Smurf-based Anti-Money Laundering in Time-Evolving Transaction Networks

Michele Starnini*[1], Charalampos E. Tsourakakis*[1,4], Maryam Zamanipour*[1], André Panisson[1], Walter Allasia[2], Marco Fornasiero[2], Laura Li Puma[3], Valeria Ricci[3], Silvia Ronchiadin[3], Angela Ugrinoska[2], Marco Varetto[2], and Dario Moncalvo[2]

[1] ISI Foundation, via Chisola 5, Torino, 10126, Italy
{michele.starnini,babis.tsourakakis,maryam.zamanipour}@isi.it
[2] Intesa Sanpaolo, Corso Inghilterra 3, Torino, 10138, Italy
{walter.allasia,dario.moncalvo}@intesasanpaolo.com
[3] Intesa Sanpaolo Innovation Center, Corso Inghilterra 3, Torino, 10138, Italy
silvia.ronchiadin@intesasanpaolo.com
[4] Boston University, 111 Cummington Mall Boston, MA 02215, USA

Abstract. Money laundering refers to the criminal attempt of concealing the origins of illegally obtained money, usually by passing it through a complex sequence of seemingly legitimate financial transactions through several financial institutions. Given a large time-evolving graph of financial transactions, how can we spot money laundering activities? In this work, we focus on detecting smurfing, a money-laundering technique that involves breaking up large amounts of money into multiple small transactions. Our key contribution is a method that efficiently finds suspicious smurf-like subgraphs. Specifically, we find that the velocity characteristics of smurfing allow us to find smurfs by using a standard database join, thus bypassing the computational complexity of the subgraph isomorphism problem. We apply our method on a real-world transaction graph spanning a period of six months, with more than 180M transactions involving more than 31M bank accounts, and we verify its efficiency. Finally, by a careful analysis of the suspicious motifs found, we provide a classification of smurf-like motifs into categories that shed light on how money launderers exploit geography, among other things, in their illicit transactions.

Keywords: Anti-money laundering · Graph Mining · Subgraph Isomorphism · Data Mining

1 Introduction

Money laundering is an umbrella term, that captures the processing of criminal proceeds to disguise their illegal origin in order to legitimize the ill-gotten gains of crime [12]. While this definition may not include money related to terror

* Equal contribution

financing, which does not necessarily have a criminal origin, it is broad enough to cover all possible activities aimed at hiding the origin of illicitly gained assets. Money laundering has three well-defined stages: (i) placement, (ii) layering, and (iii) integration. Ebikake [10] describes in great detail how money launderers adapt to reality. In the placement stage, illicitly gained assets are introduced into the legitimate financial system while being cleansed of the most obvious traces of illegality. For example, forged documentation can be used to justify the money introduced as a legitimate receipt from the sales of real estate or interest in a business. In this phase, many different bank accounts across different banks can be used, or front companies that can belong even to high net-worth people. Once the money is deposited and its origin successfully explained, the placement stage is complete. In the layering phase, money-launders move around the money through a series of transactions that have no real purpose other than hiding the criminal nature of the money. For layering, money-launders may use banks in countries with poor law enforcement, or which do not cooperate with international financial authorities. Possible layering activities include investment in financial products which have good liquidity or which can be bought and sold easily with limited tracking (e.g. unlisted stocks and shares), real estate, fake loans that allow transfer of money to a business when in reality there is no loan, sending money overseas for education purposes, donations, and transferring money to shell companies [10]. Finally, in the integration stage, these assets are integrated into the legal economy and other assets can be legally purchased.

Despite the worldwide efforts against it, it is estimated that money laundering involves from 2% to 5% of the world's domestic product [29,13]. Fighting organized crime is of paramount importance for financial institutions: Failures in anti-money laundering (AML) controls may result in huge fines for financial institutions by national and foreign authorities. For example, Danske bank, the major Danish bank, faces a possible fine of around 2 billions euros for a money-laundering case of about 200 billions euros occurring through Danske's branch in Estonia, from 2007 to 2015 [15]. Recently, US authorities fined HSBC by 1.9 billion US dollars in a settlement over missing money laundering controls [19]. In order to comply with the current legislation, financial institutions generally follow several guidelines and recommendation, either official [12,3] or informal and internal best practices [21,20] that impose specific controls to be carried out on customers and on their activities/operations. These money-laundering controls have been historically implemented as a set of rules, such as fixed threshold flagging suspicious transactions, or transactions through countries considered at high risk, which are later manually inspected. Note that due to the heterogeneous financial services landscape and transaction means, there is no regulator guidance so technically detailed to play a standard-setting role. Each financial institution has thus the freedom and the responsibility to implement the controls with the techniques it deems most useful and efficient for the purpose. Such implementations are often made with deterministic approaches based on fixed rules and conditions to be calibrated over time and adapted to the various cases. Rule-based approaches are simple to implement, but suffer from several drawbacks: rules need to be constantly updated, and performance of single rules is very difficult to disentangle. Furthermore, rule-based systems perform badly on unstructured data and expert knowledge is needed to design rules. Finally, as a result of poor rule-based system design and data quality issues, classifiers for spotting alerts tend to aim for high recall by introducing a large number of false positives, that have to be manually inspected later on.

Therefore, there is a need for new data-driven tools for anti-money laundering able to overcome rule-based approaches. In this paper, we will focus on the central stage of money-laundering, i.e. layering, to detect suspicious transactions aimed at hiding the real origin and target of money transfers. A common method used by money-launders is to break down the amount of money to launder into smaller amounts and through various entities. This structuring technique is known as "smurfing", where smurfs are the financial actors (either companies or physical persons) responsible for organizing money transfers. These multiple intermediaries make small cash deposits or buy assets in amounts under a certain threshold, which is thought to be relevant and more likely to be reported by the banks to financial authorities. In this way, they try to avoid raising suspicions. The detection of smurfs in financial transactions is a pivotal task in the financial industry [37]. Smurfs naturally translate into specific subgraph structures within transaction graphs, where nodes are financial actors (i.e. bank accounts) and links represent money transfers between accounts. It is worth mentioning that, in a completely different field, smurf-like structures play an important role in security applications, e.g., [8].

Here, we focus on the two smurf-like motifs shown in Figure 1.

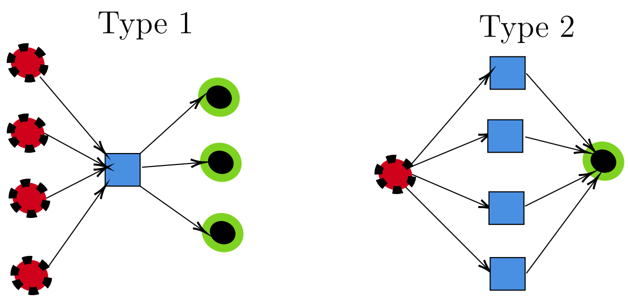

Fig. 1. Type 1 (left) and type 2 (right) smurf-like motifs. Source (red dotted circles), middle (squares), and target (green circles) are shown from left to right.

The first motif consists of a set of source nodes that send money to a middle node, who then sends that money to a set of target nodes. The second motif consists of a single source, sending money to multiple middle nodes, who then send money to a single target node. We refer to these two subgraphs, as motif type 1, and motif type 2. We outline that the number of source and target nodes in motif type 1, and similarly the number of middle nodes in motif type 2 may vary. While prior domain knowledge gives certain bounds on these node counts, searching for each possible motif instantiation using a state-of-the-art subgraph isomorphism algorithm is computationally expensive, and infeasible on large-scale transaction graphs. Our contributions include the following:

• We propose a pipeline that efficiently finds suspicious smurf-like subgraphs as shown in Figure 1. Our pipeline exploits the *velocity* of real-world money laundering transactions, and allows us to bypass the computational complexity lower bound of subgraph isomorphism. Perhaps surprisingly, our pipeline is based on a standard database join, and careful pre-, and post-processing filtering.

• We evaluate our pipeline on a large real-world transaction network with more than 184 million transactions using the financial services of a major Italian bank (from now on just referred as MIB). We observe that our pipeline allows us to find suspicious smurfs efficiently.

• We analyze the output motifs, and provide a systematic classification of suspicious motifs. For instance, we observe that certain suspicious motifs have a *u-turn* form. The source(s) and the target(s) are MIB bank accounts, whereas the middle node(s) is (are) non-MIB account(s), that may exist in high risk countries. Our classification sheds light into money launderers behavior, especially regarding how they exploit geography.

2 Related work

For a general overview of machine learning, and data-driven techniques used for anti-money laundering, see the recent survey by Chen et al. [7]. Here, we briefly review work that lies close to ours.

Flowscope is a novel tool for discovering dense flows from sources to untraceable destinations via many middle accounts that on purpose create chains to avoid getting flagged. The key intuition behind Flowscope is that large amounts of money need to be transferred through "dummy" accounts that serve as intermediaries before the dirty money reaches the final destination(s). The authors focus on detecting dense multi-partite subgraphs. While the Flowscope formulation and the proposed algorithm are important contributions towards AML, there exist important money laundering schemes that use few intermediary accounts, and thus do not induce dense subgraphs. Furthermore, Flowscope relies on the assumption that intermediate accounts have low balance, namely, they receive a certain amount and transfer it almost entirely. Real bank transaction data available to MIB indicate that intermediary nodes may transfer an amount only approximately similar to the one received from the source. For the aforementioned reasons, Weber et al. use graph convolutional networks [23] for fighting money laundering in bitcoin transactions [36]. Their method takes as input the transaction network, possibly node features, and some labels that are used to train the neural network. Lee et al. [24] propose a minimum description length approach to reorder the node ids in order to reveal all smurf-like subgraphs in a transaction network. However, many of these smurf-like subgraphs do not correspond to money laundering activities. Such false positives have an immense cost. The false positives are between 75% and 99% of the total alerts issued. This consumes bank resources, and places in inconvenient spot entities and people that abide by the law.

Isomorphism. Graph isomorphism is the problem of determining whether two graphs G_1 and G_2 are isomorphic. Formally, this is equivalent to determining if there exists a bijective mapping f from the set of the nodes of G_1 to the node set of G_2 such that any two nodes u, v of G_1 are adjacent in G_1 if and only if $f(u)$ and $f(v)$ are adjacent in G_2. The state-of-the-art algorithm is due to Babai and Luks [2], and despite the recent progress made by Babai it is not yet clear whether the problem is solvable in polynomial time or not [1].

The subgraph isomorphism problem asks whether a *pattern graph* H appears as a subgraph of a *target graph* G. This problem is known to be NP-complete as it generalizes well-known NP-complete problems including the *Maximum Clique*, and the *Hamiltonian Cycle* [14]. Formally, a subgraph isomorphism is an injective map f from the vertices of H to the vertices of G such that if two vertices u and v are adjacent in H, then $f(u)$ and $f(v)$ are adjacent in G. In our work, we focus on the variant of the subgraph isomorphism that aims to list the occurrences of the pattern H in the target graph G, rather than just decide if any occurrence of H exists in G. In general, searching for a motif with k nodes requires $O(n^k)$ time. Despite this asymptotic tight lower bound, there exist many algorithms that perform significantly better in practice compared to brute force. The classic algorithm is Ullman's backtracking algorithm with a look ahead function [35]. Given the importance of subgraph isomorphism in mining networks and graph databases, a lot of research has focused on efficient algorithm design. Notable algorithms include VF2 [9], GraphQL [16], QuickSI [32], GADDI [38], SPath [39]. ISMAGS is a recent algorithm that provides one solution per symmetry group [18]. This algorithm is particularly valuable when there is an exponential number of isomorphisms that are symmetrically equivalent. Another line of research has focused on designing efficient algorithms for special classes of graphs. A recent notable algorithm is due to Bressan et al. [6] that finds all occurrences of an induced k-vertex subgraph in a d-degenerate graph. Their algorithm runs in $O(f(k, d) \cdot n^\ell)$ where ℓ is the size of the largest induced matching in the motif to be searched. It is worth mentioning that subgraph isomorphism lies at the heart of frequent pattern discovery [22].

3 Dataset description

In this section we describe in detail the dataset of financial transactions we used in our experiments. The dataset encompass all wire transfers performed by the Head Office services of MIB in a period of six months, from August 1st, 2020 to January 31st, 2021 thus including SEPA [11] SCT and SWIFT-enabled [34] national and international wire transfers. Data were made available to the research team in a fully anonymized form respecting the strictest privacy and security requirements.

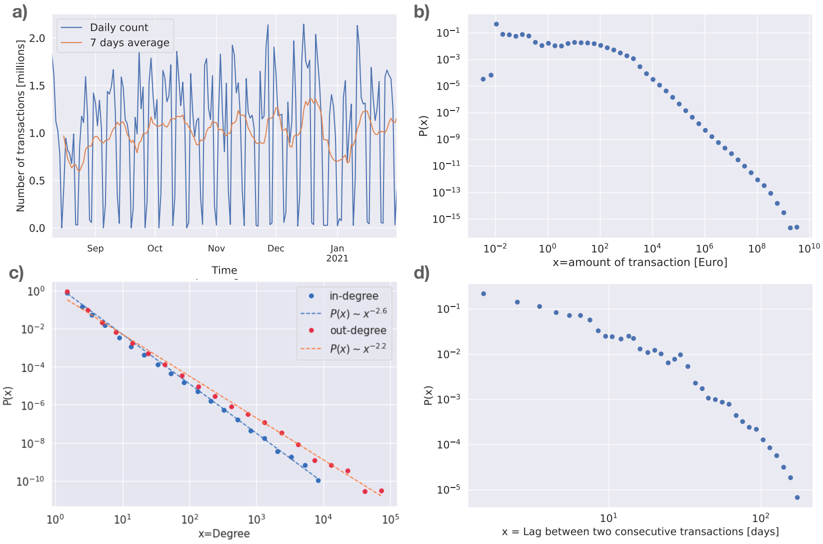

Fig. 2. Some empirical properties of the dataset: (a) Number of transactions in time, on a daily basis. (b) Probability distribution of the amount transferred in euros (log-log scale). (c) Probability distribution of the in- and out-degree (log-log scale). (d) Probability distribution of the time interval τ between two consecutive transactions involving the same sender and/or receiver (log-log scale).

The average monthly volume is close to 30 million transactions. Fig. 2(a) shows the number of transactions in time, aggregated on a daily basis. One can see that the number of transactions monitored is more than one million per day, excluded weekends. There is a considerable decrease in activity around the middle of August and during Christmas break. Each data entry includes a set of features, regarding both the sender/receiver parties and the transaction characteristics. For sender/receiver parties, data includes their anonymized bank account number, anonymized bank's BIC, party's and bank's country of residence (both at ISO alpha 2 level), and if the party is legal or physical person. For each transaction, features include timestamp, amount transferred, currency used, and transaction means (SEPA or SWIFT).

Figure 2(b) plots the empirical probability distribution of the amount transferred within the whole data set, in euros. One can see that most transactions regard an amount between few hundreds and few thousands euros. However, much larger amounts are present in the data set, up to a few billions euros. After a few thousands euros, the amount distribution decays as power-law function, indicating that very large transactions occur with very small probability, yet different than zero.

The dataset is naturally modeled as a time-evolving, directed multi-graph, a special instance of temporal networks [17]. In such graphs, nodes are a static collection of elements, edges are dynamic. In our dataset, nodes represent bank

Time period	N	E	# WCCs
Aug. 1st, 2020 -Jan. 31st, 2021	31M	184M	847K
Aug. '20	16M	26M	859K
Sep. '20	17M	31M	853K
Oct. '20	17M	33M	829K
Nov. '20	18M	32M	831K
Dec. '20	18M	33M	1073K
Jan. '21	17M	30M	869K

Table 1. Approximate number of nodes, edges, and weakly connected components of the entire dataset, and broken down by month.

accounts while edges transactions. Table 1 shows the number of nodes N, edges E, and number of weakly connected components (WCCs), for graphs reconstructed from the whole dataset and from single months. Out of the 847 092 connected components of the whole dataset graph, the giant component spans 29 693 858 nodes whereas the second largest contains only 304 nodes. We represent the information that a node i sent $w(i, j)$ financial amount to node j at time t as the quadruplet (i, j, w, t). We denote by n_{ij} and W_{ij} the number of transactions and the total amount of money transferred from node i to node j, respectively. The in-degree (out-degree) of node i, k_i^{in} (k_i^{out}), corresponds to the total number of counter-parties sending (receiving) money from (to) node i, over the whole time interval under consideration. The total amount of money sent (received) by node i, W_i^{out} (W_i^{in}) is obtained by summing all outgoing (incoming) transactions involving node i, $W_i^{out} = \sum_j W_{ij}$ ($W_i^{in} = \sum_j W_{ji}$).

Figure 2(c) shows the in-degrees and out-degrees of the whole transaction graph in the 6-month period in log-log scale. Both distributions are heavy-tailed, compatible with a power-law function $P(k) \sim k^{-\gamma}$, with similar exponents $\gamma_{in} \simeq 2.6$ and $\gamma_{out} \simeq 2.2$. This indicates that most actors are involved in transactions with few counter-parties, only very few parties engage with many others. However, a typical scale for the number of counter-parties is missing: in the data set there are present actors receiving money from up to one thousands different peers, and sending money to up ten thousands different parties. Nodes with large in- or out-degree typically correspond to companies that are not suspicious of money laundering activities; this could involve transferring money to a large number of employees, and receiving money from numerous business partners. As we will see in the following, we are interested in spotting actors interacting with relatively few counterparties. Fig 2(c) shows that, despite highly-connected nodes being a tiny fraction of the network, their presence is non-negligible. The scale-free form of the degree distribution suggests that pruning hubs might be effective in reducing the amount of data to monitor, as we will see. Indeed, removing a hub implies to remove all connected edges and this might affect the network's connectivity, possibly breaking the graph into disconnected components and thus making the motifs extraction easier [30]. This theoretical observation has been also specifically validated in empirical transaction networks

[31]. A different result would hold if the network had an homogeneous degree distribution (e.g. Erdos–Renyi graphs).

The time-varying graph representation allows us to take into account the activation dynamics of nodes and edges, corresponding to the dynamical features of sender/receiving parties [17]. Figure 2(d) shows the inter-transaction time distribution $P(\tau)$ between two consecutive activation of the same node, i.e. the time interval τ between two consecutive transactions involving the same sender and/or receiver, aggregated over the whole data set, expressed in hours. The inter-transaction time distribution $P(\tau)$ is heavy-tailed, indicating that the transaction dynamics follows a bursty behavior, as common in several human and natural contexts [33]: most transactions involving the same parties occur at small timescales, while large time intervals are increasingly less likely. Here, we are interested in spotting transactions occurring within a relatively small time interval, like a few days. Fig. 2(d) shows that, accordingly to the bursty nature of the transaction dynamics, these kind of transactions represent a large fraction of the total. For instance, 85% of consecutive transactions involving the same parties occur within 7 days. Therefore, an a priori filter aimed at pruning transactions occurring within large time-intervals would not be effective in significantly reducing the amount of data to monitor, as we will see in the following.

Note that one can generate synthetic time-evolving graphs with properties similar to the original data, by means of the probability density functions showed in Figures 2(b),(c),(d). The degree distribution $P(k)$ (Fig. 2(c)) can be exploited to generate a directed network by means of the so-called configuration model [5], allowing the possibility of multiple edges between nodes. The distribution of amounts (Fig. 2(b)) can be used to generate weights for each edge. The dynamics of the network can be taken into account by recent modelling frameworks developed to generate temporal networks, such as activity-driven networks [28]. Finally, the broad-tailed form of the inter-transaction time distribution (Fig. 2(d)) can be reproduced by using models for bursty temporal networks [27], in which the the link activation dynamics follows a non-Poissonian process.

4 Extraction of smurf-like motifs from transaction graph

In this Section we exactly define the problem of interest and propose a framework to efficiently solve it. Then, we show the motifs extracted by our method, classified from the perspective of anti-money laundering stakeholders.

4.1 Proposed pipeline

Problem definition. Figure 1 shows type 1 and type 2 subgraphs that we wish to extract efficiently from a large transaction graph. Observe that when there is one source and one target in motif 1, and one middle node in motif 2, the two motif types coincide. We are interested in finding a set of motifs as shown in Figure 1, that may have varying number of nodes, but involve few bank accounts (less than 20 in total), and are suspicious. The key characteristic we encode as suspiciousness is the velocity that the transactions within the motif take place. We state this as the following problem:

Problem 1. Given a time-evolving transaction network, find all motifs of type 1 and type 2, that involve at least 3 nodes, and at most k nodes, and all transactions take place within a time window of ΔT days.

Typically all the transactions from the source(s) to the middle node(s), take place before the transactions from the latter to the target(s). However, there can exist some asynchrony. From now on, let $\mathcal{S}, \mathcal{M}, \mathcal{T}$ be the sets of sources, middle nodes, and targets in motifs type 1, and type 2 respectively. Let $s = |\mathcal{S}|, t = |\mathcal{T}|$. We outline that existing anti-money laundering tools based on graph mining, including Flowscope [25] and AutoAudit [24], are not satisfactory formulations in our application. Perhaps, the most appropriate formulation is to cast the aforementioned problem as a subgraph isomorphism problem. Specifically, we can create a dictionary of motifs that we are interested in, and roll a time window spanning over the dataset to search for each motif using an efficient subgraph isomorphism algorithm, e.g., [35]. Unfortunately, this formulation is computationally expensive and does not scale well to large networks.

Proposed Framework. Before we delve into the details of our proposed framework, it is worth summarizing our key contributions. Our framework consists of a pipeline that involves few, computationally inexpensive steps, that pre-process the graph, perform simple database joins, and post-process the output, and is able to find suspicious subgraphs. Furthermore, by mining the output, we classify the motifs into categories that are of independent interest to anti-financial crime investigators and practitioners.

The pre-processing part removes nodes and edges that the bank knows or believes with high confidence that are not involved in money laundering. This part imposes the following constraints on the graph: edges whose weight is less than a certain threshold are removed, nodes with in-degree and out-degree above a certain threshold are removed. Transactions involving a small amount are indeed not suspicious for money laundering, as well as bank accounts with very large activity. Furthermore, we ensure that each path of length 2 involves at least one cross-border transaction. Since most bank accounts are Italian, this implies that in each three-nodes path at least one node is non-Italian. Table 2 shows an example (by using data from the month of November) of how the pre-processing steps greatly reduce the graph's size. For instance, even if nodes with in- or out-degree above 50 are just 0.2% of the transaction network, these account for almost 50% of edges. Altogether, the pre-processing constraints reduce the graph's size of about 1000 times.

Our search step is a standard graph database join that finds common neighbors between different pairs of nodes within the time window ΔT we are interested in. For instance for motif type 2, for a given ordered pair of nodes (u, v) we find the set of nodes that is out-going neighbors of u, and in-coming neighbors

Graph	N	E
Original	18M	26M
Min Edge Weight	1.22M	1.28M
Max k^{in},k^{out}	152K	125K
Min cross-border transactions	21K	46K

Table 2. Effects of the 3 pre-processing steps (highlighted in Table 3) on the graph's number of nodes N and edges E. At each pre-processing step the graph's size significantly decreases. As an example, we show data from November.

Pipeline	**Constraint**	**Values**
Pre-processing	Min Edge weight	Non-disclosed threshold
Pre-processing	Max k^{in}, k^{out}	50
Pre-processing	Min cross-border transactions	1
Motifs extraction	Motif 1	$s, t > 1$
Motifs extraction	Motif 2	$s = t = 1$
Motifs extraction	Max inter-transaction time ΔT	Non-disclosed threshold
Post-processing	Min total flow	Non-disclosed threshold
Post-processing	Flow ratio	Non-disclosed thresholds

Table 3. Values of constraints applied in the pipeline.

of v. The perhaps surprising finding is that this naive search algorithm that bypasses the constraint that the middle nodes should not have any edges between them (or induce few in general) is *automatically* satisfied by most of the output of the search step, due to our pre-processing step, and due to enforcing the velocity constraint. Furthermore, we find that one large motif may unpack into several smaller suspicious motifs, where the source, and target nodes remain the same, and the set of intermediary nodes may change over time.

Finally, motifs extracted are post-processed, in order to respect some additional constraints related to nodes and edges features. For each motif, the total incoming and outgoing flow can be computed, as the sum of the amount transferred through incoming and outgoing edges of the middle nodes, respectively. Similarly, the total flow transferred from source to target nodes is equal to the minimum between incoming and outgoing flows. Motifs must have a total flow transferred above a certain threshold, and the ratio between outgoing and incoming flows between a certain interval. The topological, dynamical, and additional constraints applied to extract the suspicious subgraph are summarized in Table 3. Note that for security reasons, we do not disclose the exact values used in the pipeline.

4.2 Results

Here we show the results of our pipeline. First, we compare the efficiency of our method with a state of the art algorithm for subgraph isomorphism search,

ISMAGS ($s = t = 1$)	**ISMAGS** ($s = t = 3$)	Proposed Method
88.0 secs	$> 1h$	84.0 secs
30.4 secs	$> 1h$	43.3 secs
94.0 secs	$> 1h$	87 secs
38.4 secs	$> 1h$	44.3 secs
15.6 secs	$> 1h$	26.9 secs
73 secs	$> 1h$	69 secs

Table 4. Running times in seconds of ISMAGS [18] for searching an induced path $i \to j \to k$ (column 1), a motif of type 1 with $s = t = 3$ (column 2), and our proposed method on searching *all* motifs of type 1 where $1 \leq s, t \leq 6$ over five different three-day windows (one per row). Running ISMAGS for searching a motif with $s = t = 3$ requires hours.

ISMAGS [18]. Then, we highlight a few interesting motifs extracted from the transaction network. Finally, we provide a systematic classification of motifs found according to the geography of countries involved.

Comparison to subgraph isomorphism. Table 4 compares the running time of ISMAGS [18] and our proposed method on five different time-windows of lenght ΔT for finding efficiently motif type 1. ISMAGS runs efficiently only when $s = t = 1$. Even when $s = 2$, ISMAGs may require more than an hour for certain time windows. When $s = t = 3$, for all considered time-windows, ISMAGS consistently requires time at the order of hours to find the motifs. This comes in sheer contrast to our proposed method, that forgets the constraint of finding induced subgraphs. Once we find a set of candidate subgraphs, our method checks which ones are isomorphic to the desired motif. We find all 36 possible instatiations of induced motifs of type 1 with $1 \leq s, t \leq 6$. The running time is always less than a minute and half. This happens since the time constraint we impose by looking into time-windows biases the dataset towards having this property, i.e., our proposed method finds induced subgraphs even if it is not explicitly searching for such. Furthermore, the number and size of subgraphs extracted is relatively small, so it is possible to check a posteriori if these subgraphs are induced.

Anomalous subgraphs. Figure 3 shows a subset of the output of our pipeline, colored accordingly to the geographical risk of each country involved: green for Italian bank accounts (considered non risky), orange for medium risk countries, yellow for low risk countries, and red for the high risk countries. Figure 3(a) shows a type 1 motif, with $s = 1, t = 8$. The middle node receives on day 1 a large amount of money from a German (DE) account, and then within the next couple of days distributes it in smaller amounts to 8 different bank accounts, all within Italy (IT). Figures 3(b), (c) show two more motifs of type 1 that involves multiple countries. In Figure 3(b), the middle node resides in Belgium (BE), while source and target nodes are in Italy and Croatia (HR), while in Figure 3(c) the amount is transferred entirely outside Italy. Figure 3(d) shows an induced

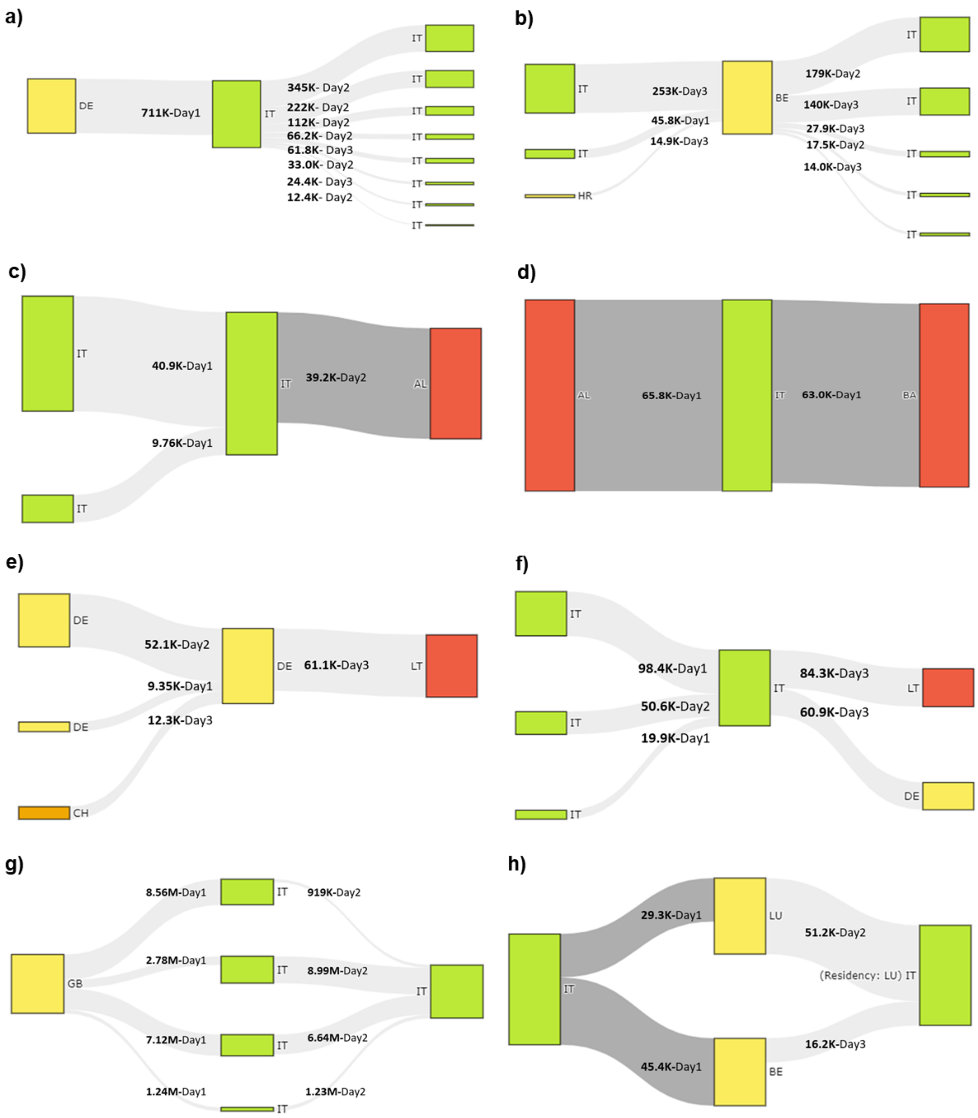

Fig. 3. Different groups of transactions extracted from the platform that are classified as suspicious due to their smurf-like behaviour. For each motif, nodes are colored accordingly to the geographical risk of each country involved: green for Italian bank accounts (considered non risky), orange for medium risk countries, yellow for low risk countries, and red for the high risk countries. Edge thickness indicates the amount transferred, also labeled on top of the edges.

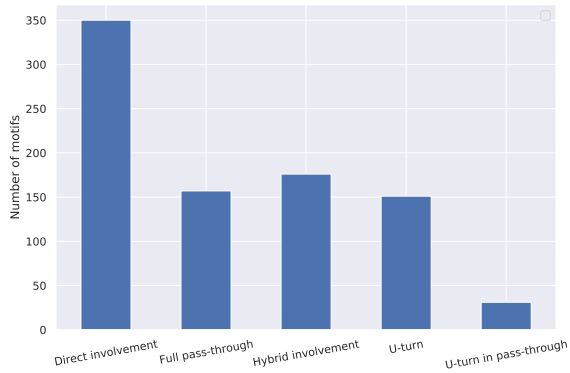

Fig. 4. Number of motifs extracted from the whole data set for each class, defined in the main text.

path of length 2 where the source and target nodes reside in Albania and Bosnia, respectively, while the middle node in Italy. Note that the two transactions take place within a single day. It is worth outlining that Albania is ranked as one of two of the countries most at risk from money laundering according to the Money Laundering and Terrorist Financing Index, published by the Basel Institute [4]. Similarly, Figures 3(e),(f) show two more suspicious motifs, involving Germany (DE), Switzerland (CH), Italy (IT), and Lithuania (LT) Figures 3(g), (h) show two examples of type 2 motifs: in Figure 3(e) the source node resides in Great Britain (GB), in Figure 3(e) both middle nodes are outside Italy while source and target nodes are in Italy.

Motif classification The motifs extracted can be classified according to the needs of further manual inspection, to be performed by anti-financial crime specialists. Figure 4 shows the distribution of the motifs detected according to our classification, which relies on the geography of the bank accounts involved. This classification is performed from the point of view of the financial institutions monitoring transactions (MIB in this case), but it can be generalized to any financial institution. The largest share of the motifs extracted can be classified as "direct involvement". In these motifs, MIB customers are engaged as pivotal figures (i.e. middle nodes), while being both beneficiary as well as ordering party of conspicuous transactions in the velocity schema. Another substantial share of motifs are classified as "full pass-through". In these motifs, MIB is supporting the payment delivery of others banks, so all the nodes involved are not MIB customer. Another case can be classified as "hybrid involvement", in which, while the pivotal middle node is external to MIB, some of the wire transfers start from or are directed to MIB customers. In this case, we have MIB nodes only in one side of the motif. An example of this class are motifs sketched in Fig. 3(c). Another important category is the one in which all source and target nodes belong to MIB customer base, while the middle node is external to the bank. This case is defined as “U-turn” in the literature [26]. The middle node is frequently located abroad in specific countries with inexplicable business reasons. Those cases are remarkably interesting since they present an enhanced “lack of economic purpose” feature, combined with the typical triggering red flag of “money laundering high risk geographies”. Finally, the last class is composed by motifs in which the “U-turn” is embedded in clusters of “pass-through” payments. In this case, the middle node is external to MIB, as well as a subset of sources and/or targets, thus we label it as a specific class “U-turn in pass-through”.

5 Conclusion

In this work we have proposed a practical pipeline for finding sets of transactions suspicious of money laundering. We show that our method scales gracefully with the size of the dataset, and bypasses the computational complexity lower bound of subgraph isomorphism by exploiting the high velocity characteristics of smurf-like transactions. Specifically, we show that simple database joins when combined with prior knowledge result in efficiency, which is crucial for real-time detection of such illicit activities. Furthermore, by studying the output of our pipeline, we provide a novel characterization of smurf-like motifs that is of independent interest to anti-money laundering practitioners and financial crime units. The latter provides insights on how money launderers use geography and the efficiency limitations of real-world transaction monitoring systems to perform their activities. An interesting open direction is learning more complex motifs that money launderers form by leveraging labeled transactions.

According to the perspective of anti-financial crime stakeholders, mainly interested into the practical monitoring power of the tools regardless the underlying mathematical approach, it is to be stressed that the “direct involvement” schema may be, at least in a partial manner, spotted with traditional rule-based algorithms based on counters and thresholds applied to wire transfers involving the customer base. These methods rely on relational databases only and are largely popular inside the banking industry. However, they present relevant limitations intrinsic to the fact that they do not consider the features of the whole transaction graph. Such limitations become almost a state of blindness for the cases “full pass-through”, “hybrid involvement”, “U-turn”, and “U-turn in pass-through”. These cases are to be taken into account when not only the customer base of bank but also counter-parties partially or totally external to it are to be considered. In this line of work, the presented results are a seminal contribution far from being maturely exploited in improving transaction monitoring systems.

Acknowledgements The research was conducted under a cooperative agreement between ISI Foundation, Intesa Sanpaolo Innovation Center, and Intesa Sanpaolo. The authors would like to thank Paolo Baracco, Piero Boccassino, Valerio Cencig, Raffaele Cosimo, Guido de Vecchi, Emmanuele Di Fenza, Maurizio Montagnese, Alessandro Raso, Mauro Ronzano and Luigi Ruggerone for their useful comments.

Data availability statement The data supporting the findings of this study is available from Intesa Sanpaolo upon request to Intesa Sanpaolo Innovation Center (innovationcenter@pec.intesasanpaolo.com). Please note that restrictions for data availability apply. Researchers interested in having access to data for academic purposes will be asked to sign a non-disclosure agreement.

References

1. L. Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.
2. L. Babai and E. M. Luks. Canonical labeling of graphs. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 171–183, 1983.
3. Banca d'Italia - Unita di Informazione Finanziaria per l'Italia. Rapporto annuale 2019. (12), 2020.
4. Basel institute on Governance. Basel aml index: 9th public edition ranking money laundering and terrorist financing risks around the world, 2020.
5. E. A. Bender and E. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, 1978.
6. M. Bressan and M. Roth. Counting homomorphisms, subgraphs, and induced subgraphs in degenerate graphs: new hardness results and complete complexity classifications. *arXiv preprint arXiv:2103.05588*, 2021.
7. Z. Chen, L. D. Khoa, E. N. Teoh, A. Nazir, E. K. Karuppiah, and K. S. Lam. Machine learning techniques for anti-money laundering (aml) solutions in suspicious transaction detection: A review. *Knowl. Inf. Syst.*, 57(2):245–285, Nov. 2018.
8. S. Choudhury, L. Holder, G. Chin, A. Ray, S. Beus, and J. Feo. Streamworks: a system for dynamic graph search. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1101–1104, 2013.
9. L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1367–1372, 2004.
10. E. Ebikake. Money laundering. *Journal of Money Laundering Control*, 2016.
11. European Payments Council (EPC). Sepa single euro payment area.
12. FATF, Financial Action Task Force. International standards on combating money laundering and the financing of terrorism & proliferation. 2012-2020.
13. F. I. G. From suspicion to action, converting financial intelligence into greater operational impact. *Financial intelligence group*, 2017. Available at https://www.europol.europa.eu/publications-documents/suspicion-to-action-converting-financial-intelligence-greater-operational-impact.
14. M. R. Garey and D. S. Johnson. Computers and intractability. *A Guide to the Theory of Np-Completeness*, 1979.
15. Guardian. Danske bank money laundering is biggest scandal in europe, 2018. Available at https://www.theguardian.com/business/2018/sep/20/danske-bank-money-laundering-is-biggest-scandal-in-europe-european-commission.
16. H. He and A. K. Singh. Graphs-at-a-time: query language and access methods for graph databases. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 405–418, 2008.

17. P. Holme and J. Saramäki, editors. *Temporal networks*. Springer, Berlin, 2013.
18. M. Houbraken, S. Demeyer, T. Michoel, P. Audenaert, D. Colle, and M. Pickavet. The index-based subgraph matching algorithm with general symmetries (ismags): exploiting symmetry for faster subgraph enumeration. *PloS one*, 9(5):e97896, 2014.
19. J. Y. Huang. Effectiveness of US anti-money laundering regulations and HSBC case study. 18(4):525–532, 2021/03/29 2015.
20. IntesaSanpaolo. Anti-money laundering rulebook, international branches, 2019. Internal document (restricted).
21. IntesaSanpaolo. Guidelines for combating money laundering and terrorist financing and for managing embargoes, 2019. Internal document (restricted).
22. C. Jiang, F. Coenen, and M. Zito. A survey of frequent subgraph mining algorithms. *Knowledge Engineering Review*, 28(1):75–105, 2013.
23. T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
24. M.-C. Lee, Y. Zhao, A. Wang, P. J. Liang, L. Akoglu, V. S. Tseng, and C. Faloutsos. Autoaudit: Mining accounting and time-evolving graphs. *arXiv preprint arXiv:2011.00447*, 2020.
25. X. Li, S. Liu, Z. Li, X. Han, C. Shi, B. Hooi, H. Huang, and X. Cheng. Flowscope: Spotting money laundering based on graphs. In *AAAI*, pages 4731–4738, 2020.
26. MAS – Monetary authority of Singapore. Red flag indicators for banks, 2015.
27. A. Moinet, M. Starnini, and R. Pastor-Satorras. Burstiness and Aging in Social Temporal Networks. *Phys. Rev. Lett.*, 114:108701, 2015.
28. N. Perra, B. Gonçalves, R. Pastor-Satorras, and A. Vespignani. Activity driven modeling of time varying networks. *Scientific Reports*, 2(1):469, 2012.
29. P. A. Schott. Reference guide toanti-money laundering andcombating the financing of terrorism. 2006. `https://openknowledge.worldbank.org/bitstream/handle/10986/6977/350520Referenc1Money01OFFICIAL0USE1.pdf;sequence=1`.
30. N. Schwartz, R. Cohen, D. Ben-Avraham, A.-L. Barabási, and S. Havlin. Percolation in directed scale-free networks. *Phys. Rev. E*, 66(1):15104, jul 2002.
31. A. Semeraro, M. Tambuscio, S. Ronchiadin, L. Li Puma, and G. Ruffo. Structural inequalities emerging from a large wire transfers network. *Applied Network Science*, 5(76), 2020.
32. H. Shang, Y. Zhang, X. Lin, and J. X. Yu. Taming verification hardness: an efficient algorithm for testing subgraph isomorphism. *Proceedings of the VLDB Endowment*, 1(1):364–375, 2008.
33. C. Song, T. Koren, P. Wang, and A.-L. Barabasi. Modelling the scaling properties of human mobility. *Nature Physics*, 6(10):818–823, sep 2010.
34. S.W.I.F.T. Society for worldwide interbank financial telecommunication.
35. J. R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.
36. M. Weber, J. Chen, T. Suzumura, A. Pareja, T. Ma, H. Kanezashi, T. Kaler, C. E. Leiserson, and T. B. Schardl. Scalable graph learning for anti-money laundering: A first look. *arXiv preprint arXiv:1812.00076*, 2018.
37. S. N. Welling. Smurfs, money laundering and the federal criminal law: The crime of structuring transactions. *Florida Law Review*, 41:287–343, 1989.
38. S. Zhang, S. Li, and J. Yang. Gaddi: distance index based subgraph matching in biological networks. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 192–203, 2009.
39. P. Zhao and J. Han. On graph query optimization in large networks. *Proceedings of the VLDB Endowment*, 3(1-2):340–351, 2010.