Themis: Detecting Anomalies from Disguised Normal Financial Activities

Rui Ding
School of Comp. Sci. and Eng.
Northeastern University
Shenyang, China
ruiding.neu@outlook.com

Xiaochun Yang✉
School of Comp. Sci. and Eng.
Northeastern University
Shenyang, China
yangxc@mail.neu.edu.cn

Bin Wang
School of Comp. Sci. and Eng.
Northeastern University
Shenyang, China
binwang@mail.neu.edu.cn

2023 IEEE International Conference on Data Mining (ICDM) | 979-8-3503-0788-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDM58522.2023.00016

***Abstract*—Financial supervision plays a pivotal role in society as it provides early warnings of financial activities and aids the government in detecting financial crimes. Detecting anomalous activities from normal financial activities is extremely challenging due to their disguise and complexity. However, existing anomaly detection methods in real-world financial scenarios typically suffer from some limitations: (a) Their formulations are overly simplistic to effectively identify complex anomalies; (b) Machine learning-based anomaly-detection methods lack enough training label, interpretability, and confidence, making it difficult to obtain approval from governments or financial institutions; (c) Many of them only focus on the financial transaction itself, ignoring the spatio-temporal characteristics of transaction and social relationships. To circumvent the challenges mentioned above, this paper proposes a novel anomaly-detection framework to detect the anomalies from disguised normal financial activities and infer clue chains for them. In particular, we are the first to formalize ten anomalies by reference to actual bank statements, and then three types of anomaly-detecting algorithms are proposed to discover these anomalies from financial activities. Next, we utilize an intelligent search algorithm to trace the most suspicious activities (clue chains) for institutions, improving the interpretability compared with learning-based methods. More importantly, we developed an anomaly-detection system, Themis, to detect these complex financial anomalies, which has been deployed in some real scenarios. The performance of Themis is demonstrated through some comprehensive extensive experiments and case studies on synthetic datasets and real bank statements.**

***Index Terms*—financial activities, anomalies, clue chains**

I. Introduction

Financial supervision plays a vital role in maintaining the stability of financial systems and supporting sustainable economic growth [1]–[3], providing early warning of abnormal financial activities. Given some financial activities, one of the major tasks in this field is to detect anomalous financial activities. Anomalous financial activities can be defined as a series of illegal transactions forbidden by financial institutions, such as money laundering and bridge loans [4], [5]. Explicit anomalies are prior anomalies that can be easily detected, such as exceeded amounts and limited log-in. They have been well inspected by rule-based methods [6]. Implicit anomalies are complex posterior-anomalous patterns hidden in normal financial activities, where accounts or transactions might be normal and only turn out to be anomalous when considered as correlated subgraphs. They are common core components of illegal financial cases. For example, Fig. 1 is an illicit bridge-loan case. In normal cases, Charlie should wait for the bank to release the loan after submitting the application to the bank. However, he chooses to get the loan from Bob (a bank staff) at high interest during this period. Bob takes advantage of his occupation to earn illicit income via an implicit anomaly($e_3 \rightarrow$ Bob $\rightarrow e_4$ with the brokerage as an implicit anomaly). A group of financial activities with this pattern may indicate a potential anomaly is taking place. However, there are no pioneer works to detect these anomalies since their anomalous patterns are complex and undetectable. In particular, we focus on detecting these anomalies from disguised normal financial activities.

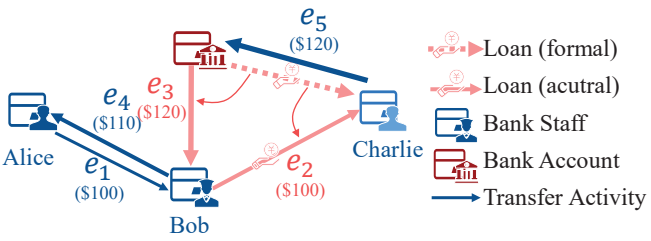


Fig. 1: A bridge-loan case: (a) Charlie has submitted a loan application for \$120 (e_5) to the bank, but there was a long waiting period; (b) Bob (a bank staff) promises to help Charlie get the loan as soon as possible, on the condition that charlie pays \$20 in interest. This means that Charlie needs to pay back \$120 as usual (e_5) but only get \$100. (c) Then Bob raises \$100 (e_1) from Alice to Charlie (e_2), promising to repay the principal and interest for \$110 (e_4) as long as the bank appropriates (e_3). In this case, Bob illegally obtained \$10.

Existing financial anomaly-detection methods can be categorized into traditional rule-based methods [6], [7] and deep learning-based methods [2], [8]–[11]. Previous rule-based approaches have mainly focused on predefined rules manually and detecting node- or edge-level anomalies based on these rules. Although these methods are simple and intuitive, they suffer from limited flexibility, failing to detect complex anomalies. Recently, deep learning-based methods use supervised learning techniques to predict anomalies of nodes (cards) or edges (transactions) on the financial graph [2], [8], [12] (out-

979-8-3503-0788-7/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDM58522.2023.00016

liers and dense subgraphs with unusual topological structure are not regarded as reasonable financial anomalies due to lack of transaction information.). However, they also suffer from many limitations in real-world scenarios: (1) *The scarcity of training labels is significant, posing a great challenge to supervised detection.* They formalize anomaly detection as a supervised prediction task, highly dependent on financial anomalies with precious and rare labels in real-world scenarios. (2) *The ability to detect complex anomalies is inadequate.* Existing financial anomaly formulations that only consider node- or edge-level do not support complex anomalies, e.g., bridge loans shown in Fig. 1. (3) *The interpretability of these anomalies detected is poor.* Although deep learning-based methods have already obtained good detection accuracy, it is hard to explain why abnormal. The explainability of financial anomalies is of great concern to financial institutions and only those explainable anomalies can attract the attention of institutions, thus playing the role of early warning.

Despite the success in detecting financial anomalies, most previous works only focus on detecting node- or edge-level anomalies from transactions. There is no academic pioneer to detect complex implicit anomalies from disguised normal financial activities. This is mainly because the detection of them presents several intractable challenges practically: (1) Difficulty in formalizing and detecting complex-diverse anomalies from disguised normal financial activities. Although there are many subgraph anomaly detection methods, they mainly focus on the topological structure of the graph (outlier and dense subgraphs), which is significantly different from our tasks "detecting anomalies from disguised normal financial activities". (2) Difficulty in improving interpretability and confidence of financial anomalies from an end-to-end supervised perspective. (3) Difficulty in formalizing and modeling heterogeneous information (social relationships and spatial-temporal features of financial activities) to enhance the accuracy of detection. The heterogeneous information mentioned above is not considered when detecting anomalies due to its complex structures and attributes, but this information is decisive in anomaly detection, as shown in Fig. 2. (4) Difficulty in tracing clue chains starting from anomalies to assist financial institutions in detecting anomalies. The discovery of clue chains can free the labor force from time-consuming and tedious verification tasks. However, there is no pioneer academic work to explore the tracing of clue chains in financial activities.

In order to effectively detect complex anomalies from disguised normal financial activities, we have worked closely with a financial institution to understand implicit anomalies and verify that they are the core components of illegal cases in financial activities. In this paper, we develop a novel uniform framework to detect anomalies, considering the heterogeneous and complex spatial-temporal and social features. Specifically, we are the first to formalize ten complex implicit anomalies by reference to actual bank statements where every account and transaction is legitimate. Then we design a family of detection algorithms to dynamically detect these anomalies under a practical uniform framework. In particular, we propose

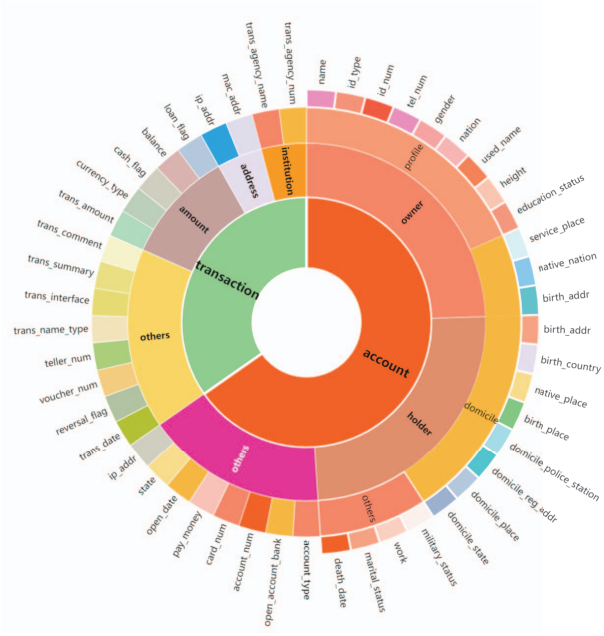


Fig. 2: Complex heterogeneous features in financial activities.

the clue chain tracing technology to trace potential clues based on the anomalies mentioned above, and then we evaluate and recommend the most suspicious clue chains to financial institutions. Here, we abandon learning-based methods due to their extremely poor interpretation and confidence. More importantly, we developed a practical anomaly-detection system, Themis, which has been deployed and applied in many real institutions. In experiments, we conduct comprehensive experiments to evaluate Themis's efficiency and effectiveness on actual bank statements and synthesis datasets.

The main contributions of this paper are as follows: (1) We design a novel practical framework to detect anomalous financial activities from disguised normal financial activities. (2) We are the first to formalize three patterns of normal financial activities and ten complex anomalies based on real financial scenarios, meanwhile considering necessary heterogeneous features, including social relations and spatial-temporal features of financial activities. (3) We design a family of detection algorithms to dynamically monitor complex anomalies in financial activities. (4) We propose a clue chain tracing technology to infer potential clue chains starting from the anomalies mentioned above and recommend the most suspicious clue chains to financial institutions. with high interpretability since the funding flows of clue chains are clear and transparent. These clue chains are highly interpretable since the funding flows of clue chains are clear and transparent. (5) More importantly, we developed an anomaly-detection system, Themis, to detect anomalies from disguised normal financial activities, which has been deployed in some real scenarios. Specifically, we evaluate the efficiency and effectiveness of Themis on actual bank statements and synthesis datasets.

II. Related Work

Financial supervision plays a vital role in maintaining the stability of financial ecosystems and supporting sustainable economic growth, providing early warning of abnormal activities [1], [2], [13]. There has been a long time of research efforts in this field [14]–[19]. Traditional anomaly-detection methods in financial supervision detect intuitive node(card)- or edge(transaction)-level anomalies by pre-defined rules [6], [7] or learning-based strategies [20], [21]. For instance, Kim et al. [6] formalize several anomalous rules to detect anomalies in financial activities, such as limited log-in time, limited log-in number, changed log-in location, and so on. These rules can only filter many node-/edge-level simplistic anomalies well. In addition, some researchers [2], [8] formalize the anomaly detection tasks as classification tasks and predict node(account)- or edge(transaction)-level anomalies in a learning-based manner [7], [22], [23]. They mainly learn the representation of nodes and edges with the help of attribute features and the topological structures of the graph [10], [24]–[27]. For instance, SeqFD [28] predicted fraud by aggregating statistical features of historical transactions within a time-based sliding window. In addition, Reddy et al. [29] modeled fraudulent transactions by introducing the temporal features and the structural features captured by GNN. Meng Shen et al. [30] propose a TSRGL framework, which uses R-GCN to learn the topology structure of the historical object-relation snapshot graph, and realizes the threat prediction of abnormal transaction behaviors. Cao et al. [12] construct financial transaction networks based on historical transactions, then learn users' topological structures in an unsupervised manner and predict the anomalies by tree-based classifiers. These methods all suffer from limited flexibility, only detecting the simple anomalies (node- or edge-level anomalies), and failing to inspect complex abnormal patterns in financial activities. The scarcity of training labels poses a huge challenge to learning-based methods since their detection accuracy is highly dependent on the volume of data with labels. What's worse, these methods all suffer from limited flexibility, only detecting the simple anomalies (node- or edge-level anomalies), and failing to inspect complex abnormal patterns in financial activities.

Recently, some learning-based methods regard outliers [15], [19] or dense subgraph [19], [31], [32] as significant abnormal patterns and mine these patterns from the perspective of structural characteristics. For instance, Zhang et al. [33] designed an anomalous subgraph autoencoder (AS-GAE) to detect outliers from the perspective of topological structure. Anting Zhang et al. [34] designed a subgraph embedding method to identify fraud communities that regard dense subgraphs as anomalies. All these works define anomalies from the perspective of structural characteristics and only utilize the topological information to detect significant outliers and dense subgraphs. However, outliers and dense subgraphs are not necessarily abnormal patterns in financial activity, and it is significantly unreasonable to detect financial anomalies based on topological structure in real scenarios, leading to many false positive samples. The vital factors influencing financial anomalies should be the transfer amount, the flow of funding indicating where the money comes from and what it is used for, social relations among operators, and so on.

In conclusion, existing financial anomaly-detection methods can only detect node-/ edge-level anomalies accurately, failing to detect complex anomalies from disguised normal financial activities. Although some works focus on subgraph detection, anomalies defined by them are significantly different from financial anomalies, and it is unreasonable completely to detect financial anomalies based on outliers and dense subgraphs. In addition, very few pioneers introduce heterogeneous information when exploring abnormal financial activities due to their complexity. They are necessary when detecting anomalies.

III. Problem Formulation

In this section, we formulate the problem we focus on. First, we introduce the definitions of the four inputs problem, namely bank account, person, financial activity, and social relation. They are the original data that can be exploited by the officers for the analysis of financial crime. Then we give the statements of the Malicious Financial Activity Detection Problem and explain its significance in reducing the cost of institutions.

Definition 1 (Bank Account)**.** *A bank account, denoted by v, is related to the following attributes: $owner(v)$ denotes the person who apply for the card, $regcity(v)$ is the registration city of card, and $type(v)$ is the type of account including "bank account", "enterprise account" and "personal account".*

Definition 2 (Person)**.** *A person, denoted by p, is related with the following attributes: $Accounts(p) = \{v_1^p, v_2^p, \cdots, v_n^p\}$ represents the bank accounts owned by p, $loc_t(p)$ represents p's location at time slot t, and $type(p)$ is the type of person, e.g., "citizens" and "bank staff".*

Definition 3 (Financial Activity)**.** *A financial activity from bank account v_i to account v_j at time slot t, denoted by $e_{i,j}^t$, is related with the following attributes: $value(e_{i,j}^t)$ represents the amount delivered in $e_{i,j}^t$, $actloc(e_{i,j}^t)$ is the activity location of $e_{i,j}^t$, and IP$(e_{i,j}^t)$ is the operating IP address of $e_{i,j}^t$. Correspondingly, $type(e_{i,j}^t)$ denotes the transaction type, including "transfer", "deposit", and "withdraw".*

Definition 4 (Relative Relation)**.** *A relative relation from person p_i to person p_j, denoted by $r_{i,j}$, is a binary variable. i.e. $r_{i,j} \in \{0, 1\}$. $r_{i,j} = 1$ means p_i is immediate relatives of p_j. In particular, $r_{i,i} = 1$.*

Malicious Financial Activity Detection Problem. Given a set of bank account $V = \{v_i\}$ and the related person set $P = \{p_i\}$, the financial activity set $E = \{e_{i,j}^t\}$, and the social relation set $R = \{r_{i,j} | r_{i,j} = (p_i, p_j), p_i, p_j \in P\}$, find abnormal financial activities from disguised normal financial activities and infer a clue chain (i.e. a sequence of financial activities among different accounts) for these suspect financial activities $e_{i,j}^t$ that can tell where the money $value(e_{i,j}^t)$ comes from and what it is spent on.

The practical significance of this problem is to assist financial institutions in showing clues of suspect activities. The clue chains are finally determined to be involved in financial crimes by the law officers (can be verified in practice). Generally, many anomalies are disguised in normal financial activities, which harms the financial ecosystems. Thus, it is important to define and detect abnormal financial activities.

IV. Modeling

We formulate a *financial activity network* to represent the bank transactions from historical databases, including transfers, deposits, and withdraws. Instead of using one vertex to denote one person, we use a bank account to indicate the basic smallest atomic unit in the system since the bank account is the core component of financial activities.

Definition 5 (Financial Activity Network, FAN)**.** *A financial activity network $G = (V, E)$ is a directed parallel graph recording financial activities. $V = \{v_1, \ldots, v_n\}$ is a set of bank accounts (e.g., debit card, credit card, and so on), and $E = \{e^t_{i,j} | e^t_{i,j} = (v_i, v_j)\}$ is a set of directed edges, representing a financial activity from v_i to v_j at time slot t. $E^{in}_{t_1,t_2}(v_i)$ and $E^{out}_{t_1,t_2}(v_i)$ are the set of in/out edges of node v_i whose timestamps t satisfy $t \in [t_1, t_2)$, respectively.*

(a) Transfer (NA_1) (b) Deposit (NA_2) (c) Withdraw (NA_3)

Fig. 3: The transaction type of financial activities.

To better illustrate the FAN, we use a card to represent a bank account and use a solid line to define a financial activity. Fig. 3 shows the three types of financial activities in an FAN.

- **Transfer Activity**: a direct edge $e^t_{i,j}$ from v_i to v_j, as shown in Fig. 3(a). It means money is transferred from one account v_i to another account v_j at time slot t, including transferring-in timestamp t_{in} and transferring-out timestamp t_{out}. If not specified, the default is a transferring-out timestamp.
- **Deposit Activity**: an anticlockwise loop $e^t_{i,i}$ from v_i to itself, as shown in Fig. 3(b). It means that the owner deposits cash with amount $value(e^t_{i,i})$ into his/her bank account at time slot t.
- **Withdraw Activity**: a clockwise loop $e^t_{i,i}$ from v_i to itself, as shown in Fig. 3(c). It means that the owner withdraws cash with the amount $value(e^t_{i,i})$ from his/her bank account.

Since G contains self-loops and parallel edges, it is not a simple graph. Additionally, each v_i only represents a bank account, while a person may have multiple bank accounts for transactions. These transaction records also reflect human behaviors socially. Therefore, to better depict financial activities, we need to construct another graph, say, a relative network, as an auxiliary graph for help.

Definition 6 (Relative Networks)**.** *A relative Network is a directed graph $S = (P, R)$, where $P = \{p_1, \cdots, p_m\}$ denotes the person set, and $R = \{r_{i,j} | r_{i,j} = (p_i, p_j)\}$ denotes the relative relation set from p_i to p_j. We use a solid line pointing from p_i to p_j to express such immediate relative relationships.*

There exists a many-to-one matching relationship between FAN G and relative network S. A person p_i in S may hold multiple accounts in G, while an account v_j only belongs to a legal person in S. We demonstrate such relationships in Fig. 4 and define $V(p_i) = \{v^i_1, \ldots, v^i_w\}$ as the account set owned by person p_i, and $owner(v_i)$ as the owner of v_i, where $owner(v_i) \in P$.

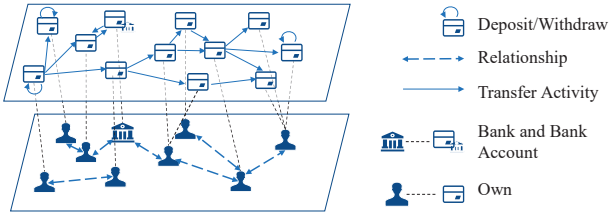


Fig. 4: Mapping between FAN and relative network.

With the help of FAN G and relative networks S, we can identify normal financial activities before detecting anomalies in a financial system. Obviously, the three basic activities mentioned in Figs. 3(a)-3(c) are normal activities, and we denote them as NA_1, NA_2, and NA_3, respectively.

Table I lists symbols and notations used in this paper, some of which are defined by their appearances.

V. Exploring Anomalies

Implicit anomalies disguised in normal financial activities pose a huge challenge to financial supervision since every single transaction is legitimate due to individuals' concealment. In this section, we attempt to explore these implicit anomalies and formalize ten implicit anomalies hidden in three normal financial activities. These implicit anomalies are the core components of financial anomalies and have potential risks leading to financial crimes.

Definition 7 (Sensitive-Region Anomaly, AA_1)**.** *A bank account v_i may exist anomaly if it is issued in sensitive regions C_s listed by the financial institution. Say, v_i has AA_1, if its registration city $regcity(v_i) \in C_s$.*

Definition 8 (Transaction-Address Anomaly, AA_2)**.** *A bank account v_i has a transaction-address anomaly AA_2 if v_i has a financial activity in one city but its owner is verified in another city at the same time, i.e. $\exists e^t_{i,j} \in E$, $actloc(e^t_{i,j}) \neq loc_t(owner(v_i))$.*

Definition 9 (Transaction-IP Anomaly, AA_3)**.** *A transaction-IP anomaly happens if multiple transfer activities $E' \subseteq E_{t,t+\Delta}$ from different account owners $P' \subseteq P$ are operated on the same IP address, i.e., $\forall e^{t_1}_{i,a}, e^{t_2}_{j,b} \in E'$, $IP(e^{t_1}_{i,a}) = IP(e^{t_2}_{j,b})$ and $|P'| > \tau_c$. Here, τ_c is the frequency threshold specified by the financial institutions.*

TABLE I: Primary notations

Notation	Description
$G=(V,E)$	An FAN with account set $V=\{v_1,\ldots,v_n\}$ and financial activity set $E=\{e^t_{i,j}\|e^t_{i,j}=(v_i,v_j)\}$.
$type(v)$	Types of account $v\in V$: $type(v)$ ={"bank account", "enterprise account", "personal account"}.
$type(e^t_{i,j})$	Types of financial activity $e^t_{i,j}\in E$: $type(e^t_{i,j})$={"transfer", "deposit", "withdraw"}.
$owner(v)$	The owner of the account $v\in V$.
$regcity(v)$	The registration city of the account $v\in V$.
$value(e^t_{i,j})$	The transfer amount of $e^t_{i,j}\in E$.
$actloc(e^t_{i,j})$	The activity location of $e^t_{i,j}\in E$.
$IP(e^t_{i,j})$	The operating IP address of $e^t_{i,j}\in E$.
$\mathcal{P}(v_i,v_j)$	A financial path $(v_i,v_1,\cdots,v_j)$ in the FAN.
$S=(P,R)$	A relative network, with people set P=$\{p_1,\ldots,p_w\}$ and relative relationship set R=$\{r_{i,j}\|r_{i,j}=(p_i,p_j)\}$.
$type(p_i)$	Types of person $p_i\in P$: $type(p_i)$ ={"citizens", "bank staff"}.
$loc_t(p_i)$	The location of the person p_i at time slot t.
E_{t_1,t_2}	The edge set of all activities within time interval $[t_1,t_2)$.
$E^{in}_{t_1,t_2}(v_i)$	The edge set of transferred-in activities of v_i within time interval $[t_1,t_2)$.
$E^{out}_{t_1,t_2}(v_i)$	The edge set of transferred-out activities of v_i within time interval $[t_1,t_2)$.
$\|E_{t_1,t_2}\|$	The number of edges in set E_{t_1,t_2}.
$\mathcal{C}$	Evidence Chain.
C_s	The sensitive region set.
λ	Upper bound on the legal ratio.
τ_c	A frequency threshold specified by financial institutions.
ϵ_v	A small monetary threshold, indicating kickbacks.
ϵ_t, ϵ_a	A time interval and amount threshold when tracing clue chains.
Δ	The threshold of a small time interval.

Definition 10 (Funding-Frequency-Fluctuation Anomaly, AA_4)**.** *A bank account v_i has a funding-frequency-fluctuation anomaly AA_4, if the frequency of recent transactions far exceeds the frequency of previous ones, i.e., $|E^{in}_{t-\Delta,t}(v_i)|+|E^{out}_{t-\Delta,t}(v_i)|\geq\lambda\cdot(|E^{in}_{t'-\Delta,t'}(v_i)|+|E^{out}_{t'-\Delta,t'}(v_i)|)$, where t' is the previous time stamp of t, and λ is a larger threshold.*

Definition 11 (Funding-Amount-Fluctuation Anomaly, AA_5)**.** *A bank account v_i has a funding-amount-fluctuation anomaly AA_5, if the recent transaction amount far exceeds the previous one, i.e., $\sum_{e\in E^{in}_{t-\Delta,t}(v_i)\cup E^{out}_{t-\Delta,t}(v_i)}value(e)\geq\lambda\cdot\sum_{e\in E^{in}_{t'-\Delta,t'}(v_i)\cup E^{out}_{t'-\Delta,t'}(v_i)}value(e)$.*

Definition 12 (Split Anomaly, AA_6)**.** *An account v_i has a split anomaly AA_6, if there is a star structure in FAN, centered at v_i with outgoing edge set $E'\subseteq E^{out}_{t-\Delta,t}(v_i)$ to accounts with different owners, i.e., $\forall e^{t_1}_{i,j}\in E'$, $t_1\in[t-\Delta,t)$, check if $r=(owner(v_i),owner(v_j))\notin R$ and $|E'|>\tau_c$, where $E'\subseteq E^{out}_{t-\Delta,t}(v_i)$.*

Definition 13 (Merge Anomaly, AA_7)**.** *An account v_i has a merge anomaly AA_7, if there is a star structure in FAN, centered at v_i with incoming edges $E'\subseteq E^{in}_{t-\Delta,t}(v_i)$ from unfamilar accounts, i.e., $\forall e^{t_1}_{j,i}\in E'$, $t_1\in[t-\Delta,t)$, check if $r=(owner(v_j),owner(v_i))\notin R$ and $|E'|>\tau_c$, where $E'\subseteq E^{in}_{t-\Delta,t}(v_i)$.*

Definition 14 (Immediate In-Out Anomaly, AA_8)**.** *A bank account v_i has an immediate in-out anomaly AA_8, if it is frequently transferred in $E_1\subseteq E^{in}_{t-\Delta,t}(v_i)$ and transfers out $E_2\subseteq E^{out}_{t-\Delta,t}(v_i)$ a sum of money within a short period of time, i.e., $\sum_{e\in E_1}value(e)-\sum_{e'\in E_2}value(e')\leq\epsilon_v$.*

Definition 15 (Road-Toll Anomaly, AA_9)**.** *The road-toll anomaly happens in a path with at least two edges in FAN, if the intermediate account v_i in the path belongs to a bank staff or his/her relatives and the account receives some kickback, i.e., $0<value(e^{t_1}_{a,i})-value(e^{t_2}_{i,b})<\epsilon_v$, and the owner of v_i meets any of the following conditions: (i) $type(owner(v_i))$="bank staff", or (ii) there exists a person $p\in P$, $type(p)$="bank staff" and $r=(owner(v_i),p)\in R$.*

Definition 16 (One-Way-Transfer Anomaly, AA_{10})**.** *If the account v_i transfers money to the account v_j through different paths (maybe across other persons, enterprises, or bank accounts) without paths from v_j to v_i, there may be illegal transactions). Say, the account v_i and v_j have AA_{10}, if $|\mathcal{P}(v_i,v_j)|>0$ and $\mathcal{P}(v_j,v_i)|=0$, where v_i is the transfer-out account and v_j is the account transferred in. $\mathcal{P}(v_i,v_j)$ denotes the path set containing all reachable paths from v_i to v_j, and $|\mathcal{P}(v_i,v_j)|$ is the number of paths from v_i to v_j.*

VI. Algorithm Design

In this section, we propose a uniform framework to dynamically detect anomalies from disguised normal financial activities. Firstly, we design a family of algorithms to detect implicit abnormal patterns (anomalies) dynamically from normal financial activities. Then, we propose the "Clue Chains Tracing" algorithm to find and infer clue chains from these suspect financial anomalies that can tell where the money comes from and what it is used for.

A. Overview of the Detection Framework

Detecting anomalies from disguised normal financial activities is one of the challenging tasks, especially from disguised normal activities. In this part, we propose a novel uniform framework to detect anomalies from disguised normal financial activities by utilizing database search techniques. Newly discovered anomaly patterns can be added to the framework easily. We classify ten implicit anomalies of financial activities ($\text{AA}_1\sim\text{AA}_{10}$) into three categories, including (i) single online abnormal activity: a new activity is abnormal; (ii) composite online abnormal activity: a combination of new activity and some historical activities in FAN; and (iii) composite history abnormal activity: anomalies hidden in historical financial activities in FAN. Accordingly, we design trigger-based detection, monitor-based detection, and mining-based detection algorithms, respectively. The structure tree of detecting anomalies efficiently is demonstrated in Fig. 5.

The framework of malicious financial activity detection is as follows: (1) Anomaly Detection Algorithms. We design a family of detecting algorithms to monitor intractable implicit anomalies induced by new arrival transactions in real time. (2) Clue Chains Tracing. Trace the detected anomalies (abnormal patterns) to form clue chains. (There may be multiple possible chains related to an anomaly.) (3) Clue chains rank and recommendation. Estimate the clue chains traced and recommend the most suspicious chains to financial institutions.

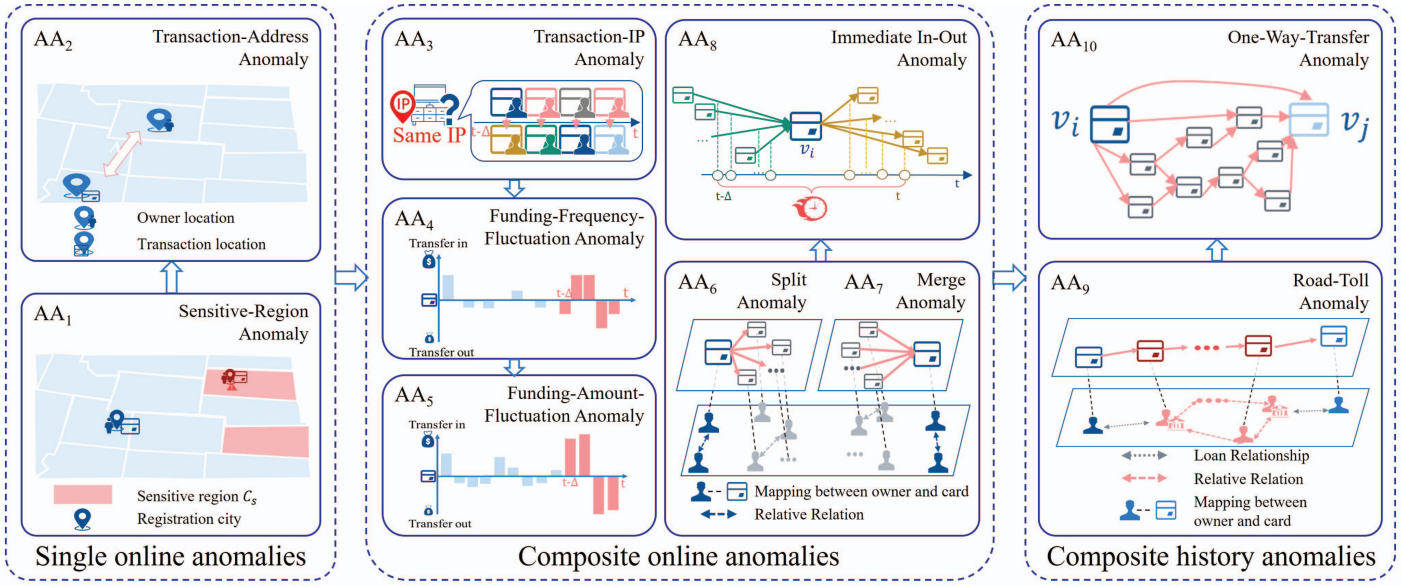


Fig. 5: Structure tree of Themis.

B. Anomaly Detection Algorithms

The three types of anomalies are detected in different ways. We give the three corresponding algorithms as follows.

Trigger-based detection. The trigger-based detection algorithm dynamically checks every new arrival single financial activity by examining its relative relations and spatio-temporal features, shown as Algorithm 1. The algorithm can find single online anomalies AA_1 and AA_2 in constant time.

Algorithm 1: Trigger-based detection (AA_1 and AA_2)

Input: A new arrival financial activity act_n;
Output: Detected anomalies AA_1 and AA_2;
$\text{AA}_1, \text{AA}_2 \leftarrow \emptyset$;
if act_n *is a "NewAccount"* v *and* $regcity(v) \in C_s$ **then** AA_1.add(act_n);
if act_n *is a "NewTransfer"* $e = (v_i, v_j, t)$ *and* $actloc_t(e) \neq loc_t(owner(v_i))$ **then** AA_2.add(act_n);
Return AA_1 and AA_2.

Monitor-based detection. The monitor-based detection algorithm detects a set of abnormal activities by combining new arrival activities with historical activities, where FAN is normal but "FAN + new arrival activities" is abnormal. It is obvious that AA_3-AA_8 are all typical composite online anomalies.

The monitor-based detection algorithm checks the time sliding window $[t-\Delta, t)$, where t is the time slot of a new arrival activity act_n. A straightforward way is to examine anomalies within the sliding window $[t-\Delta, t)$ according to the definitions of AA_3-AA_8. In order to accelerate the calculations, we propose an incremental detection (shown as Algorithm 2) by comparing with its previous sliding window $[t'-\Delta, t')$, where act_e is the set of expired activities (if any) for the new window $[t-\Delta, t)$.

Algorithm 2: Monitor-based detection (AA_3 -AA_8)

Input: A new activity act_n with time slot t;
Previous time window $[t'-\Delta, t')$;
Output: Detected anomalies $\text{AA}_3 - \text{AA}_8$;
$\text{AA}_3, \text{AA}_4, \text{AA}_5, \text{AA}_6, \text{AA}_7, \text{AA}_8 \leftarrow \emptyset$;
foreach $\text{AA}_i \in \text{AA}_3 - \text{AA}_8$ **do**
  Calculate expired act_e within $[t'-\Delta, t-\Delta)$;
  if act_n *and* act_e *satisfy the alarm condition* **then**
    AA_i.add(act_n);
Return $\text{AA}_3 - \text{AA}_8$;

The alarm conditions of $\text{AA}_3 - \text{AA}_8$ are listed as follows.

- Alarm condition of AA_3 (Transaction-IP Anomaly). Suppose a new activity act_n is initiated on an account v and completed on a device with a certain IP address ip. Let $E^{out}_{t'-\Delta,t'}(v)$ be the set of activities initialed on v within $[t'-\Delta, t')$. We could maintain a hash index on $E^{out}_{t'-\Delta,t'}(v)$ for different IP addresses. Then, we can use $O(1)$ time to get the set of activities $E'_{ip} \subseteq E^{out}_{t'-\Delta,t'}(v)$ whose corresponding activities are operated on ip within $[t'-\Delta, t')$. Let $E_e \subseteq E'_{ip}$ be the set of activities within $[t'-\Delta, t-\Delta)$, then the alarm condition of AA_3 is $|E'_{ip}| - |E_e| + 1 > \tau_c$.
- Alarm condition of AA_4 (Funding-Frequency-Fluctuation Anomaly). For the account v initiated on act_n, the alarm condition of AA_4 is $(\lambda-1)\cdot(|E^{in}_{t'-\Delta,t'}(v)| + E^{out}_{t'-\Delta,t'}(v)|) + (|E^{in}_{t'-\Delta,t-\Delta}(v)| + |E^{out}_{t'-\Delta,t-\Delta}(v)|) \leq 1$.
- Alarm condition of AA_5 (Funding-Amount-Fluctuation Anomaly). For the account v initiated on act_n, the alarm condition of AA_5 is $(\lambda-1)\cdot\sum_{e\in E^{in}_{t'-\Delta,t'}(v)\cup E^{out}_{t'-\Delta,t'}(v)} value(e) + \sum_{e\in E^{in}_{t'-\Delta,t-\Delta}(v)\cup E^{out}_{t'-\Delta,t-\Delta}(v)} value(e) \leq 1$.
- Alarm conditions of AA_6 and AA_7 (Split and Merge

Anomaly). For the account v initiated on act_n, we could maintain a hash index on $E^{out}_{t'-\Delta,t'}(v)$ for cards with different owners. Then, we can use $O(1)$ time to get the set of activities $E^{out}_{owner} \subseteq E^{out}_{t'-\Delta,t'}(v)$ whose transferred-out cards have the same owner with v. Let $E_{out} \subseteq E^{out}_{t'-\Delta,t'}(v)/E^{out}_{owner}$ be the set of transferred-out activities within $[t'-\Delta, t-\Delta)$, then the alarm condition of AA_6 is $|E^{out}_{t'-\Delta,t'}(v)| - |E_{out}| + 1 > \tau_c$.

Similarly, let v be the transferred-in account of act_n. Let $E^{in}_{owner} \subseteq E^{in}_{t'-\Delta,t'}(v)$ be the activities whose transferred-in account have the same card owner with v, and $E_{in} \subseteq E^{in}_{t'-\Delta,t'}(v)/E^{in}_{owner}$ be the set of transferred-in activities within $[t'-\Delta, t-\Delta)$, then the alarm condition of AA_7 is $|E^{in}_{t'-\Delta,t'}(v)| - |E_{in}| + 1 > \tau_c$.

• Alarm condition of AA_8 (Immediate In-Out Anomaly). For the account v initiated on act_n, let e_n be the corresponding edge of act_n and val_d be the difference value $\sum_{e\in E^{in}_{t'-\Delta,t'}(v)} value(e) - \sum_{e\in E^{out}_{t'-\Delta,t'}(v)} value(e) - (\sum_{e\in E^{in}_{t'-\Delta,t-\Delta}(v)} value(e) - \sum_{e\in E^{out}_{t'-\Delta,t-\Delta}(v)} value(e))$, then the alarm condition of AA_8 is $val_d + value(e_n) > \epsilon_v$ or $val_d - value(e_n) > \epsilon_v$.

Mining-based detection method. The mining-based detection algorithm discoveries a set of composite history abnormal activities (AA_9 and AA_{10}) by searching historical activities in FAN, shown as Algorithm 3. The verification of reachability among different accounts is improved via introducing maximum-flow min-cut theory [35], [36], pruning unnecessary search paths and early stopping in DFS when detecting AA_{10}.

Algorithm 3: Mining-based detection (AA_9 and AA_{10})

Input: An FAN $G = (V, E)$
Output: Detected anomalies AA_9 and AA_{10}.

```
AA9, AA10 ← ∅;
foreach v_i ∈ V do
    foreach e^{t1}_{j,i} ∈ E^{in}(v_i) do
        start← e^{t1}_{j,i};
        foreach e^{t2}_{i,k} ∈ E^{out}(v_i) do
            if value(e^{t1}_{j,i}) − value(e^{t2}_{i,k}) < ε_v then
                end← e^{t}_{i,k} ;
        AA9.add((start,end));
    DFS from v_i, label the visited accounts, and add
      v_j to an empty set V_1 when visiting a labeled v_j;
    foreach v_j ∈ V_1 do
        if The maximum flow from v_j to v_i is 0 then
            E_c ← the minimum cut set found by the
              maximum flow algorithm from v_i to v_j;
            foreach (v_i, v_k) ∈ V_1 do AA10.add(e^{t}_{i,k});
            foreach (v_k, v_j) ∈ V_1 do AA10.add(e^{t}_{k,j});
Return AA9 and AA10.
```

C. Clue Chains Tracing

For a detected anomaly with an amount of money, a financial institution may want to trace where the money comes from and what is it used for. We mine clue chains by tracing the activities in FAN and require that the final discovery of the chain of clues includes the detected abnormal activity. The main idea of the clue chain tracing algorithm is based on the the detected activity and examine its neighborhood nodes in FAN. We trace the source of the money by the similarity of the timestamp and the amount of money.

In order to recommend the most suspicious clue chains to institutions, we evaluate these chains by scoring rules and choose the most suspicious chains. The scoring rules are dynamically adjusted according to the business scenario, taking into account time interval, transfer amount, intimacy, and so on, since different clue chains and relationships have different importance for institutions. The framework for chain tracing and recommendation is as follows: (a) Set the time and money thresholds; (b) Start from suspect anomalies detected and trace abnormal financial activities according to the human-money relationship in FAN, and form some suspect clue chains; (c) Rank these chains based on dynamically adjusted scoring rules. In the evaluation part, top-k clue chains will be recommended to financial institutions to assist in detecting anomalies effectively.

VII. Experiments

We show the efficiency and effectiveness of our approach. More importantly, we developed an anomaly-detection system, called Themis, to detect abnormal activities. The interface of Themis and case studies are also presented. We conducted all the experiments on a machine with an Intel(R) Core(TM) i7-10710U and 16GB memory in Windows OS. All the methods are implemented in C++ compiled by g++ with O3 turned on. All the detection algorithms are run in memory.

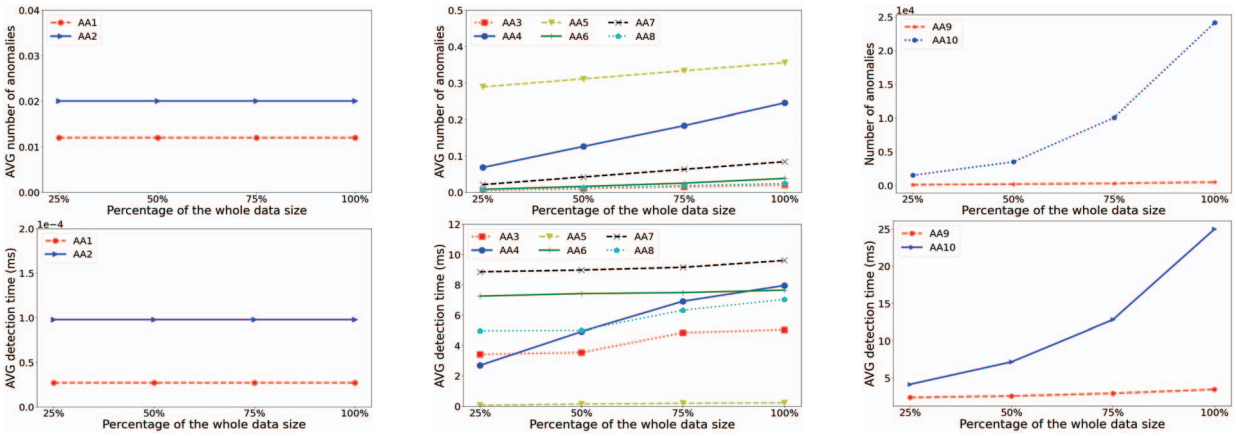
A. Datasets

Synthetic dataset: In order to evaluate the performance of Themis in detecting anomalies, we apply an existing graph generator, Watts-Strogatz[1] to generate the background financial graph and relative network, including $1,000$ nodes and $993,133$ edges. Then abnormal patterns are inserted into the graph as the ground truth to evaluate our anomaly detection algorithms, clue chains trancing technology, and Themis.

Real bank statements: To further evaluate the performance of Themis in real-world scenarios, some financial activities of a bank are utilized as a benchmark dataset in our experiments, including $110,509$ transfer records and 47 features. We provide a brief description of this dataset as follows, including the bank statement (transaction amount, transaction category, etc.), cardholder-related information (name, card number, identification number, etc.), counterparty information (account, ID number, etc.), and relative relationships.

New arrival activities: Based on the synthetic dataset, we generate 100 new nodes and 500 new edges with a sorted time stamp to simulate newly generated financial activities. The distribution of generated time stamps follows the same time distribution in the financial graph.

[1]https://github.com/sleepokay/watts-strogatz



(a) Trigger-base detection. (b) Monitor-based detection. (c) Mining-based detection.

Fig. 6: Performance of detection algorithms on a synthetic dataset.

B. Evaluation of anomaly detecting algorithms

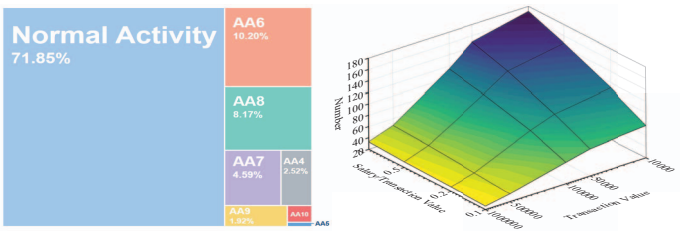
Anomalies analysis on synthetic datasets. Fig. 6 shows the performance of three detection algorithms on synthetic datasets with different data sizes. We record the average number of detected anomalies and detection time for each new arrival activity for AA_1-AA_8, and the number of detected anomalies and detection time for AA_9-AA_{10}.

Fig. 6(a) shows that the average number of anomalies using trigger-based detection is small and independent from the data size because most of the incoming activity is normal and only related to the distribution of incoming activities. The average detection time is constant for every new arrival activity. The results in Fig. 6(b) show that the monitor-based detection algorithms have good scalability for detecting anomalies. The number of detected anomalies and the detection time increase linearly with the increase in dataset size. Fig. 6(c) shows the results of mining-based detection. When increasing the size of the dataset, the number of detected anomalies increases linearly for AA_9 and quadratically for AA_{10} since we want to find every One-Way-Transfer pair in FAN. Mining-based detection requires more time since the complexity of the mining-based detection for AA_9 is $O(|E|^2)$ and for AA_{10} is $O(|V|^2|E|)$.

Anomalies analysis on real bank statements. Fig. 7 (a) demonstrates the proportion of abnormal patterns detected in real bank statements. Obviously, these anomalies are significantly rare compared with normal activities. Tracing the suspicious clue chains based on anomalies can improve the efficiency of institutions. Fig. 7 (b) shows the funding-amount-fluctuation anomalies detected, indicating that most customers' transferring amount is positively correlated with their historical transaction amount, and only a small part of users have funding-amount-fluctuation anomalies.

C. Evaluation of "clue chains Tracing Algorithms"

1) Efficiency: In this part, we show the efficiency of clue chain tracing in real bank statements.



(a) Proportion of anomalies. (b) Funding-Amount-Fluctuation.

Fig. 7: Statistics of anomalies on real bank statements.

TABLE II: Number of chains for individual accounts.

	Alice	Bob	Charlie	David	Emma
Total number of chains	3,975	11,313	11,654	1,576	7,029
Chains (ϵ_t=6,ϵ_a=\$10,000)	217	503	179	47	219
Chains (ϵ_t=1,ϵ_a=\$10,000)	15	21	8	3	9

Verification benefit for individual accounts. Table II demonstrates the number of chains for a specific account of a person. "Total number of chains" is the account's total financial clues, i.e., the account of David has $1,576$ transaction-related chains, which should be checked by financial institutions manually before using Themis. In particular, Chains (ϵ_t, ϵ_a) is the number of clue chains traced by Themis under a given time interval ϵ_t and amount threshold ϵ_a (months and dollars), then top-k chains are recommended to institutions to verify manually (top-10 chains in Themis). For example, the institutions only need to check 47 and 3 suspect chains about David after deploying Themis under the threshold of ($\epsilon_t = 6$ and $\epsilon_a = \$10,000$) and ($\epsilon_t = 1$ and $\epsilon_a = \$10,000$), improving the efficiency compared with previous verification.

Verification benefit for anomalies. Table III shows the average number of clue chains and their running time. It shows that when the time interval extends from one month to six months, more suspicious clues appear. The financial institution could use these two parameters to trace the clue chain easily.

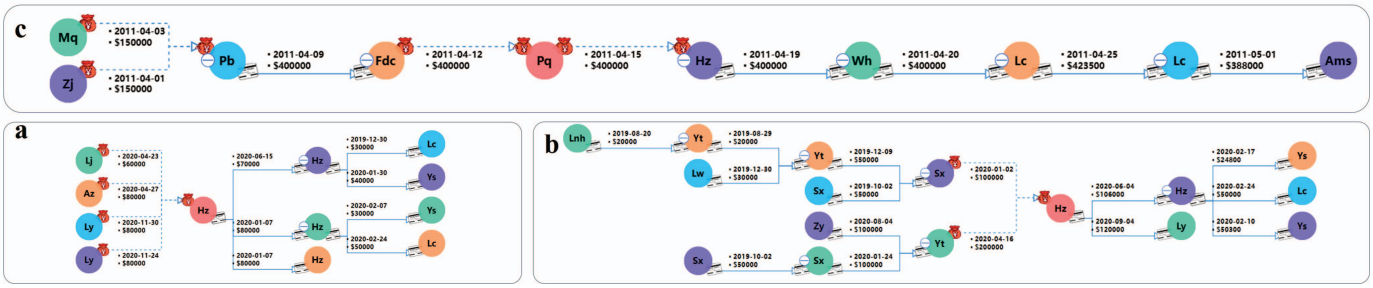


Fig. 8: Case studies: clue chains detected by our system Themis.

a. Relation Tracing

b. Relation Tracing with Abormal Link

c. The Personal Financial Overview

d. Personnel Funds Link

Fig. 9: Interface-Diagram of Themis.

TABLE III: Performance of chains for anomalies.

	AA_4	AA_5	AA_8	AA_9	AA_{10}
Average number of chains	17,647	537	8,673	2,130	47,586
Chains (ϵ_t=6,ϵ_a=\$10,000)	425	64	328	165	760
Time (ϵ_t=6,ϵ_a=\$10,000)	2.87s	1.25s	2.67s	0.78s	3.18s
Chains (ϵ_t=1,ϵ_a=\$10,000)	200	45	229	121	521
Time (ϵ_t=1,ϵ_a=\$10,000)	1.66s	0.99s	1.54s	0.61s	2.56s

Performance. Table III demonstrates the average running time (seconds) of clue chain generation in Themis, including the process of tracing, evaluating, and recommending chains. We can find that the process of clue chain inference is significantly efficient (all in seconds level), supporting anomaly detection in large-scale bank statements.

2) Effectiveness: We conduct some case studies of clue chains generated by Themis, whose anomalies have been verified by institutions. In fig. 8, the node represents a person's account, and the edge denotes financial activities between two accounts where the activities with solid lines are anomalies traced and the activities with the dashed line are inferred via the account's deposits and withdrawals.

Case 1 (Clue chain detected by tracing AA_5): As shown in Fig. 8(a), `Hz` is `Lc`'s driver. He is detected as funding-amount fluctuation by using "Themis". These activities significantly exceed `Hz`'s income level ($5,000 monthly salary). Our approach traces `Hz`'s transactions and recommends this suspicious clue chain. In this anomaly (AA_5), `Hz` transferred massive money to `Lc` and `Ys`.

Case 2 (Clue chain detected by tracing AA_8): As shown in Fig. 8(b), our algorithms detect that `Hz` acts as an intermediary and receives two cash deposits from `Sx` and `Yt` respectively. Then `Hz` transferred the money to `Ly` and his other account in a short period of time. Our approach traces that he successfully transfers from `Sx` and `Yt` to `Lc` and `Ys` as an intermediary.

Case 3 (Clue chain detected by tracing AA_4 and AA_8): Fig. 8(c) demonstrates a clue chain that tracks where $400,000 comes from and what is it used for. $388,000 was spent by `Lc` in a luxury shop at the end.

D. Themis: An anomaly-detection system

More importantly, we developed an anomaly-detection system (Themis) that can detect anomalies from disguised normal financial activities and find suspicious clue chains. It has been deployed in many real scenarios, including banks and financial institutions. The pipeline of Themis is demonstrated as follows: (a) Anomalies are detected by "Anomaly Detecting Algorithm"; (b) Suspect clue chains are traced based on anomalies via "clue chains Tracing Algorithms"; (c) Themis evaluates and recommends suspicious clue chains to institutions. The interface of Themis is shown in Fig. 9, including individuals' assets, bank statements, cash transactions, relative relationships, and so on. With the help of Themis, the anomalies (red solid lines in Fig. 9(b)) can be detected.

VIII. Conclusion

In this paper, we design a uniform framework to detect anomalies from disguised normal financial activities. We are the first to formalize and detect complex anomalies, meanwhile considering heterogeneous features. In particular, we propose a clue chain tracing technology to recommend suspect clue chains for institutions. What's more, we deploy a system, Themis, to detect anomalies and infer clue chains in some real scenarios. Experiments on synthetic datasets and real bank statements show the efficiency and effectiveness of the Themis.

IX. Acknowledgements

The work is partially supported by the National Natural Science Foundation of China (Nos. U22A2025, 62072088, 62232007), and Liaoning Provincial Science and Technology Plan Project - Key R&D Department of Science and Technology (No. 2023JH2/101300182).

References

[1] B. L. Handoko, R. N. A. Putri, and S. Wijaya, "Analysis of fraudulent financial reporting based on fraud heptagon model in transportation and logistic industry listed on idx during covid-19 pandemic," in *International Conference on Software and e-Business*, 2022, pp. 56–63.

[2] E. Hytis, V. Nastos, C. Gogos, and A. Dimitsas, "Automated identification of fraudulent financial statements by analyzing data traces," in *The South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE, 2022, pp. 1–7.

[3] S. Dhankhad, E. Mohammed, and B. Far, "Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study," in *IEEE international conference on information reuse and integration (IRI)*, 2018, pp. 122–125.

[4] J. C. Ying, J. Zhang, C. W. Huang, K. T. Chen, and V. S. Tseng, "Fraudetector +: An incremental graph-mining approach for efficient fraudulent phone call detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 6, pp. 1–35, 2018.

[5] S. Zhou, J. He, H. Yang, D. Chen, and R. Zhang, "Big data-driven abnormal behavior detection in healthcare based on association rules," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2020.

[6] A. C. Kim, W. H. Park, and D. H. Lee, "A framework for anomaly pattern recognition in electronic financial transaction using moving average method," in *IT Convergence and Security*, 2013, pp. 93–99.

[7] J.-S. Chang and W.-H. Chang, "Analysis of fraudulent behavior strategies in online auctions for detecting latent fraudsters," *Electronic Commerce Research and Applications*, vol. 13, no. 2, pp. 79–97, 2014.

[8] F. Rahmani, C. Valmohammadi, and K. Fathi, "Detecting fraudulent transactions in banking cards using scale-free graphs," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 19, p. e7028, 2022.

[9] H. Zhang and W. Zhou, "A two-stage virtual machine abnormal behavior-based anomaly detection mechanism," *Cluster Computing*, vol. 25, no. 1, pp. 203–214, 2022.

[10] M. Y. Turaba, M. Hasan, N. I. Khan, and H. A. Rahman, "Fraud detection during financial transactions using machine learning and deep learning techniques," in *IEEE International Conference on Communications, Computing, Cybersecurity, and Informatics*, 2022, pp. 1–8.

[11] E. E. Papalexakis, A. Beutel, and P. Steenkiste, "Network anomaly detection using co-clustering," in *Encyclopedia of Social Network Analysis and Mining, 2nd Edition*, 2018.

[12] S. Cao, X. Yang, J. Zhou, X. Li, Y. Qi, and K. Xiao, "Poster: Actively detecting implicit fraudulent transactions," in *The ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2475–2477.

[13] X. Gu and H. Wang, "Online anomaly prediction for robust cluster systems," in *Proceedings of the 25th International Conference on Data Engineering*, 2009, pp. 1000–1011.

[14] Z. Wang, "Abnormal financial transaction detection via ai technology," *International Journal of Distributed Systems and Technologies (IJDST)*, vol. 12, no. 2, pp. 24–34, 2021.

[15] R. A. L. Torres and M. Ladeira, "A proposal for online analysis and identification of fraudulent financial transactions," in *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[16] J. He, C.-C. M. Yeh, Y. Wu, L. Wang, and W. Zhang, "Mining anomalies in subspaces of high-dimensional time series for financial transactional data," in *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, (ECML PKDD)*. Springer, 2021, pp. 19–36.

[17] Y. Li, Y. Sun, and N. Contractor, "Graph mining assisted semi-supervised learning for fraudulent cash-out detection," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 2017, pp. 546–553.

[18] X. Mao, M. Liu, and Y. Wang, "Using gnn to detect financial fraud based on the related party transactions network," *Procedia Computer Science*, vol. 214, pp. 351–358, 2022.

[19] Y. Pei, F. Lyu, W. V. Ipenburg, and M. Pechenizkiy, "Subgraph anomaly detection in financial transaction networks," 2020.

[20] D. Wang, Y. Qi, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, and S. Yang, "A semi-supervised graph attentive network for financial fraud detection," in *2019 IEEE International Conference on Data Mining*, 2019, pp. 598–607.

[21] W. Kudo, M. Nishiguchi, and F. Toriumi, "Gcnext: graph convolutional network with expanded balance theory for fraudulent user detection," *Social Network Analysis and Mining*, vol. 10, pp. 1–12, 2020.

[22] S. Pathan and V. Shrivastava, "Identifying linked fraudulent activities using graphconvolution network," *arXiv:2106.04513*, 2021.

[23] X. Wang, Z. Wan, and Y. Zhang, "A dqn-based internet financial fraud transaction detection method," in *International Conference on Computer Science and Application Engineering (CSAE)*, 2021, pp. 1–5.

[24] B. Can, A. G. Yavuz, M. E. Karsligil, and M. A. Güvensan, "A closer look into the characteristics of fraudulent card transactions," *IEEE Access*, vol. 8, pp. 166 095–166 109, 2020.

[25] X. Mao, H. Sun, X. Zhu, and J. Li, "Financial fraud detection using the related-party transaction knowledge graph," *Procedia Computer Science*, vol. 199, pp. 733–740, 2022.

[26] T. Chen, L. Tang, Y. Sun, Z. Chen, and K. Zhang, "Entity embedding-based anomaly detection for heterogeneous categorical events," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 1396–1403.

[27] H. Xiang, H. Hu, and X. Zhang, "Deepiforest: A deep anomaly detection framework with hashing based isolation forest," in *IEEE International Conference on Data Mining (ICDM)*, 2022, pp. 1251–1256.

[28] C. Wang and H. Zhu, "Wrongdoing monitor: A graph-based behavioral anomaly detection in cyber security," *Trans. Info. For. Sec.*, vol. 17, p. 2703–2718, jan 2022.

[29] S. Reddy, P. Poduval, A. V. S. Chauhan, M. Singh, S. Verma, K. Singh, and T. Bhowmik, "Tegraf: temporal and graph based fraudulent transaction detection framework," in *The ACM International Conference on AI in Finance (ICAIF)*, 2021, pp. 1–8.

[30] M. Shen, A. Sang, P. Duan, H. Yu, and L. Zhu, "Threat prediction of abnormal transaction behavior based on graph convolutional network in blockchain digital currency," in *Blockchain and Trustworthy Systems (BlockSys)*. Springer, 2021, pp. 201–213.

[31] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: bounding graph fraud in the face of camouflage," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 895–904.

[32] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Min. Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, 2015.

[33] Z. Zhang and L. Zhao, "Unsupervised deep subgraph anomaly detection," in *IEEE International Conference on Data Mining (ICDM)*, 2022, pp. 753–762.

[34] A. Zhang, B. Wu, and Y. Li, "A heterogeneous graph-based fraudulent community detection system," in *IEEE International Conference on e-Business Engineering (ICEBE)*, 2021, pp. 43–48.

[35] H. Yildirim, V. Chaoji, and M. J. Zaki, "GRAIL: a scalable index for reachability queries in very large graphs," *VLDB J.*, vol. 21, no. 4, pp. 509–534, 2012.

[36] H. Wei, J. X. Yu, C. Lu, and R. Jin, "Reachability querying: An independent permutation labeling approach," *Proc. VLDB Endow.*, vol. 7, no. 12, pp. 1191–1202, 2014.