A. Proof of Theorem 3

Theorem 3. *The derivative $\partial p_k / \partial w$ for $k \geq 1$ is given by:*

$$\frac{\partial p_k}{\partial w} = \left(\sum_{l=0}^{k-1} p_l Q^{k-1-l} \right) \frac{\partial Q}{\partial w} \tag{12}$$

Proof. By Induction:
p_k is defined by the recursive formula

$$p_k^T = p_{k-1}^T Q, \tag{13}$$

and applying the chain rule to Eq. (13) yields a recursive expression for the derivative as well:

$$\frac{\partial p_k}{\partial w} = \frac{\partial p_{k-1}}{\partial w} Q + p_{k-1} \frac{\partial Q}{\partial w}.$$

For $k = 1$ the derivative is given by

$$\frac{\partial p_1}{\partial w} = \underbrace{\frac{\partial p_0}{\partial w}}_{=0} Q + p_0 \frac{\partial Q}{\partial w} = \left(\sum_{l=0}^{0} p_l Q^{0-l} \right) \frac{\partial Q}{\partial w},$$

which anchors the induction.

Assume Eq. (12) is true for a fixed k then for $k + 1$ the derivative is

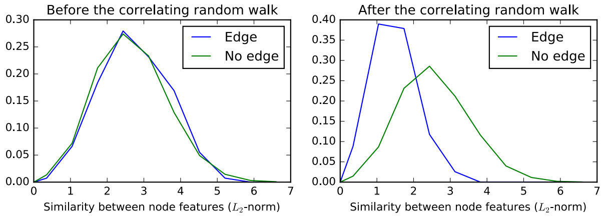
$$\begin{aligned}
\frac{\partial p_{k+1}}{\partial w} &= \frac{\partial p_k}{\partial w} Q + p_k \frac{\partial Q}{\partial w} \\
&= \left(\sum_{l=0}^{k-1} p_l Q^{k-1-l} \right) + \frac{\partial Q}{\partial w} Q + p_k \frac{\partial Q}{\partial w} \\
&= \left(\left(\sum_{l=0}^{k-1} p_l Q^{k-l} \right) + p_k \right) \frac{\partial Q}{\partial w} \\
&= \left(\sum_{l=0}^{k} p_l Q^{k-l} \right) \frac{\partial Q}{\partial w}.
\end{aligned}$$

This completes the induction and proves Theorem 3. □

B. Generating the Node Features

There are two properties the feature generation has to fulfill. First the Sybil network is assumed to behave adversarial and therefore tries to mimic the feature distribution of the honest network such that a simple feature-based detection does not work. To simulate this behavior the same underlying distribution is used to generate the features for the honest and for the sybil nodes. This ensures that any classification algorithm that is only based on the features will fail. Second the features should *not* be i. i. d., instead we want homophily within the network. This means that the features of two adjacent nodes to be more similar than the features of two non-adjacent nodes. To achieve this kind of feature distribution on the nodes the following two-step generation process is used.

Figure 7. Comparison of feature similarity of adjacent nodes (edge) and non adjacent nodes (no edge) before and after the correlating random walk.



In the first step the initial feature vector $x_v^{(0)} \in \mathbb{R}^d$ for each node $v \in V$ is chosen from the same multivariate random distribution, e.g., Gaussian distribution $\boldsymbol{x}_v^{(0)} \sim \mathcal{N}_d(\mu, \Sigma)$. In the second step the features in each dimension are correlated along the edges which is done with a short lazy random walk.

To get the transition matrix for the random walk, we first define the matrix

$$Q' = \alpha I + A(\tilde{G}) \,,$$

where I is the identity Matrix and $A(\tilde{G})$ is the adjacency matrix of the network graph $\tilde{G}$. The final transition matrix Q of the random walk is the normalized version of Q' where the row Q_v corresponding to the node v and is defined by

$$Q_v = Q'_v \frac{1}{\alpha + \deg(v)} \,.$$

With this transition matrix and the initial feature distribution $x^{(0)}$ the new correlated feature vector can be computed by applying the random walk for a few (k) iterations:

$$x_v := Q^k x_v^{(0)} = x_v^{(k)}$$

Figure 7 shows the distribution of similarity between adjacent nodes and non-adjacent nodes before and after the correlating random walk. Similarity is measured as the L_2-distance between the features of the nodes.

The edge features function $\psi_{u,v}$ simply stacks together the node features of the two adjacent nodes x_u and x_v .

C. Discussion on Fairness of Empirical Evaluation

We see our method (TSR) as a logical successor to Integro and SybilRank, as such, it was designed in a way to keep all the positive attributes of its predecessors while addressing their weak points to reflect more complex attacking scenarios. The reason why the detection performance is superior is that TSR (a) utilizes more information (i.e., labels and node features) and (b) it is more economical with the available information (solving an integrated optimization problem instead of a two step approach). We argue that the experiments are ...

- ... fair because each method is presented the same data. As in all comparisons, methods will use the available information in different ways, or, neglecting some parts of the available information (e.g. comparing semi-supervised with supervised methods).
- ... fair because each (state-of-the-art) method was designed with the goal to detect Sybil accounts. As such, they ought to be compared how well they solve this problem.
- ... unfair because we assume *perfect* victim detection performance for Integro. Hence, Integro looks better than actually expected in practice. As Fig. 5 shows, the detection performance for Integro deteriorates quickly if the preceding victim detection is not perfect (AUC of 0.9 instead 1.0). Also, to train a victim prediction, positive and negative labels are necessary and hence, Integro uses the same kind of information as TSR (P and N victim labels and node features).
- ... unfair because, due to a general lack of real data (i.e. labeled Sybil networks in real OSN), they were designed with our assumptions in mind (cf. Assumption 1) and our proposed method TSR was specifically designed to succeed in such scenarios. Throughout the paper, we argue that these assumption are actually more realistic than previous scenarios. However, a final verification in a real production environment is missing.

To sum up, we made sure that the empirical evaluation is as fair as possible given the constraints.