced if it corresponds to a disk that was introduced as a result of a local displacement. Let L be the total number of intermediate nodes. One can easily verify that G has a vertex cover of size r if and only if G' has an independent set of size $r' = r + L/2$ (this follows from the work of [3]). We refer to the penny graphs obtained by this construction as *almost integral*, and we use the fact that VC is NP-hard for them.

Let $\theta \in [0, 1]$ be any real number in the unit interval. A real-valued function $\mathbb{R}^D \to \mathbb{R}$ is θ-sensitive if there exists a non-negative real number ρ such that the following equivalence holds for any $x \in \mathbb{R}^D$:

$$f(x) \geq \theta \text{ iff } ||x|| \leq \rho.$$

Theorem 1. *The problem* SALPSN *is* NP*-hard for any θ-sensitive function $f : \mathbb{R}^D \to \mathbb{R}$ In particular, the problem is* NP*-hard for $D = 2$.*

Proof. We will actually prove a somewhat stronger hardness result. Namely, we will demonstrate that, already for the Euclidean metric on a two-dimensional plane, it is NP-hard to decide if we can displace at most k nodes in order to hide the links by reducing the edge likelihood functions.

The proof goes by a reduction from Vertex Cover for almost integral penny graphs. Let $I = (G, k)$ be an instance of the Vertex Cover problem for almost integral penny graphs, where $G = (X, E')$, and let $n = |X|$, and $m = |E'|$. We construct an instance J of SALPSN with the network being represented by a graph in the real plane with Euclidean metric such that there exists a vertex cover of size at most k in I iff there exists a displacement set of size at most k in J. First, we define the spatial network $G' = (V_{G'}, E_{G'})$. For each node $x \in X$ of the graph G construct a node of the network $v_x \in V_{G'}$ located in the same point as x, i.e., let $V_{G'} = \{v_x : x \in X\}$. Further let $E_{G'}$ be any set of pairs **not** containing any of the links from the penny graph G, i.e. let

$$E_{G'} \subseteq (V_{G'} \times V_{G'}) \setminus E'.$$

Let the target set of pairs H in J contain all the pairs that have been elements of the target set from the graph G, i.e., let $H = E'$. In what follows we actually demonstrate that our reduction works also for an arbitrary instance of SALPSN, where the target set is a superset of the set of edges from the penny graph, i.e., even if $H \supsetneq E'$, the reduction will be correct. By assumptions, for a fixed θ, the edge likelihood functions is θ-sensitive. Without loss of generality, put ρ from the definition of θ-sensitivity to be equal

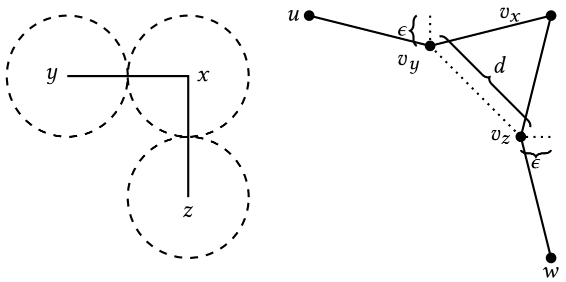

Figure 2: Estimating the distance between points – case 1: $\overline{xy}$ and $\overline{xz}$ are orthogonal. Even if the Attacker decides to displace the nodes v_x and v_y in a direction that makes them closer to each other than before the displacement took place, even if the pair $\{v_x, v_y\}$ was in the target set, then the distance between them remains above ρ so that the edge-likelihood θ threshold is still not exceeded. Observe how, in this case, the nodes u and w, neighboring v_y, and v_z, correspondingly, are also made distant from these nodes, allowing for the edge likelihood function to drop below θ for $\{u, v_y\}$, and $w, v_z\}$.

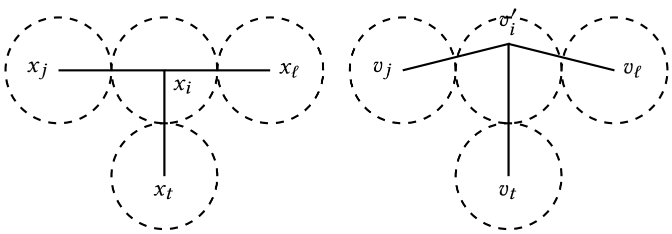

Figure 3: Case 5: The node x_i in the vertex cover is one of the vertex points of the penny graph that has three neighbors, one of which, say x_t, is connected to x_i along the line segment perpendicular to the line segment on which two of the other neighbors of x_i lie. Then we move v_i by a vector δ_i (of length ϵ) along the line connecting v_i to v_t, in the direction further away from v_t. Observe that the fragment of the picture consisting of the balls around the nodes x_j, x_i, and x_ℓ, adequately depicts the case 1, i.e. such that node x_i in the vertex cover is an intermediate node that has not been subject to local displacement in the process of constructing the almost integral penny graph, neither of its neighbors has been locally displaced, and x_i and both of its neighbors are colinear. Then we move the node v_i by δ_i (of length ϵ) vertically or horizontally, along the line that is perpendicular to the line segment on which the node v_i and its neighbors lie.

to 1, i.e., let it be that:

$$f\left(\sqrt{(v_{x,1} - v_{y,1})^2 + (v_{x,2} - v_{y,2})^2}\right) \geq \theta$$

if and only if

$$\sqrt{(v_{x,1} - v_{y,1})^2 + (v_{x,2} - v_{y,2})^2} \leq 1,$$

for any $v_x, v_y \in \mathbb{R}^2$.

Let $\epsilon > 0$ be any positive real number less or equal to $\frac{2-\sqrt{2}}{2} \approx 0.29$. The exact value of ϵ will directly follow from the estimations in the correctness proof below.

Finally, let k' be equal to k, i.e., we require that the maximal size of the displacement set in the constructed instance of SALPSN is equal to the maximal size of the vertex cover in the given penny graph. Observe that for each edge $(x_i, x_j) \in X'$ of the penny graph the distance between x_i and x_j is equal to 1. This ends the description of the reduction, now we will demonstrate its correctness. First, assume there exists a vertex cover of I of size at most k. Let it be denoted by

$$U = \{x_1, \ldots, x_k\},$$

and let the set of corresponding nodes of the network G' in J be denoted by

$$U' = \{v_1, \ldots, v_k\}.$$

Then, it is possible to displace each of the nodes v_i by a vector δ_i of norm ϵ in such a way that the distance between each pair of nodes corresponding to the links in the penny graph will be grater than 1. The way we obtain these displacements depends exactly on the relative position of the corresponding nodes in the penny graph:

(1) Node x_i in the vertex cover is an intermediate node that has not been subject to local displacement in the process of constructing the almost integral penny graph, neither of its neighbors has been locally displaced, and x_i and both of its neighbors are colinear. Then we move the node v_i by δ_i (of length ϵ) vertically or horizontally, along the line that is perpendicular to the line segment on which the node v_i and its neighbors lie.
(2) Node x_i in the vertex cover is an intermediate node that has not been subject to local displacement in the process of constructing the almost integral penny graph, neither of its neighbors has been locally displaced, but the lines connecting x_i to its neighbors are perpendicular. Then we move the node v_i by δ_i (of length ϵ) along the line that is within the angle of $\frac{\pi}{4}$ w.r.t. the lines on which the neighbors of v_i. lie.
(3) Node x_i in the vertex cover is an intermediate node that has not been subject to local displacement in the process of constructing the almost integral penny graph, but one of its neighbors has been locally displaced. Denote its non-locally displaced neighbor by u, and its locally displaced neighbor by w. Then move the node v_i by the vector δ_i (of length ϵ) along the line perpendicular to the line segment on which both v_i and u lie.
(4) Node x_i in the vertex cover is a locally displaced intermediate node of the penny graph. Then x_i is one of the nodes of the parallelogram x_j, x_i, x_ℓ, x_t, as depicted in Figure 4. We then move the node v_i by the vector δ_i (of length ϵ) along the line of the diagonal of the parallelogram in the direction *outside* of the parallelogram.
(5) Node x_i in the vertex cover is one of the vertex points of the penny graph that has 3 neighbors, one of which, say u, is connected to x_i along the line segment perpendicular to the line segment on which two of the other neighbors of x_i lie. Then we move v_i by a vector δ_i (of length ϵ) along the line

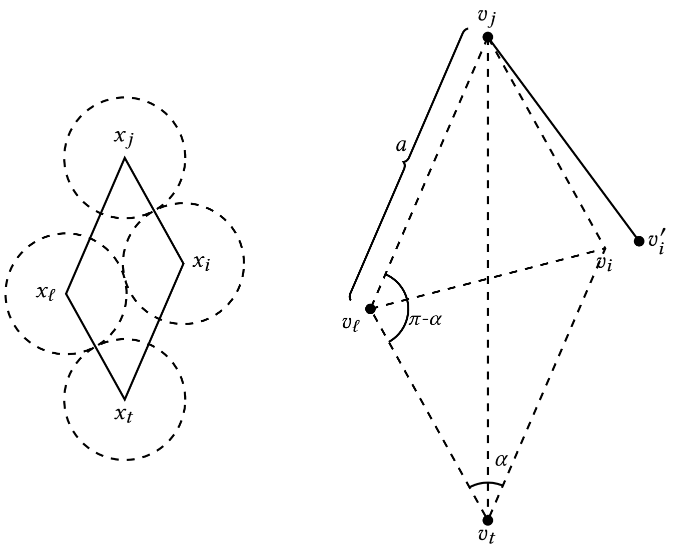

Figure 4: The disks at x_j and x_ℓ form a local displacement. Observe that moving node v_i to v_i' keeps the distances between v_i' and v_j and between v_i' and v_ℓ above threshold $\rho = 1$. The distance a between the nodes v_ℓ and v_j is unaffected by manipulating the node v_i and strictly greater than 1. This picture illustrates the displacement that can be made in the cases 3 and 4 in the description of the reduction.

connecting v_i to v_t, in the direction further away from v_t, as depcited in Figure 3.

In case 1, it is clear that the distance between the nodes on the line is equal to $\sqrt{1+\epsilon^2} > 1$. In case 2, the distance of the displaced node from its neighbors is:

$$\sqrt{(1+\epsilon\sqrt{2}/2)^2 + (\epsilon\sqrt{2}/2)^2} > 1.$$

In cases 3 and 4, the distance from the displaced node and its neighbors is again equal to $\sqrt{1+\epsilon^2}$ and:

$$\sqrt{(1+\epsilon\sqrt{2}/2)^2 + (\epsilon\sqrt{2}/2)^2}.$$

In case 5, additionally, the distance from v' to v_t is equal simply to $1+\epsilon$. If the target set of pairs contains more pairs than merely the set of links of the input almost integral penny graph, there is a problematic case we need to consider. It occurs when the vertex x_i corresponds to one of the original nodes from the planar graph, and by the grid structure of the penny graph G resulting from Valiant's lemma makes the lines $|x_i, x_j|$ and $|x_i, x_t|$ orthogonal to each other. In such a case the distance between v_j' and v_t' is easily seen to be:

$$d = (\rho - \epsilon)\sqrt{2},$$

where ϵ is the distance of v_j from the v_j' and the distance of v_t from v_t'. It is immediate to check that $d > \rho$ for:

$$\epsilon < \rho - \frac{\sqrt{2}}{4},$$

which for $\rho = 1$ is $1 - \frac{\sqrt{2}}{2} \approx 0.292$ which is greater than $\frac{2-\sqrt{2}}{2}$. Hence, requiring the points v' to be within distance $\epsilon < \frac{2-\sqrt{2}}{2}$ is clearly sufficient for the distances between corresponding agents in the network to be kept sufficiently large.

Observe that the above actually proves the correctness of the reduction - if I is a positive instance of the VC problem for almost integral penny graphs, the displacement described above gives the solution to the instance J SALPSN, and if I is a negative instance of VC, then at least one pair in the target set of links in J has to stay unsolved, as the corresponding link in I is uncovered. □

It is sometimes the case that geometric computational problems that are hard when stated for vector spaces of arbitrary finite dimension are polynomial-time solvable when the dimension is fixed. The result above, however, gives us a hardness phenomenon that is stronger than such fixed-parameter tractability with respect to the dimension. Since the theorem states NP-hardness for functions f defined on $\mathbb{R}^2$, it immediately follows that deciding SALPSN is NP-hard in general, i.e., the following holds:

Corollary 1. *The problem* SALPSN *is* NP*-hard for the Euclidean metric any θ-sensitive function $f : \mathbb{R}^D \to \mathbb{R}$, where D is any natural number greater or equal than* 2*. In particular, the problem is* NP*-hard even if the dimension D is fixed.*

Observe that the reduction used in the proof actually gives us an alternative, simpler proof of a couple of general graph-theoretic results relevant for the study of link or sign prediction in social networks. In particular in the problem of Eliminating Similarity, studied, e.g., by [5], the input consists of a graph, a set of targeted pair of nodes, a set C of links that can be removed, and the maximum number k of links that can be deleted. In this problem, we are asked to compute if there exist at most k links in C such that removing them results in every pair of nodes from the target set having a disjoint neighborhood. Assuming that we deal with spatial networks and nodes can be only connected by a link if their distance does not exceed a given threshold, our reduction immediately gives:

Corollary 2. *The problem of Eliminating Similarity is* NP*-hard even for spatial networks.*

Further, we can notice that the reduction is approximation-preserving. This means that inapproximability properties of Vertex Cover transfer to the SALPSN. In particular, since the construction of (almost integral) penny graphs is performed on the planar graphs of bounded degree, it follows that our reduction preserves inapproximability results for (planar) graphs with bounded degree. For specific results, one can consult the work by [10]. To be more specific, recall that an NP-optimization problem has an efficient polynomial-time approximation scheme (EPTAS) if it admits a polynomial-time approximation scheme whose time complexity is bounded by $O(f(1/\varepsilon)|x|^c)$, where f is a computable function and c is a constant. The class SNP (Strict NP) consists of the NP-problems that can be defined by a second-order formula $\exists S \forall \overline{x} \psi(\overline{x}, S)$, where ψ is quantifier-free. It is commonly believed that it is unlikely that all problems in SNP are solvable in subexponential time. Together with Theorem 2 from the paper by [10], our reduction implies the following:

Theorem 2. *The* SALPSN *problem has no* EPTAS *of running time* $2^{o(\sqrt{1/\varepsilon})} n^{O(1)}$*, where* $\varepsilon > 0$ *is the given error bound, unless all* SNP *problems are solvable in subexponential time.*

In the next section, we will demonstrate positive approximation results for versions of the problem. Before that, however, we need to note that

Theorem 3. *The problem* ALPSN *is* NP*-hard, if the features of the vectors* $x, y \in \mathbb{R}^D$ *are binary, even if the edge likelihood function* f *is convex.*

Proof. The proof goes by a reduction from the Subset Sum problem. Fix a set of integers $S = \{s_1, \ldots, s_n\}$ and an integer t that constitute an instance of the Subset Sum problem. Given even a linear function f, we can easily transform the integers into a binary vector x in an obvious way such that setting the threshold of the function f to t, and scaling it if necessary gives us an equivalence between a solution to the original problem and the solution to ALPSN. □

Let us end this section with a comment. It might be meaningfully asked if it is the number of nodes participating in the target set of links that is, in some sense, the source of hardness for the Attacker.

In the case of general graphs (as opposed to spatial ones), the following is true: if the likelihood function predicting links is increasing with the set of common neighbors and one targets all links connecting a group U of nodes (i.e., the target is actually a group of nodes U and that the target link set is the collection of edges between each pair of nodes in U, i.e., the target set of links is $H = \{\{x, y\} : x, y \in U\}$), the problem of hiding these links is in P [6, 26]. This can be seen via a greedy algorithm (formalized in terms of relevant matrices or induced subgraph matchings) and follows directly from results in the literature (see the results by [26] (Proposition 3.9) and by [6] (Theorem 3.4, independently)). The results do not directly transfer to the setting of spatial networks with the likelihood function based on the distance between nodes, but we conjecture that for some classes of geometric graphs (including geometric random graphs) the problem is also in P. We base the conjecture on the fact that, e.g., for the so-called geometric random graphs it is more likely for a pair $\{x, y\}$ to have a large common neighborhood, if the nodes x and y are closer to each other in Euclidean distance.

6 ALGORITHMIC RESULTS

So far, the results we have presented, were mostly negative. Now we demonstrate that if the edge likelihood function is convex, the problem can be solved with convex programming if we first fix the set of nodes that the adversary needs to manipulate. Furthermore, if this function is linear, we present approximation algorithms for:

- the case when the features are binary and we wish to hide only a single edge, and
- the case when the features are real-valued but we need to hide an arbitrary collection of edges.

Observe that convexity of f is a natural property: it means that f decreases at a slower rate as nodes grow farther apart.

Most importantly, in contrast to the hardness result for SALPSN, we demonstrate that, if we fix the set of nodes that the Attacker can manipulate to hide links between all the pairs of nodes in the target set, then we obtain a feasible solution:

Theorem 4. *The problem Fixed-*SALPSN *can be solved by convex programming, provided the edge likelihood function* f *is convex.*

Proof. Recall, that in this problem, for a fixed p, and a convex p-vector edge likelihood function, we are given a tuple $(G, \theta, \epsilon, H, F)$, where $F \subseteq V_G$ is a fixed subset of nodes v which we are able to manipulate in the sense of moving them to some vectors v' with a constraint that for each point $v \in F$ it has to be the case that

$$||v - v'|| \leq \epsilon.$$

We are asked to decide if there exists a set F' of such vectors $\{v'\}_{v \in F}$ such that in the graph $G' = (V'_G, E_G)$, where:

$$V'_G = V_G \setminus F \cup \{v' : v \in F\},$$

for each $\{x, y\} \in H$ it holds that

$$f(z(x, y)) \leq \theta.$$

It is clear that, if we denote by $dom(H)$ the domain of the target set of pairs, i.e., the set:

$$dom(H) = \{x \in V_G : \exists h \in H \; x \in e\},$$

then it is only reasonable to search for the set of nodes to displace inside $dom(H)$. That is, the task is to decide whether there exists the a set:

$$W = \{x'_1, \ldots, x'_m\} \subseteq dom(H) \cap F$$

such that, for each $i \leq m$, it holds that:

$$||x_i - x'_i|| \leq \epsilon$$

with the property that in the graph G' with

$$V'_G = V_G \setminus W \cup \{x' : x \in W\}$$

(with x' replacing x also in the domain of the target set) for each pair $\{x, y\} \in H'$ we have that

$$f(|x_1 - y_1|^p, \ldots, |x_D - y_D|^p) \leq \theta.$$

But then, by the choice of W and convexity of the edge likelihood function f, it is clear that the optimization version of the problem can be solved by the ellipsoid method for convex programming. Recall that in the optimization form, we are given a positive real number ϵ, a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$, a target set H of pairs of nodes $x, y \in V_G$ such that $\{x, y\} \notin E_G$, and we are asked to find

$$\min_{\delta \in \mathbb{R}^D} f(z(x, y + \delta)), \quad \text{s.t.} \quad ||\delta|| \leq \epsilon$$

for each pair $(x, y) \in H$ from the target set. We can do this iteratively, by solving a version of the problem for each pair $(x, y) \in H$. Without loss of generality, we may assume that the set F is contained in the domain of H. We assume that, for each pair $(x, y) \in H$, we are given a convex set K_x over which we minimize. We assume that each K_x is a superset of a ball of radius r and at every step of the search of the basic ellipsoid algorithm the algorithm works with an appropriate choice of the inner ball parameter. We also assume that all the values of f over K_x lie in the interval $[L_0^x, R_0^x]$. The details are presented in the pseudocode of the Algorithm 1.

By the general properties of the Ellipsoid algorithm (for convex optimization with constraints), i.e. that the access to the values

Algorithm 1 ALPS algorithm for convex f using the Ellipsoid algorithm

Input: a real number $\epsilon > 0$, a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$, a pair of nodes $x, y \in V_G$ such that $\{x, y\} \notin E_G$

for $(x, y) \in H$ **do**
  $L := L_0^x$ and $R := R_0^x$
  while $R - L > \frac{\epsilon}{2}$ **do**
    $\theta := \frac{L+R}{2}$
    $r' := \frac{r\epsilon}{2(R_0 - L_0)}$
    Apply the Ellipsoid algorithm to the set
$$K_x^\theta := \{z \in K : f(z) \leq \theta\}$$
  with parameter r'
    if Ellipsoid returns YES **then**
      $u := \theta$
      Set $\hat{z} \in K_x^\theta$ as the point returned by Ellipsoid
    else
      $L := \theta$
  Return the point $\hat{y}$ for which $\hat{z}$ is the value of the function z applied to the pair (x, y).

and the gradients of f is sufficient for the choice of r' to make the binary search algorithm to give correct answers, it follows that the above is the correct algorithm. □

Furthermore, there exist algorithms for some other restricted variants of ALPSN and SALPSN.

Theorem 5. *There exists a 2-approximation polynomial time algorithm solving the problem of* ALPSN, *if the features of the vectors are binary and if the edge likelihood function f is linear.*

Proof. Recall that we are given a real number $\theta \in [0, 1]$, and a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$, and a pair of nodes $x, y \in V_G$ such that $\{x, y\} \notin E_G$, and we are asked to compute a D-vector $y' = (y'_1, \ldots, y'_D)$ with $||y - y'||$ minimized such that $f(z(x, y')) \leq \theta$. If the features of the vectors are binary, then the result follows from Theorem 5.4 of [13] on ACRE-learnability. The reason the Lowd & Meek algorithm works is that ALPSN with the conditions as above is indeed an adversarial linear classification problem. Since the features of the vectors are binary, the Attacker wishes to switch some of these binary values, thus dislocating the vectors themselves, so as the linear function f outputs values bounded by θ. The fact that f is linear amounts to being expressible as
$$\sum_{i \leq D} w_i |z_i - z_i^a|$$
for some base instance $z^a = (z_i)_{i \leq D}$ and numbers w_i. This serves as an ideal minimum-cost instance and gives us the opportunity to use the property that linear functions define the loss of an instance as a weighted sum of differences in features, relative to the base instance. Let v be a D-vector and denote by C_v the set of features that have different values between z^a and v. Since f is linear, it is sufficient to consider the loss or cost of the D-vector v as the cardinality of the set C_v. The algorithm begins with an arbitrary instance y'' (i.e., such that $f(z(x, y'')) \leq \theta$.) The goal now is to find subsequent replacements of y'' that keep the value of the likelihood function $f(z(x, y''))$ below the threshold, while minimizing the norm of $x - y''$, up to the point where any swap in some value of any feature y_i would result in $f(z(x, y''))$ surpass θ. Since, by Theorem 5.4 of [13], Boolean linear classifiers are ACRE (adversarial classifier reverse engineering) 2-learnable (i.e., where costs are within a constant factor of 2 of the so-called minimal adversarial cost) under uniform linear cost functions. □

Theorem 6. *There exists a 2-approximation polynomial time algorithm solving the problem of* SALPSN, *if the features are real-valued.*

Proof. If the features of the vectors are real-valued, then the result follows by the use of a box-constrained L-BFGS to perform a line-search of approximation, along the lines of applying the algorithm from Section 4.1 in [20]. Recall that we are given a real number $\theta \in [0, 1]$, a spatial graph $G = (V_G, E_G)$ defined on $\mathbb{R}^D$, a positive integer $k < |V_G|$, a subset $S \subseteq V_G$ of vectors from the set of nodes of G, and a target set $H \subseteq V_G \times V_G$ of pairs of nodes such that for each $\{x, y\} \in H$ it holds that $\{x, y\} \notin E_G$. We are now asked to compute the set $A \subseteq S$ of at most k vectors $v' = (v'_1, \ldots, v'_D)$ with minimal $||v - v'||$ such that in the graph $G' = (V'_G, E_G)$, where
$$V'_G = V_G \setminus A \cup \{v' : v \in A\},$$
for each $\{x, y\} \in H$ it holds that $f(z(x, y)) \leq \theta$. We can assume that f has an associated continuous loss function, denoted as usual by $loss_f$. The task amounts to solving the following box-constrained optimization problem:

- for all the vectors in A minimize $||v - v'||$ subject to:
  (1) $f(z(x, y)) \leq \theta$, and
  (2) $|A| \leq k$.

Let $D(v, \theta)$ denote any function attaining an appropriate single minimizing vector v'. As the exact computation of $D(v, \theta)$ is hard, one can apply a box-constrained L-BFGS in order to approximate the minimizers. We therefore greedily find the least $\xi > 0$ such that
$$\xi ||v' - v|| + loss_f(v', \theta^*)$$
is minimized for each $v \in A$, where $\theta^* \leq \theta$. If f was convex and $k = 1$, then we could get an exact solution, but in the general case, we get a 2-approximation. □

7 CONCLUSION

We presented the first systematic study of adversarial link prediction in spatial networks, where the likelihood of a link is an increasing function of weighted distance between attribute vectors associated with the incident nodes. In this problem, the adversary aims to prevent positive identification of a set of links by manipulating the attributes of a subset of nodes on the network. We showed that this problem is NP-hard (and inapproximable) in general, and when features are binary, it is NP-hard even if there is a single target link that the adversary wishes to hide. However, if the features are real-valued and we fix the set of nodes that the adversary can manipulate, the problem can be solved using convex programming. Furthermore, the single-target-link case with binary features admits a 2-approximation in polynomial time, as does the general case with real-valued features.

ACKNOWLEDGMENTS

Tomasz Michalak and Michał Tomasz Godziszewski were supported by the Polish National Science Centre (grant 2016/23/B/ST6/03599). Yevgeniy Vorobeychik was partially supported by NSF (IIS-1905558, IIS-1903207) and ARO (W911NF1810208, W911NF1910241).

REFERENCES

[1] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. 2011. Node classification in social networks. In *Social network data analytics*. Springer, 115–148.

[2] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84 (2018), 317–331.

[3] M. Cerioli, L. Faria, T Ferreira, and F Protti. 2011. A note on maximum independent sets and minimum clique partitions in unit disk graphs and penny graphs: complexity and approximation. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications* 45, 3 (2011), 331–346.

[4] Palash Dey and Sourav Medya. 2019. Covert Networks: How Hard is It to Hide?. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS, Montreal, Canada, 628–637.

[5] Palash Dey and Sourav Medya. 2019. Manipulating node similarity measures in networks. *arXiv preprint arXiv:1910.11529* (2019).

[6] Palash Dey and Sourav Medya. 2020. Manipulating Node Similarity Measures in Networks. In *International Conference on Autonomous Agents and MultiAgent Systems*. 321–329.

[7] Michał T. Godziszewski, Paweł Batko, Piotr Skowron, and Piotr Faliszewski. 2021. An Analysis of Approval-Based Committee Rules for 2D-Euclidean Elections. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 6 (May 2021), 5448–5455. https://ojs.aaai.org/index.php/AAAI/article/view/16686

[8] Michał T. Godziszewski, Marcin Waniek, Yulin Zhu, Kai Zhou, Talal Rahwan, and Tomasz P. Michalak. 2021. Attacking Similarity-Based Sign Prediction. In *21st IEEE International Conference on Data Mining Security (ICDM)*. IEEE, 1367–1372.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

[10] Xiuzhen Huang and Jianer Chen. 2006. On PTAS for planar graph problems. *International Federation for Information Processing Digital Library; Fourth IFIP International Conference on Theoretical Computer Science- TCS 2006;* 209 (01 2006). https://doi.org/10.1007/978-0-387-34735-6_24

[11] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. 2018. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1666–1674.

[12] Siqi Liu, Sidhanth Mohanty, Tselil Schramm, and Elizabeth Yang. 2021. Testing thresholds for high-dimensional sparse random geometric graphs. https://doi.org/10.48550/ARXIV.2111.11316

[13] Daniel Lowd and Christopher Meek. 2005. Adversarial Learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (Chicago, Illinois, USA) *(KDD '05)*. Association for Computing Machinery, New York, NY, USA, 641–647. https://doi.org/10.1145/1081870.1081950

[14] Linyuan Lü and Tao Zhou. 2011. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications* 390, 6 (2011), 1150–1170.

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

[16] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[17] Alan Mislove, Bimal Viswanath, Krishna P Gummadi, and Peter Druschel. 2010. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*. 251–260.

[18] Bojan Mohar. 2001. Face Covers and the Genus Problem for Apex Graphs. *Journal of Combinatorial Theory, Series B* 82, 1 (2001), 102 – 117. https://doi.org/10.1006/jctb.2000.2026

[19] Mathew Penrose. 2003. *Random geometric graphs*. Oxford University Press.

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6199

[21] L. Valiant. 1981. Universality Considerations in VLSI Circuits. *IEEE Trans. Comput.* 30, 2 (Feb. 1981), 135–140.

[22] Yevgeniy Vorobeychik and Murat Kantarcioglu. 2018. Adversarial machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12, 3 (2018), 1–169.

[23] Marcin Waniek, Tomasz Michalak, and Talal Rahwan. 2020. Hiding in Multilayer Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. AAAI, New York, USA, 1021–1028.

[24] Marcin Waniek, Tomasz P Michalak, Michael J Wooldridge, and Talal Rahwan. 2018. Hiding individuals and communities in a social network. *Nature Human Behaviour* 2, 2 (2018), 139–147.

[25] Tomasz Was, Marcin Waniek, Talal Rahwan, and Tomasz Michalak. 2020. The Manipulability of Centrality Measures-An Axiomatic Approach. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS, Auckland, New Zealand, 1467–1475.

[26] Kai Zhou, Tomasz P Michalak, and Yevgeniy Vorobeychik. 2019. Adversarial robustness of similarity-based link prediction. In *IEEE International Conference on Data Mining*. IEEE, 926–935.

[27] Kai Zhou, Tomasz P Michalak, Marcin Waniek, Talal Rahwan, and Yevgeniy Vorobeychik. 2019. Attacking Similarity-Based Link Prediction in Social Networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 305–313.