Of Strategies and Structures: Motif-based Fingerprinting Analysis of Online Reputation Networks

Matthias Wichtlhuber*, Sebastian Bücker*, Roland Kluge‡, Mahdi Mousavi†, and David Hausheer*
* Peer-to-Peer Systems Engineering, ‡ Real-Time Systems Lab, † Communications Engineering,
Technische Universität Darmstadt, Germany
Email: {mwichtlh|sbuecker|hausheer}@ps.tu-darmstadt.de,
m.mousavi@nt.tu-darmstadt.de, roland.kluge@es.tu-darmstadt.de

Abstract—**Reputation networks are an important building block of distributed systems whenever reliability of nodes is an issue. However, reputation ratings can easily be undercut: colluding nodes can spread good ratings for each other while third parties are hardly able to detect the fraud. There is strong analytical evidence that reputation networks cannot be constructed in a way to guarantee security. Consequently, only statistical approaches are promising. This work pursues a statistical approach inspired by the idea that colluding node's behavior changes the local structure of a reputation network. To measure these structural changes, we extend a graph analysis method originating from molecular biology and combine it with a machine learning approach to analyze fingerprints of node's interactions. We evaluate our method using an adaptive Peer-to-Peer (P2P) streaming system and show that a correct classification of up to 98% is possible.**

I. Introduction

Reputation networks play an outstanding role in the daily interaction with online services. For instance, Amazon's or eBay's star rating influence the buying decisions of millions of customers and Facebook's like button determines the importance of postings. In all of these settings, the ratings reflect an aggregated opinion on the posting, service or product of interest [1]. In many distributed networking systems such as email, ad hoc networks or P2P systems, reputation networks work similarly: nodes rate each other according to the service received in the past.

Despite their importance, it is trivial to undercut reputation networks. A very simple strategy is the creation of two identities by the same node spreading good ratings for each other (*sybil strategy*). The same strategy can be applied to two physically different nodes. In this case, the strategy is called *collusion strategy*. We use both terms interchangeably in the following. Seuken et al. [2] proved that even with tight monitoring and control from a central entity collecting reputation ratings, no mechanism can provide a 100% guarantee to prevent the mentioned strategies under realistic conditions.

Consequently, the only viable approaches remaining to solve the problem at a satisfactory level are statistical approaches, e.g. [3]. This work proposes a novel methodology to classify nodes in reputation networks by fraud strategy. For that purpose, we combine the motif-counting graph analysis method originating from molecular biology and a rule inference machine learning approach. The proposed algorithms are capable of reaching a classification accuracy of up to 98%.

We apply our methodology to a set of 4 different fraud strategies in an adaptive hybrid Content Delivery Network (CDN)/P2P live streaming system. The system resembles the architecture of recently emerging hybrid CDN/P2P deployments such as Akamai NetSession [4] with 32 million active[1] installations and a centralized control plane with sufficiently large capacity to compensate bandwidth bottlenecks of the peers. Using this system, we demonstrate the positive impact of excluding classified nodes on the overall Quality of Experience (QoE) of clients.

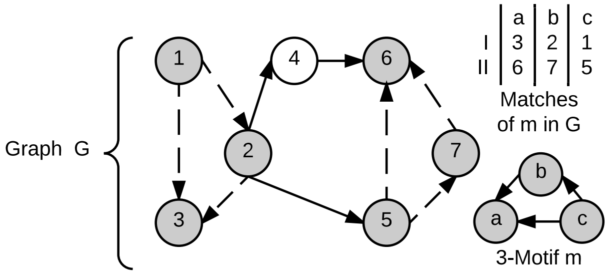
The remainder of this work is organized as follows: Section II details background information on reputation networks and graph analysis. Moreover, the section describes the architecture of the streaming system used as a case study for evaluation. Section III defines the methodology of motif-based fingerprinting analysis and details the integration of our approach into the streaming system. In Section IV, the approach is evaluated with respect to classification performance as well as impact on the streaming system's performance, i.e., the QoE of nodes in the network. Section V places this work in the context of related approaches. Finally, Section VI concludes the work and discusses extended use cases of our methodology.

II. Background

In the following, we define Reputation Networks formally, explain some details on Network Motifs and Transit, the streaming system used as a case study for evaluating the proposed methods and algorithms.

Reputation Networks: A *reputation network* is defined as a directed graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is a set of directed edges. Moreover, each edge has a weight defined by a weighting function $w : E \rightarrow \mathbb{R}_0^+$ mapping each edge to a weight. The weight of edge (v_i, v_j) defines a measure for the quality of service node v_j claims to

[1] According to Akamai's own statistics: http://wwwnui.akamai.com/gnet/globe/index.html, last visited 04/21/2016.



	a	b	c
I	3	2	1
II	6	7	5

Figure 1: Target graph G and 2 matches for 3-motif m.

have received from node v_i in the past. The term quality can be related to different metrics (e.g., delivery times of an online shop or the bandwidth delivered by a network node). The sum (or average, or minimum) of the weights of all outgoing edges of a node v_k defines the reputation of a node, e.g., $r_k = \sum_{v_i \in E \setminus v_k} w(v_i, v_k)$. Notably, this definition includes the possibility to lie on provided service. In particular, two nodes may collude to raise each other's reputation.

Network Motifs: A *network motif* is a small graph $m = (V_m, E_m)$ usually characterized by its number of nodes, e.g., $|V_m| = 3$. A *match* G' of a motif m in a target graph G is a subgraph of G that is isomorphic to m, i.e., a bijective mapping of the nodes in m to G' can be found such that the edges in both graphs are equivalent. More intuitively, a motif is a connected substructure in a graph as depicted in Figure 1.

The *frequency* $\mathcal{F}(m)$ of a motif in a target graph G is the number of matches of m in G. For identifying interesting motifs, the frequency of a motif in a target graph is compared to the average frequency $\overline{\mathcal{F}_r(m)}$ of the same motif in a sufficiently large set of randomly generated graphs with comparable properties (index r).

The z-score of a motif m is a metric for comparing $\mathcal{F}(m)$ with the average frequency $\overline{\mathcal{F}_r(m)}$ and is defined as:

$$Z(m) = \frac{\mathcal{F}(m) - \overline{\mathcal{F}_r(m)}}{\sigma_r(m)}, \qquad (1)$$

where $\sigma_r(m)$ denotes the standard deviation of the motif frequency in the set of randomly generated graphs [5]. More intuitively, the z-score measures the difference of motif occurrence in multiples of the standard deviation.

TRANSIT *streaming system:* TRANSIT is a hybrid CDN/P2P live streaming system serving to evaluate the proposed methods in a realistic scenario. It has a strong, centralized control and mechanisms to compensate bandwidth bottlenecks of peers with CDN resources. The system was first proposed in [6] as a fixed bitrate streaming system. Later on, it was extended with a reputation system [7] and adaptive video streaming capabilities using Scalable Video Coding (SVC) [8]. SVC is an extension of the H.264 video codec standard and allows splitting the video stream into layers of increasing quality, where a layer n can be decoded when all lower layers $[0 \ldots n-1]$ are present for decoding as well [9].

III. System Design

The main idea of this work evolves around the observation that subversive behavior in reputation networks changes the structure of the network interactions and thus the frequency of certain motifs. More precisely, a node trying to undercut a reputation scheme will change the structure of the network around itself, which is reflected in a local change of motif frequencies in the node's neighborhood. In the following, we first describe a methodology to identify a set of distinctive motifs. Using the distinctive motifs as features, we show how to learn a classification of nodes according to their applied subversion strategies.

A. Identifying Distinctive Motifs

The number of motifs grows exponentially with the motif size. At the same time, motif counting on a graph is based on the graph isomorphism problem, which is believed to be in $\mathcal{NP}$ [5]. Consequently, it is advisable to reduce the number of different motifs to be counted and their size as far as possible. This will reduce the number of motifs that have to be counted for feature extraction and minimizes the computational overhead for later processing.

The reduction of the number of motifs is done in three steps. First, four different simulation scenarios (S1 to S4) are defined on top of the TRANSIT simulation model [7], each with a composition of 80% *honest* nodes, i.e., nodes not undercutting the reputation scheme, and a share of 20% subversive nodes applying a certain strategy to undercut the scheme (see Table I for more details).

Namely the four strategies applied by the nodes are *freeriding*, the simple refusal of uploading any content, *reduced service*, a reduction of the upload capacity to a certain share of the total available upload capacity, *2-collusion*, two nodes boosting each other's reputation, and *n-collusion*, a node colluding with n other nodes from the neighborhood to boost each other's reputation, where n is chosen randomly from a normal distribution.

The two collusion strategies are implemented by introducing a malicious alternative back-end structure. All nodes willing to collude can register additionally with this back-end structure to find other nodes to collude with. The colluding nodes pick 2 or more of the other colluding nodes to send faked reputation values to, but may receive faked reputation values for themselves from two different nodes. That means, two colluding nodes cannot easily be identified by a back-and-forth connection between nodes.

The simulation approach allows to create a large number of periodical graph snapshots containing (a) the structure of the network and (b) a ground truth of the subversion strategy applied by each node in the network. Afterwards, all $Z(m)$-scores are calculated for all motifs in all graph snapshots over all scenarios. This allows constructing an $N \times N$ matrix $T_{m,\alpha=0.05}$ for each motif m, where N is the total number of subversion strategies to be classified and each element $t_{ij} = 1$, if there is a significant difference between $Z(m)$ in Scenario i and $Z(m)$ in Scenario j, and 0 otherwise. As

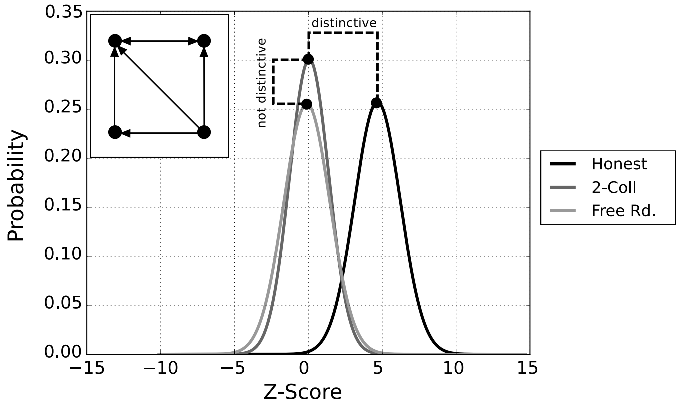


Figure 2: Z-score probability density of a 4-motif.

a statistical test for the significance of differences, Welch's two-sided unequal variance test [10] statistic for the average Z-score in both scenarios estimated as $\overline{Z(m)_i}$ and $\overline{Z(m)_j}$ with variance $var(Z(m)_i)$ and $var(Z(m)_j)$ is used. More formally, $T_{m,\alpha=0.05}$ is defined as follows:

$$T_{m,\alpha=0.05} = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & ... & N \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \vdots \\ N \end{array} & \left[\begin{array}{cccc} 0 & t_{12} & \dots & t_{1n} \\ t_{21} & 0 & \dots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & t_{n3} & 0 \end{array}\right] \end{array}.$$

Figure 2 illustrates the procedure for a 4-motif. Intuitively, the respective 4-motif is powerful for telling apart the honest nodes from nodes applying the 2-collusion strategy due to a minimal overlap of the respective Z-score distributions. However it is not distinctive for the 2-collusion and the freeriding strategy as the Z-score distributions have a large overlap.

Constructing T_m enables an identification of a subset S of all possible motifs being distinctive for the mentioned strategies. We chose a greedy strategy to compose S, i.e., all motifs are ordered by the number of strategies they can tell apart. In particular, we use the ranking function

$$r(m) = \vec{e}^T T_{m,\alpha=0.05} \vec{e}, \tag{2}$$

where e is a vector of length N with $e_{i,1 \leq i \leq N} = 1$. Afterwards, the most distinctive Motifs are added to S until all strategies can be distinguished. More precisely, we add those motifs with the highest rank $r(m)$ to the set of distinctive motifs S, until the condition

$$g(\sum_{m \in S} T_{m,\alpha=0.05}) \geq t \tag{3}$$

is met, where $g(A)$ maps a matrix A to its minimum element not considering the diagonal elements, i.e., $\min\{a_{i,j} | i \neq j\}$. The condition contains a parameter t to determine the minimal number of distinctive motifs per strategy combination. The parameter can be used to tune the accuracy that can be gained from using S for classification at the cost of the number of features, i.e., the size $|S|$ of the set.

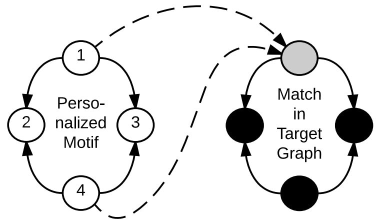


Figure 3: Personalized 4-motif and two possible matches of a substructure.

Applying this methodology to our system with $t = 1$, the five strategies (including the honest strategy) defined beforehand, and an initial set of 199 directed 4-motifs and 13 directed 3-motifs yields a set S of two 3 motifs and two 4 motifs as an output, i.e., only four motifs are sufficient to classify all strategies.

B. Feature Vector Extraction

After having identified the set S of distinctive motifs, S should be used as efficiently as possible. For that purpose we introduce *personalized motifs*. A personalized motif is a motif with the extension of vertex indices, which express that a node is not only participating in a certain motif in the graph, but also in which position within the respective motif.

Figure 3 shows a personalized motif and the possible matches to a subgraph structure. The given example illustrates two advantages of personalized motifs. First, personalized motifs inherently distinguish symmetric matches, i.e., in Figure 3, a match at position 1 can be distinguished from a match at position 4. Second, personalized motifs can account for the ”passiveness” or ”activeness” of a certain position in the motif's structure. In particular, in Figure 3, a match at positions 1 or 4 involves the delivery of service of the node to the two other nodes, while positions 2 or 3 only have incoming edges. We denote a personalized motif m with respect to a node n by the following syntax: $m_{n,p}$, where m denotes a motif in S and $1 \leq p \leq |V_m|$ denotes the position of n in the respective motif.

This work utilizes the FAst Network MOtif Detection (FANMOD) algorithm for motif counting [11]. Compared to other motif counting algorithms, FANMOD has a better runtime performance and enumerates each subgraph only once [12]. We modify the original algorithm with respect to two properties: first, the algorithm only counts the motifs contained in the set of distinctive motifs S. Second, the algorithm is extended to not only count all motif occurrences but also to return which node matches the motif at which position in order to count personalized motifs. Notably, the latter extension does not affect the complexity class of motif matching, as all possible graph isomorphisms for each tested substructure have to be enumerated anyways. However, the reduced number of motifs contained in S compared to the full number of possible

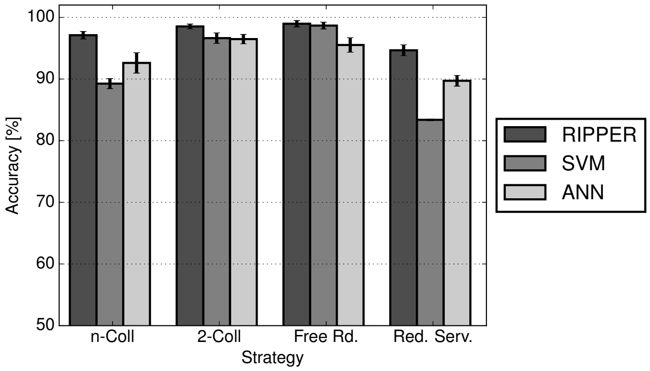


Figure 4: Performance of Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Support Vector Machine (SVM), and Artificial Neural Networks (ANN) machine learning algorithms per strategy.

motifs yields a reduction in overall run time depending on the size of S compared to the full number of possible motifs. In our scenario, this factor is as low as 2.3% for $t = 1$.

As a result of the feature extraction step, the following feature vector

$$\vec{v}_n = [v_1, \dots, v_k, v_{k+1}] \tag{4}$$

is generated for each node n in the system, where $v_1, \dots, v_k$ represents the motif count $\mathcal{F}(m_{n,p})$ for all personalized motifs from the distinctive motif set S and v_{k+1} is the difference of uploads and downloads n claims to have performed. The difference of uploads and downloads is added as an additional feature as it is easily available and cannot be captured by purely considering the edges, i.e., it cannot be expressed by pure motif counting.

C. Node Classification

Figure 4 compares the accuracy for a number of popular machine learning algorithms using $\vec{v}_n$ as the feature vector. The measurement was performed using the WEKA toolkit [13] with standard settings. As a performance metric, the established machine learning definition of accuracy

$$A = \frac{p_t + n_t}{p_t + n_t + p_f + n_f}, \tag{5}$$

where p_t and p_f refer to the number of true and false positives and n_t and n_f refer to the number of true and false negatives, is used.

Interestingly, the rule inference based RIPPER [14] algorithm outperforms the other algorithms in the set. Rule inference refers to the process of learning propositional logic rules over a set of features such that the classification accuracy metric is maximized. The algorithm utilizes a repeating grow and prune approach by adding propositional rules to a rule set until a classification precision of 100% is reached. Afterwards, rules with little classification precision gain are pruned to reach compact rules. This result is encouraging from a practical perspective, as a set of fixed, propositional logic rules can easily be implemented and checked against feature vectors. More precisely, the rules can easily be implemented with a decent amount of code in a real world system and do not incur large computational overhead to be tested for matches.

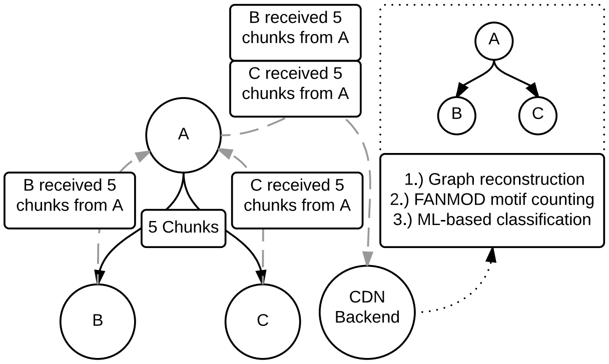

B received 5 chunks from A
C received 5 chunks from A
A
B received 5 chunks from A
5 Chunks
C received 5 chunks from A
B
C
CDN Backend
A
B
C
1.) Graph reconstruction
2.) FANMOD motif counting
3.) ML-based classification

Figure 5: Overview of streaming system integration.

D. Streaming System Integration

For the integration into the TRANSIT P2P streaming system, the tracker managing the list of present nodes in the network is extended for monitoring capabilities. As a goal, the CDN backend managing the streaming nodes should be capable of reconstructing the complete graph for motif analysis.

For that purpose, a receipt-mechanism based on the exchange of cryptographically signed receipts is used, i.e., if a peer A provides service to a peer B, B sends a signed receipt message certifying the amount of provided service to A by B. We refer to the amount of provided service certified by a receipt as *contribution* from now on. Notably, the usage of receipts aligns well with the general definition of reputation networks at the beginning of Section II and can be understood as a real-world implementation of the weighting function.

After having received the receipt, A forwards the receipt to the CDN backend node. The CDN collects all receipts from all nodes and processes the information in three steps as depicted in Figure 5. First, the aggregated information on the flow of data in the network is used to reconstruct the graph of claimed contributions. Each claimed contribution is held as an edge in the graph representation for a constant amount of time. Alongside with each edge, the contribution is annotated to the edge.

Afterwards, the modified FANMOD algorithm described in Section III-B is executed on the graph to extract the feature vector $\vec{v}_n$ of personalized motifs contained in set S. Third, the rules learned by the RIPPER algorithm are applied to the feature vector to classify nodes according to their strategy.

After having identified the strategy of a node, two measures can be taken to counteract subversive behavior. The classification can either be ignored, i.e., a subversive node can continue to stream without any counter measures, or the node can be excluded (*banned*) from the system.

Strategy	S1	S2	S3	S4	M5	M6	M7	M8	M9
Honest	80%	80%	80%	80%	80%	70%	60%	40%	20%
n-Coll.	20%	-	-	-	5%	7.5%	10%	15%	20%
2-Coll.	-	20%	-	-	5%	7.5%	10%	15%	20%
Free Rd.	-	-	20%	-	5%	7.5%	10%	15%	20%
Red. Serv.	-	-	-	20%	5%	7.5%	10%	15%	20%
# Nodes	200	200	200	200	200	200	200	200	200

Table I: Definition of evaluation scenarios. Scenarios names starting with an S indicate single strategy scenarios, whereas M-scenarios describe scenarios with multiple strategies present in the system at the same time.

			Predicted		
Actual	**Free rd.**	**n-Coll**	**Honest**	**2-Coll.**	**Red. Serv.**
Free Rd.	**64.81%**	3.71%	5.04%	26.35%	.09%
n-Coll	8.76%	**60.52%**	11.47%	16.44%	2.80%
Honest	**.02%**	**.21%**	**98.90%**	**.03%**	**.85%**
2-Coll.	39.07%	14.34%	7.80%	**36.29%**	2.51%
Red. Serv.	.80%	4.08%	18.97%	1.95%	**74.20%**

Table II: Confusion matrix of classification. A binary classification of honest and subversive strategies is highly accurate. Distinguishing between multiple subversive strategies turns out to be less precise.

IV. Evaluation

The simulation setup is designed to accurately resemble the conditions in a real-world system. It is composed of a *bandwidth*, *latency*, and *workload* model running inside the event based simulator PeerFactSim.KOM [15].

Three bandwidth classes for peers exist: *high*, *mid*, and *low*. Each classes' available asymmetric up-/download bandwidth and the share of peers in each class is modeled according to the annual OECD broadband report [16]. As a latency model, a normally distributed latency with $\mathcal{N}(\mu{=}100\text{ms}, \sigma{=}50\text{ms})$ between peers is used. In order to challenge the proposed algorithms, we use a flash-crowd workload (see Figure 6, grey solid line). The workload constitutes a worst-case scenario with a steep increase of nodes at the beginning resulting in a highly dynamic reputation graph. The workload is scaled to 200 present peers, as this is the smallest amount of nodes yielding significant differences of the results while still allowing for a decent simulation time.

Table I shows the six scenarios used throughout the evaluation. Scenarios S1 to S4 are single strategy scenarios as they were used to identify the relevant motif set S in the beginning, whereas M5-M9 are multi strategy scenarios with multiple subversive strategies present in the system at the same time. Especially M8/9 are challenging scenario as more than 60% of the nodes follow a subversive strategy, i.e., on average more than 60% of an honest node's neighbors try to cheat. Moreover, the M scenarios allow judging the performance of our approach in a graph having a high interaction of malicious nodes among each other, which is expected to considerably change the structure of the reputation network.

A. Machine Learning Performance

In this section, the stability of the classification's performance is evaluated with varying scenarios. All metrics used in this section are standard machine learning metrics composed from the basic p_t/p_f metrics referring to the number of true/false positives and n_t/n_f referring to the number of true/false negatives (see Section III-C). *Accuracy* is used as defined in Equation 5, *recall* is defined as $R = \frac{p_t}{t_p+f_t}$ and *precision* is defined as $P = \frac{p_t}{t_p+f_n}$.

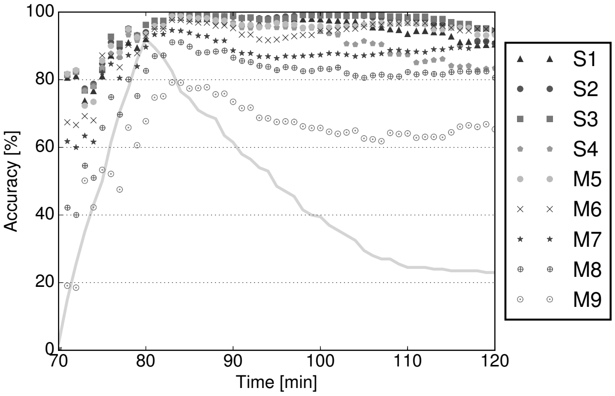


Figure 6: Stability of Accuracy over time (workload in grey).

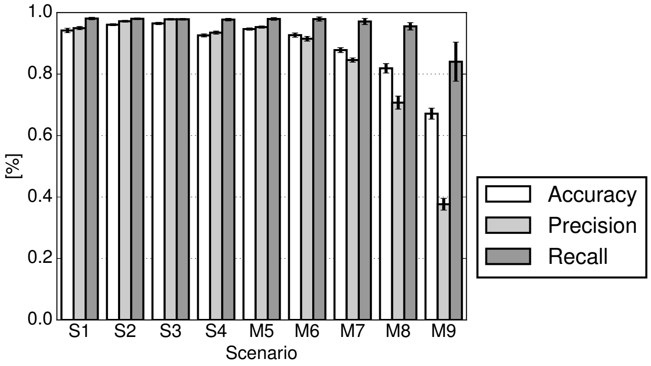
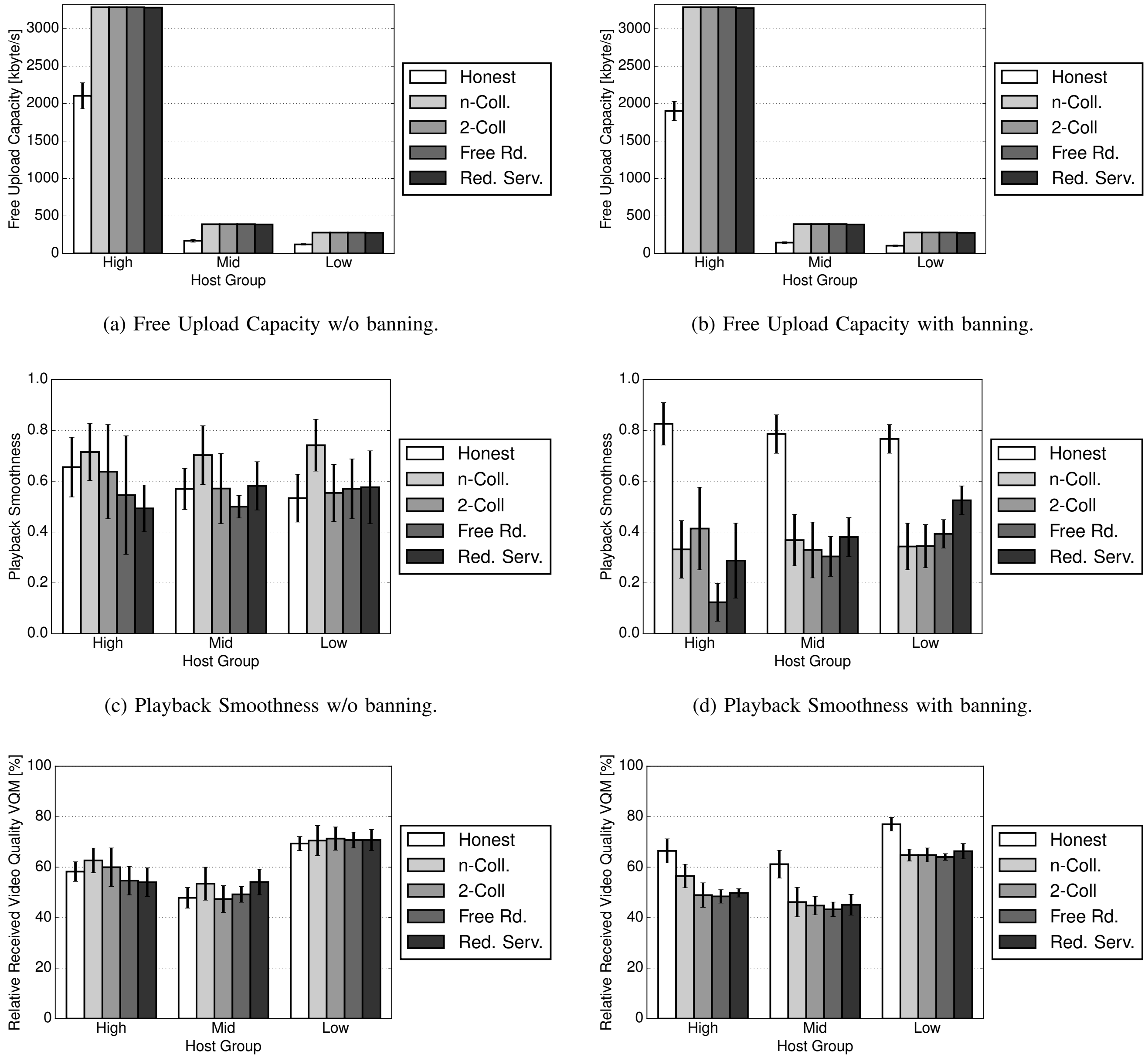


Figure 7: Accuracy, precision and recall in different scenarios. Accuracy and precision start to drop in scenarios with a high share of subversive nodes, while recall stays nearly constant.

Figure 6 depicts the accuracy depending on the time progress of the several scenarios. In the startup phase, accuracy is starting to increase quickly up to values above 90% across all scenarios except M9. The low accuracy at the beginning of some of the scenarios, e.g., M8/M9, is caused by a low number of present nodes at a comparably high ratio of subversive nodes. Consequently, false positives have a high impact in this phase. In fact this phase of the scenario is the most challenging part with little to no information on the past behavior of nodes. Contrary to that, the cool-down phase after the peak does not constitute a large relative drop of accuracy.

Figure 7 shows accuracy, precision, and recall for all scenarios. Notably, the accuracy does not drop below 90% up to scenario M5. Beyond this scenario, the loss in accuracy is related to a loss in precision, not in recall. More precisely, with more challenging scenarios, a higher number of false



(a) Free Upload Capacity w/o banning.

(b) Free Upload Capacity with banning.

(c) Playback Smoothness w/o banning.

(d) Playback Smoothness with banning.

(e) Relative Received Video Quality w/o banning.

(f) Relative Received Video Quality with banning.

Figure 8: Comparison of *Streaming System Performance Metrics* without (left column) and with banning (right column) of classified nodes split by bandwidth groups and applied strategy. The measurements are based on scenario M8.

positives is generated but a stable share of more than 95% of the subversive nodes is classified correctly to be subversive across all scenarios except M9.

Besides binary classification metrics, we also investigate the confusion matrix of our approach in Table II. Notably, the classification into honest nodes and subversive nodes classifies 98.90% of honest nodes correctly. Differentiating between the different classes of subversive nodes is more difficult. As an example, the classification of 2-Collusion nodes is correct in 36.29% of all classification attempts only.

B. *Streaming Performance*

In this section, the effects of classifying and removing subversive peers from the system are evaluated. For that purpose, we define three performance metrics describing the performance of the streaming system: the *free upload capacity* metric defining the amount of free upload capacity of a node, the *playback smoothness* metric defining the fraction of time a peer was able to play back the video stream compared to the length of the complete session without the delay to start playback and the *relative received video quality* metric, defining the relative visual quality a peer was able to receive from the system compared to the maximum visual quality. The latter measures the visual quality of the SVC video layer received on average compared to the visual quality of the highest possible SVC layer. As a metric for comparing visual quality, the Video Quality Metric (VQM) is used. VQM is a standardized, full-reference video quality model showing a high correlation to perceived visual quality of human subjects [17], [18]. The exact methodology is defined in [19].

Figure 8 shows the three metrics in the challenging scenario M8 with 60% subversive nodes, where Figures 8a, 8c, and 8e are obtained without banning subversive nodes and Figures 8b, 8d, and 8f show the performance of the system with the enabled motif fingerprinting algorithm and banning of classified nodes. The measurements are differentiated by bandwidth class as well as applied strategy.

The free upload capacity metric stays similar with banning (Figure 8b) and without banning (Figure 8a), showing that subversive nodes are at an advantage by having roughly twice as much spare capacity compared to honest nodes. At the same time, a comparison of the playback smoothness metric (Figures 8c, 8d) shows that classifying and banning of subversive nodes has a positive effect on the system's performance. In particular, without classification and banning the reachable playback smoothness is comparably low for all types of nodes. As opposed to that, Figure 8d shows a large performance drop for subversive nodes, while honest nodes can profit from system resources not wasted for subversive nodes compared to Figure 8c by gaining an advantage of more than 20% of playback smoothness.

A similar observation, albeit not that distinctive, can be made when comparing the relative received video quality in Figures 8e/8f. While in the case without banning, all nodes reach a comparable video quality regardless of their strategy, in the case with banning, honest nodes can reach a significantly higher video quality of up to 80%.

V. Related Work

We survey related work in three major categories: works investigating subversion strategies in *reputation networks in general*, works aiming at *social networks* and works aiming at *distributed systems*.

Reputation networks in general: Seuken et al. [2] make fundamental statements by formally analyzing sybil proof accounting mechanisms, which are conceptually similar to collusion proof mechanisms. We use both names interchangeably in the following. The authors prove that under reasonable assumptions, it is impossible to construct a completely sybil proof mechanism. However, a weaker form (K-sybil-proofness) can be achieved by only accepting a positive report on a node, if it is reported by K other nodes. Nevertheless, this approach would inherently slow down the update of the reputations. The work by Seuken et al. is a main motivation to think into the direction of statistical approaches to solve the problem from a practical perspective.

Social network analysis: A number of algorithms target statistical sybil detection in social networks motivated by the need to prevent spam. For instance, SybilGuard [20], SybilLimit [21], SybilInfer [22] and SumUp [23] assume some kind of clustering of subversive nodes and the fast mixing property, i.e., that a region outside a cluster of subversive nodes mixes inherently faster than within the cluster. These properties (amongst others) are used to detect subversive clusters. However, at least the fast mixing property was shown to be a poor feature for social networks by [3]. Moreover, these algorithms are tailored towards social networks and the set of features investigated are specific to this use case.

Distributed systems: From the distributed systems centric works, those focusing on hybrid CDN/P2P architectures with a strong, centralized control are most relevant to our work.

Piatek et al. propose Contracts [24], a scheme using two-hop reputations to score the contribution of a node. The scheme applies three approaches to mitigate collusion: (1) standard techniques limit the creation of identities per node, (2) the contributions of peers are checked to never exceed upload/download capacities, and (3) a global diversity weighting allows peers contributing to a more diverse set of IPs to reach a higher performance. All three methods have their drawbacks. (1) makes it difficult for new peers to join the system, (2) relies on the peer telling the truth on available capacity and (3) counteracts mechanisms to keep traffic local for ISP-friendliness.

Aditya et al. [25] propose an accounting scheme for hybrid P2P/infrastructure systems based on consistency checks, i.e., each transmission in the system is acknowledged and logged in a hash chain constituting a proof of work. The work focuses on the 32 million peers deployment of Akamai NetSession[2]. The hash-chaining approach requires every transaction to be logged and checked for inconsistencies. Additionally, the neighborhood of each peer is artificially narrowed to prevent colluding nodes from talking to each other. The approach presented in this paper may serve as a statistical extension on top to catch users applying sophisticated strategies like collusion or can replace the system entirely.

The work presented by Goncalves et al. [26] focuses on graph metrics to identify peers that are highly likely to provide a good service in a live-streaming setting. While performing an analysis of the correlation of service provided and graph metrics like the out-degree of a node using data traces from the SopCast network. The focus is not on security and the metrics are simple to be imitated for a node planning to undercut the scheme.

Our approach differs from the related approaches by relying on network structure analysis to reliably identify misbehavior in a reputation scheme. However, as opposed to [20], [21], [22] and [23], our work neither relies on clustering nor on fast mixing, but on the structural change of interactions in each node's neighborhood, as measured by a fingerprint of person-

[2] According to http://wwwnui.akamai.com/gnet/globe/index.html, last visited 2/3/2016.

alized motifs. Consequently, our approach is more generic and can be adapted to detect a multitude of subversive strategies by using the high expressiveness of motif fingerprints. At the same time, our approach differs from a practical perspective from the distributed systems centric works ([24], [25], [26]) by not relying on very fine-granular accounting, thus imposing a lower overhead. Additionally, our approach does not rely on an artificially narrowed neighborhood as in [25], which decreases the flexibility for other optimizations, e.g., topology optimization algorithms.

VI. Conclusions and Outlook

This work is motivated by the outstanding role of reputation networks in e-commerce, social networks, and distributed systems and the simplicity of undercutting their efficiency by applying cooperative, subversive strategies such as collusion. To counteract subversion, we developed a methodology to classify nodes according to their strategy. The methodology is inspired by the idea, that a node behaving subversively in a reputation network changes the structure of interactions in the neighborhood. To measure and evaluate structural changes of the reputation network, two methods are combined: the motif-counting methodology creates fingerprints of local substructures around a node, which can then be classified using a machine learning algorithm. The methodology is integrated into a hybrid CDN/P2P streaming system and shows a classification accuracy of up to 98%. Moreover, when using the proposed method to classify and ban subversive nodes, a significant increase in terms of QoE can be reached for honest nodes.

While this work uses a hybrid CDN/P2P streaming system as a case study, the developed methodology beyond that scope, as it can be applied to any reputation network to classify nodes according to their behavior. In particular, combining our approach with use case specific features can make the methodology useful to work on social graphs, virtual market places, crypto currencies, and the identification of email spam networks.

VII. Acknowledgment

This work has been supported in parts by the European Union (FP7/#317846, SmartenIT and FP7/#318398, eCousin) and the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI.

References

[1] A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[2] S. Seuken and D. Parkes, "On the Sybil-Proofness of Accounting Mechanisms," in *Workshop on the Economics of Networks, Systems, and Computation (NetEcon)*, 2011.

[3] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering Social Network Sybils in the Wild," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, pp. 2:1–2:29, 2014.

[4] M. Zhao, P. Aditya, A. Chen, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, B. Wishon, and M. Ponec, "Peer-Assisted Content Distribution in Akamai Netsession," in *ACM Internet Measurement Conference (IMC)*, 2013.

[5] H. Schwoebbermeyer, "Network Motifs," in *Analysis of Biological Network*, B. H. Junker and F. Schreiber, Eds. John Wiley and Sons, Inc., 2008.

[6] M. Wichtlhuber, B. Richerzhagen, J. Rückert, and D. Hausheer, "TRANSIT : Supporting Transitions in Peer-to-Peer Live Video Streaming," 2014.

[7] M. Wichtlhuber, S. Dargutev, S. Mueller, A. Klein, and D. Hausheer, "QTrade: A Quality of Experience Based Peercasting Trading Scheme," in *IEEE Conference on Peer-to-Peer Computing (P2P)*, 2015.

[8] M. Wichtlhuber, J. Rückert, D. Winter, and D. Hausheer, "How to Adapt: SVC-based Quality Adaptation for Hybrid Peercasting Systems," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015.

[9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H. 264/AVC Standard," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[10] M. Fagerland and L. Sandvik, "Performance of Five Two-Sample Location Tests for Skewed Distributions with Unequal Variances," *Contemporary Clinical Trials*, vol. 30, no. 5, pp. 490 – 496, 2009.

[11] S. Wernicke and F. Rasche, "FANMOD: A Tool for Fast Network Motif Detection," *Bioinformatics*, vol. 22, no. 9, pp. 1152–1153, 2006.

[12] P. Ribeiro, F. Silva, and M. Kaiser, "Strategies for Network Motifs Discovery," in *IEEE Conference on e-Science*, 2009.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, 2009.

[14] W. W. Cohen, "Fast Effective Rule Induction," in *International Conference on Machine Learning (ML)*, 1995.

[15] D. Stingl, C. Gross, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz, "PeerfactSim.KOM: A Simulation Framework for Peer-to-Peer Systems," in *IEEE International Conference on High Performance Computing and Simulation (HPCS)*, 2011.

[16] "OECD Broadband Report," OECD, Tech. Rep., 2012.

[17] M. H. Pinson and S. Wolf, "A New Standardized Method for Objectively Measuring Video Quality," *IEEE Transaction on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[18] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison," *IEEE Transaction on Broadcasting*, vol. 57, no. 2, pp. 165–182, 2011.

[19] M. Wichtlhuber, G. Wicklein, S. Wilk, W. Effelsberg, and D. Hausheer, "RT-VQM: Real-Time Video Quality Assessment for Adaptive Video Streaming Using GPUs," *ACM Multimedia Systems Conference (MMSys)*, 2016.

[20] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 576–589, 2008.

[21] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks," in *IEEE Symposium on Security and Privacy (SSP)*, 2008.

[22] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil Nodes using Social Networks," in *Network and Distributed System Security Symposium (NDSS)*, 2009.

[23] D. N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-Resilient Online Content Voting," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[24] M. Piatek, A. Krishnamurthy, A. Venkataramani, R. Yang, D. Zhang, and A. Jaffe, "Contracts: Practical Contribution Incentives for P2P Live Streaming," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2010.

[25] P. Aditya, M. Zhao, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, and B. Wishon, "Reliable Client Accounting for P2P-Infrastructure Hybrids," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2012.

[26] G. D. Gonc and A. Guimar, "Summary to Using Centrality Metrics to Predict Peer Cooperation in Live Streaming Applications," in *IFIP Conference on Networking*, 2012.