Hiding from Artificial Intelligence

Marcin Waniek
Who am I?

• **June 2017**
  Defended a PhD dissertation at **MIMUW**
  Thesis: *Hiding in Social Networks*
  Main supervisor: Piotr Faliszewski
  Auxiliary supervisor: Tomasz Michalak

• **July 2017 – February 2019**
  Post-Doctoral Fellow
  at **Khalifa University**
  Supervisor: Aamena Alshamsi

• **February 2019 – September 2023**
  Post-Doctoral Associate
  at **New York University Abu Dhabi**
  Supervisor: Talal Rahwan
Hiding from artificial intelligence

It is getting increasingly difficult to live without leaving digital traces...

...that can be scrutinized by AI algorithms.

The literature assumes that the responsibility lies with a central authority...

...which is prone to failure.

The New York Times

Cambridge Analytica and Facebook: The Scandal and the Fallout So Far

Revelations that digital consultants to the Trump campaign misused the data of millions of Facebook users set off a furor on both sides of the Atlantic. This is how The Times covered it.
The general idea of this line of research

How important is the evader?

Does the evader have any undisclosed relationships?

What is the evader’s political orientation?

The evader

The seeker
Existing literature

Our line of research

The evader

The seeker

The evader

The seeker
What am I going to be talking about?

- Hiding importance from **centrality measures**
- Hiding group membership from **community detection algorithms**
- Hiding undisclosed relationships from **link prediction algorithms**
- Hiding the origin of a social diffusion from **source detection algorithms**
- Hiding opinions from **stance detection algorithms**
Hiding from centrality measures
Centrality

Centrality measures – methods of evaluating the relative importance of nodes.

- **Degree centrality** *(the most important node is the one with the greatest number of friends)*

- **Closeness centrality** *(the most important node is the one who is close to everyone else)*

- **Betweenness centrality** *(the most important node is the one who controls the flow of information)*

- **Eigenvector centrality** *(the most important node is the one with important friends)*

\[
\begin{align*}
c_{\text{degr}}(v) &= |N(v)| \\
c_{\text{clos}}(v) &= \frac{1}{\sum_{w \in V} d(v, w)} \\
c_{\text{betw}}(v) &= \sum_{u, w \in V} \frac{|\{p \in sp(u, w): v \in p\}|}{|sp(u, w)|} \\
c_{\text{eig}}(v) &= x_v \\
\text{for } Ax &= \lambda^* x
\end{align*}
\]
# Centrality

**Centrality measures** – methods of evaluating the relative importance of nodes.

<table>
<thead>
<tr>
<th></th>
<th>Degree</th>
<th>Closeness</th>
<th>Betweenness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>4</td>
<td>1st</td>
<td>1st</td>
</tr>
<tr>
<td>2nd</td>
<td>4</td>
<td>2nd</td>
<td>2nd</td>
</tr>
<tr>
<td>3rd</td>
<td>3</td>
<td>3rd</td>
<td>3rd</td>
</tr>
<tr>
<td>4th</td>
<td>3</td>
<td>4th</td>
<td>4th</td>
</tr>
<tr>
<td>5th</td>
<td>3</td>
<td>5th</td>
<td>5th</td>
</tr>
<tr>
<td>7th</td>
<td>2</td>
<td>7th</td>
<td>7th</td>
</tr>
</tbody>
</table>

- Degree: Number of connections a node has.
- Closeness: Measure of how close a node is to all other nodes in the network.
- Betweenness: Number of shortest paths passing through a node.

**Graph**

- Nodes A, B, C, D, E, F, G, H are connected in a network.
- Nodes are ranked based on their Degree, Closeness, and Betweenness.
Independent cascade influence model

- The process begins with only the source node being active.
- Every edge in the network is marked with the probability of activation.
- Every newly activated node has a single chance to activate each of his neighbors.
- The influence of the source node on the network is measured as the expected number of activated nodes.
Linear threshold influence model

- Again, the process begins with only the **source node** being **active**.
- Every other node in the network gets assigned a **threshold** from the distribution on the \([0,1]\) interval.
- A node gets activated when the **percentage of active neighbors** reaches the **threshold**.
- Again, the influence of the source node is measured as the **expected number of activated nodes**.
Choose how to spend the budget, i.e., which edge(s) to **add** and which to **remove**

\[
\text{centrality}(v^*) = 0.9 \\
\text{influence}(v^*) = 2.5
\]

\[
\text{centrality}(v^*) = 0.5 \\
\text{influence}(v^*) = 2.4
\]

### Complexity of finding an optimal solution

<table>
<thead>
<tr>
<th>Centrality</th>
<th>Absolute values</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>$P$</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Closeness</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Betweenness</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Influence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent cascade</td>
<td>NP-hard</td>
<td>NP-hard</td>
</tr>
<tr>
<td>Linear threshold</td>
<td>NP-hard</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>

M Waniek, T Michalak, M Wooldridge, T Rahwan. *How members of covert networks conceal the identities of their leaders*. ACM TIST (2021)
Our heuristic ROAM (Remove One, Add Many)

Remove an edge between you and one of your neighbours

Add some edges between your neighbours

$v^*$

What if criminal organizations would use such evasion techniques?
Hiding in WTC 9/11 terrorist network

Original network
1st in Degree centrality ranking
1st in Closeness centrality ranking
1st in Betweenness centrality ranking
IC influence = 2.55
LT influence = 6.44

After one execution of ROAM
3rd in Degree centrality ranking
2nd in Closeness centrality ranking
5th in Betweenness centrality ranking
IC influence = 2.39
LT influence = 6.72

After two executions of ROAM
5th in Degree centrality ranking
4th in Closeness centrality ranking
11th in Betweenness centrality ranking
IC influence = 2.21
LT influence = 6.90

Building a network from scratch

• What if we do not want to reshape an existing network, but rather construct a new one from scratch?
• Assume we have a group of network leaders...
• ... and a group of followers.
• We want to connect them into a network so that:
  – there are no leaders in top centrality ranking positions,
  – the leaders can effectively communicate with the rest of the network.
The captains network

1. We start with a **group of leaders** connected into a **clique**.
2. To each leader we assign a group of **captains**.
3. We connect the captains into a **full k-partite graph**.
4. Each of the **remaining nodes** gets connected with one captain from each group.

In this network **every captain** is guaranteed to have greater degree, closeness and betweenness centrality than **any of the leaders**.

M Waniek, T Michalak, M Wooldridge, T Rahwan. *How members of covert networks conceal the identities of their leaders*. **ACM TIST** (2021)
Multilayer networks

Means of communication
- Facebook
- Email
- WhatsApp
Local centrality in multilayer networks

**Local approach** – apply standard centrality measure in each layer separately.
Global centrality in multilayer networks

**Global approach** – treat network as a whole. Requires adjustments in centrality definitions.

**Degree**

\[ c_{degr}(v) = |N_M(v)| \]

where \( N_M(v) = \{ w \in V : (v^\alpha, w^\alpha) \in E \} \)

**Closeness**

\[ c_{clos}(v) = \frac{1}{\sum_{w \in V} d(v, w)} \]

where shortest paths may run between occurrences in different layers

**Betweenness**

\[ c_{betw}(v) = \sum_{u,w \in V} \frac{|\{(v^\alpha, p) : v^\alpha \in p, p \in \Pi(u, w)\}|}{|\Pi(u, w)|} \]

i.e., we take into consideration the number of occurrences on a shortest path

Standard version for comparison: \( c_{betw}(v) = \sum_{u,w \in V} \frac{|\{p \in \Pi(u, w) : v \in p\}|}{|\Pi(u, w)|} \)

\( N_M(A) = \{B, C, D, E, F\} \)
Hiding in multilayer networks

Choose the layer of contact for each node

The evader

Nodes that evader wants to maintain contact with

The problem is **NP-complete**.

**Heuristic:** contact with densely connected group of friends in each layer.

We study hiding from centrality measures in temporal networks, where edges exist only at certain moments.

A time-respecting path is a path where contacts occur chronologically.

An equivalent of distance in temporal networks is latency.

The latency between $v$ and $w$ at time $t$ is the shortest time it takes to reach from $v$ to $w$ starting at time $t$ along time-respecting paths.
Finding an optimal way to hide from temporal centralities is NP-complete.

Instead, we tested a number of heuristic solutions.

Removing existing contacts is significantly more effective in avoiding detection than adding new contacts.

On the other hand, adding new contacts improves the influence.
Using Lasso regression analysis, we investigate what are characteristics of nodes that are successful in obscuring their central position.

The average intercontact time $\nu_m$ has a strong positive correlation with the evader’s ability to hide, suggesting it is beneficial for the evader to spread their contacts more uniformly over time.
Project idea #1 Temporal network of scientists

Research question
How important events in a scientist’s career affect their centrality?

Bedoor AlShebli
New York University Abu Dhabi
Hiding from community detection
Community detection algorithms

- The term **community** is usually understood as a group of closely cooperating individuals.

- **Community detection algorithms** divide the set of nodes of the network into communities.

- Such division is called a **community structure**.
Measuring the quality of community structure

- Intuitively, we want more edges within the communities than edges between the communities.
- A popular measure of community structure quality is modularity
  \[ Q(CS) = \sum_{C_i} \frac{|E(C_i)|}{|E|} - \left( \frac{\delta(C_i)}{2|E|} \right)^2 \]
  where
  - \( E(C_i) \) are the edges between the nodes \( C_i \)
  - \( \delta(C_i) \) is the sum of degrees of the nodes in \( C_i \)

\[ Q(CS) = 0.42875 \]
\[ Q(CS') = 0.08625 \]
Community detection algorithms

- **Betweenness** - iteratively remove edges belonging to many shortest paths
- **Greedy** - merge communities that provide greatest modularity gain
- **Walktrap** - based on a tendency of random walks to stay within the same community
- **Eigenvector** - recursively split nodes into two based on the eigenvector signs
- **Louvain** – merge locally optimal community into a single node
- **Infomap** - based on compressing a description of the probability flow
- **Spinglass** - interpreting each node as an atom in a magnet, assign community based on spin
Some people might prefer not to disclose membership of certain groups...

...e.g., minorities persecuted based on a ethnic background.

Community detection can also be used to infer other kinds of sensitive information.
Hiding from community detection

Choose how to spend the budget, i.e., which edge(s) to add and which to remove.

Additional requirement:
We want to maintain communication structure of the group.

Measure of concealment

1) **Spread out** across other communities

\[
\mu_1(C^*) = \frac{|\{C_i \in CS : C_i \cap C^* \neq \emptyset\}| - 1}{(|CS| - 1) \max(C_i \cap C^*)}
\]

2) **Join a larger community to hide in the crowd**

\[
\mu_2(C^*) = \sum_{C_i \in CS} \frac{|C_i \setminus C^*|}{n - |C^*|}
\]

Combined measure

\[
\mu(C^*) = \frac{\mu_1(C^*) + \mu_2(C^*)}{2}
\]

Our heuristic DICE (Disconnect Internally, Connect Externally)

- Every member of the community finds one new (randomly chosen) neighbour from outside the community.
- The members might also disconnect some edges inside the community.

Simulation results

Facebook fragment
(786 nodes, 14,027 edges)

Madrid bombing network
(70 nodes, 96 edges)

Scale free networks
(1000 nodes, 2994 edges)

Community detection algorithm used by the seeker

Betweenness, Greedy, Louvain, Walktrap, Eigenvector, Infomap, Spinglass

Hiding from link prediction
Link prediction algorithms

- **Link prediction algorithms** evaluate the likelihood of existence of a not-yet-discovered (or simply unknown) edge between a pair of nodes.

- **Similarity indices** are link prediction algorithms that assign a score to any pair of nodes that are not connected in the network.
<table>
<thead>
<tr>
<th>Local similarity indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common neighbors</td>
</tr>
<tr>
<td>$s_{CN}(v, w) =</td>
</tr>
<tr>
<td>Salton</td>
</tr>
<tr>
<td>$s_{Sal}(v, w) = \frac{</td>
</tr>
<tr>
<td>Jaccard</td>
</tr>
<tr>
<td>$s_{Jac}(v, w) = \frac{</td>
</tr>
<tr>
<td>Sorensen</td>
</tr>
<tr>
<td>$s_{Sor}(v, w) = \frac{2</td>
</tr>
<tr>
<td>Hub promoted</td>
</tr>
<tr>
<td>$s_{HP}(v, w) = \frac{</td>
</tr>
<tr>
<td>Hub depressed</td>
</tr>
<tr>
<td>$s_{HD}(v, w) = \frac{</td>
</tr>
<tr>
<td>Leicht-Holme-Newman</td>
</tr>
<tr>
<td>$s_{LHN}(v, w) = \frac{</td>
</tr>
<tr>
<td>Adamic-Adar</td>
</tr>
<tr>
<td>$s_{AA}(v, w) = \sum_{u \in N(v, w)} \frac{1}{\log(d(u))}$</td>
</tr>
<tr>
<td>Resource allocation</td>
</tr>
<tr>
<td>$s_{RA}(v, w) = \sum_{u \in N(v, w)} \frac{1}{d(u)}$</td>
</tr>
</tbody>
</table>

All considered indices are based in some way on the set of common neighbors.
To measure the quality of link prediction we use two measures, AUC and AP.

Area under ROC curve (AUC) - probability that similarity index assigns a greater score to a randomly chosen hidden edge than to a randomly chosen non-edge.

Average precision (AP) - average precision \( \left( \frac{TP}{TP+FP} \right) \) of a family of classifiers based on the ranking returned by the similarity index.

\[
AUC = \frac{1}{2} \times \frac{2 + 3}{6} + \frac{1}{2} \times \frac{3}{6}
\]

\[
AUC = \frac{8}{12} = 0.66
\]

\[
AP = \left( \frac{1}{2} + \frac{2}{5} \right) / 2
\]

\[
AP = \frac{9}{20} = 0.45
\]
The unwarranted use of link prediction algorithms raises a lot of privacy-related issues. We might prefer to keep some of our relationships private.

Link prediction may arrive at erroneous conclusions, associating us with people we do not know.
Hiding from link prediction

Choose how to spend the budget, i.e., which edge(s) to add and which to remove

Area under ROC curve (AUC) = 0.8
Average precision (AP) = 0.7

Area under ROC curve (AUC) = 0.3
Average precision (AP) = 0.25

M Waniek, K Zhou, Y Vorobeychik, E Moro, T Michalak, T Rahwan. How to hide one’s relationships from link prediction algorithms. Scientific Reports (2019)
## Complexity of finding an optimal solution

<table>
<thead>
<tr>
<th>Link prediction algorithm</th>
<th>Hiding complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common neighbors</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Salton</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Jaccard</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Sorensen</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Hub promoted</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Hub depressed</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Leicht-Holme-Newman</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Adamic-Adar</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Resource allocation</td>
<td>NP-complete</td>
</tr>
</tbody>
</table>

Our heuristics

Closed Triad Removal (CTR)
Decreasing scores of hidden edges by removing edges

Open Triad Creation (OTC)
Increasing scores of other non-edges by adding edges

M Waniek, K Zhou, Y Vorobeychik, E Moro, T Michalak, T Rahwan. How to hide one’s relationships from link prediction algorithms. Scientific Reports (2019)
Here, we consider hiding in a telecommunication network of one of the major European cellular providers, consisting of 248,763 nodes and 829,725 edges.
The effects of size and density

Is it easier to hide your connections in small or large networks?
Is it easier to hide your connections in sparse or dense networks?

We perform simulations on randomly-generated networks of varying size and density and compare relative value of AUC and AP after hiding.

M Waniek, K Zhou, Y Vorobeychik, E Moro, T Michalak, T Rahwan. How to hide one’s relationships from link prediction algorithms. Scientific Reports (2019)
The effects of size and density

OTC-AUC  CTR-AUC  OTC-AP  CTR-AP

Size

Size

Size

Size

Degree

Degree

Degree

Degree

Adamic-Adar  Hub Promoted  Resource Allocation
Common Neighbours  Jaccard  Salton
Hub Depressed  Leicht-Holme-Newman  Sorensen

M Waniek, K Zhou, Y Vorobeychik, E Moro, T Michalak, T Rahwan. How to hide one’s relationships from link prediction algorithms. Scientific Reports (2019)
Is the hiding effectiveness actually affected by the **strategic choice** of edges to add/remove, or rather is it just a result of performing any changes in the network?

We perform simulations comparing the effects of **our heuristics** with the effects of **random changes** (given the same sets of edges allowed to be added/removed).

Random vs strategic hiding

Relative change in AUC

Adding edges

Removing edges

Relative change in AP

Adding edges

Removing edges

Similarity index used by the seeker

M Waniek, K Zhou, Y Vorobeychik, E Moro, T Michalak, T Rahwan. How to hide one’s relationships from link prediction algorithms. Scientific Reports (2019)
Project idea #2 Hiding from GNN link prediction

Research question
Is it possible to effectively hide from link prediction algorithms based on graph neural networks?
Hiding from source detection
We consider a process spreading in a social network, e.g., an infectious disease or a piece of information.

The process begins with only one node, the source, being active.

The process then spreads in the network over $T$ rounds according to some rules.

In this presentation we will focus on results for the Susceptible-Infected model, where during each round every active nodes activates susceptible neighbors with a given probability.
Source detection

• **Source detection** is the task of inferring which node was the source based on the state of the network after the diffusion took place.

• Information available is the **structure of the network** and the **state of each node**, i.e., whether it is active or not.

• We will focus on methods that produce a **ranking of all nodes**, with the leader of the ranking being the best candidate for the source.
Source detection algorithms

- **Random walk** – approximate the diffusion with random walks
- **Monte Carlo** – repeatedly start diffusion from each node and see how similar the outcomes are to the observed state
- Degree
- Closeness
- Betweenness
- Eigenvector
- Rumor

Compute **centrality** in the network induced by the infected nodes
Two ways of hiding

Given a budget $b$, which edges to add/remove so that there are at least $\omega$ nodes above the evader in the ranking of algorithm $\sigma$?

M Waniek, P Holme, M Cebrian, T Rahwan. *Social diffusion sources can escape detection.* iScience (2022)
### Computational complexity

<table>
<thead>
<tr>
<th>Source detection algorithm</th>
<th>Adding nodes</th>
<th>Modifying edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>P</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Closeness</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Betweenness</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Rumor</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Random walk</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td>Monte Carlo</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
</tbody>
</table>

M Waniek, P Holme, M Cebrian, T Rahwan. Social diffusion sources can escape detection. *iScience* (2022)
Hiding heuristics – adding nodes

M Waniek, P Holme, M Cebrian, T Rahwan. Social diffusion sources can escape detection. iScience (2022)
Hiding heuristics – modifying edges

In Barabasi-Albert (BA) networks the source is hidden by the very structure of the network, particularly in large and dense networks. However, in Erdos-Renyi (ER) and Watts-Strogatz (WS) networks the source is exposed.

The larger the highlighted triangle, the more effective the heuristic. In general, the most effective heuristics are those that connect the bots into a clique. Degree clique is the most effective heuristic here, the hexagon’s color corresponds to its performance.
Hiding the source of a real cascade

We also attempt to hide the sources of eight new Twitter hashtags in a retweet network consisting of 241,698 nodes and 366,539 edges.

M Waniek, P Holme, M Cebrian, T Rahwan. *Social diffusion sources can escape detection.* iScience (2022)
Project idea #3 Anomaly detection for hiding

Research question
Can anomaly detection algorithms be used to identify the nodes who perform strategic rewiring of the network?
Hiding from stance detection
Stance detection

• **Stance detection algorithms** allow to infer an opinion (either positive or negative) a person holds about certain topic based on this person’s publicly available social media data (in this study we focus our attention on **Twitter**).

• Notice that the opinion does not have to expressed directly, as the algorithms can read up on subtle clues **imperceivable to a human’s eye**.
The problem with stance detection

M Waniek, T Rahwan, W Magdy. *Hiding opinions from machine learning.* PNAS Nexus (2022)
The datasets we use

To explore these issues, we use two datasets:

- To train stance detection algorithms, we used a dataset of tweets with opinions they indicate towards atheism, feminism, and Hillary Clinton.

- A survey study with 1,143 participants we recruited via Amazon Mechanical Turk, with questions based on state-of-the-art SVM classifier.
We focused on three types of features: a word used in a tweet, an account followed, and an account mentioned in a tweet.

For each of the topic and each feature type, we identify the three features most strongly associated with the “against” stance, and the three most strongly associated with the “in favor” stance, according to the SVM classifier.

For each feature, we asked participants to specify the stance that it indicates towards the topic.

If a person is using one of the below words in a tweet, what would you assume is the stance of that person towards Atheism?

<table>
<thead>
<tr>
<th>Word</th>
<th>Strongly against</th>
<th>Against</th>
<th>Neither</th>
<th>In Favor</th>
<th>Strongly In Favor</th>
</tr>
</thead>
<tbody>
<tr>
<td>hope</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>faith</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>peace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>god</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>religion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>freethinker</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

M Waniek, T Rahwan, W Magdy. Hiding opinions from machine learning. PNAS Nexus (2022)
We now try to hide people’s opinion based on insights drawn from the SVM classifier.

We either remove the features that are most indicative of the real stance, or we add the features that are most indicative of the opposite stance.

We test these hiding methods against algorithms trained either on user’s contacts (the accounts they follow) or the user’s interactions (the accounts mentioned in their tweets).

Can algorithms help people hide their opinions from AI?

M Waniek, T Rahwan, W Magdy. Hiding opinions from machine learning. PNAS Nexus (2022)
Project idea #4 Hiding using XAI

Research question
Can Explainable AI be used to develop more effective, personalized hiding methods?
Summary of proposed topics

Idea #1 Temporal network of scientists

Idea #2 Hiding from GNN link prediction

Idea #3 Anomaly detection for hiding

Idea #4 Hiding using XAI

Marcin Waniek  www.mjwaniek.com  m.waniek@mimuw.edu.pl