INSTITUTO UNIVERSITÁRIO DE LISBOA

Ciências ULisboa

# Hubs and Gatekeepers, the attributes of Capability in Complex networks

Pedro Artur de Almeida Fidalgo

PhD in Complexity Sciences

Supervisor:

Doctor Rui Lopes, Associate Professor, Instituto Universitário de Lisboa, Portugal

July 2025

Department of Information Science and Technology

Hubs and Gatekeepers, the attributes of Capability in Complex networks

Pedro Artur de Almeida Fidalgo

PhD in Complexity Sciences

Jury:

President of the Jury Doctor Jorge Louçã, Full Professor, Instituto Universitário de Lisboa, Portugal
Doctor José Torres, Associate Professor, University Fernando Pessoa, Portugal
Doctor Rui Moreira, Associate Professor, University Fernando Pessoa, Portugal
Doctor John Symons, Full Professor, University of Kansas, USA
Doctor António Fonseca, Assistant Professor, Instituto Universitário de Lisboa, Portugal
Doctor Rui Lopes, Associate Professor, Instituto Universitário de Lisboa, Portugal

July 2025

*Dedicado á minha esposa Luciana, e ao nosso filho Lucas, e aos meus pais Laura e Artur que têm sido a minha constante fonte de amor e apoio.*

# Acknowledgements

I would like to express my sincere gratitude to all those who supported and contributed to the completion of this PhD thesis.

First and foremost, I thank my family. To my wife Luciana, our son Lucas, and my parents, Laura and Artur, your constant presence, encouragement, and understanding have been fundamental throughout this journey. To Manoela and Luziel, thanks for being with me along the way.

I am especially grateful to my advisor, Rui Jorge Lopes, for his consistent guidance, insight, and support. His mentorship was fundamental in shaping the direction and rigor of this work. I also extend my appreciation to Christos Faloutsos for his valuable advice and technical expertise, which contributed meaningfully to the development of this thesis.

To my friends, thank you for your support during this journey. Your presence helped lighten the pressure and made things easier.

This thesis is the result of a collective effort, and I am deeply appreciative of everyone who contributed in ways both big and small.

Pedro Fidalgo was partly supported by the AIDA project – Adaptive, Intelligent and Distributed Assurance Platform (reference POCI-01-0247-FEDER-045907), co-financed by the ERDF – European Regional Development Fund through the Operational Program for Competitiveness and Internationalisation – COMPETE 2020 and by the Portuguese Foundation for Science and Technology.

Rui J. Lopes was partly supported by the Fundação para a Ciência e Tecnologia, under Grant Number UIDB/50008/2020, attributed to Instituto de Telecomunicações.

## Abstract

This thesis investigates new methods for fraud detection in fifth-generation (5G) mobile networks, focusing on the identification of anomalous behaviours and malicious structures in large-scale graphs. 5G networks introduce disruptive use cases such as Ultra-Reliable Low-Latency Communications (URLLC), Massive Machine-Type Communications (mMTC), and Enhanced Mobile Broadband (eMBB), and reshape the user profile, now predominantly composed of Internet of Things (IoT) devices, whose communication patterns differ significantly from those of human users. This new ecosystem creates an expanded attack surface, with distributed fraud vectors and data volumes that render traditional centralized approaches unfeasible.

The main research question lies in the ability to effectively detect fraudulent entities and organized patterns in a distributed, dynamic context subject to privacy constraints. To address this challenge, a distributed machine learning architecture is proposed, supported by real-time processing at the network edge (edge computing), ensuring data privacy.

The developed methodology includes four main components: multi-stage detection of devices compromised with malware; topological analysis to identify entities holding strategic structural positions in organized fraudulent networks; detection of anomalous patterns of traffic and connectivity in large-scale dynamic networks; and interactive visualizations designed to facilitate result explainability and enable detailed analysis of suspicious subgraphs. Additionally, federated learning models were implemented, capable of operating on decentralized data without compromising predictive accuracy.

The results demonstrate high effectiveness in early fraud detection and in identifying entities with structurally relevant roles in malicious networks. The combination of graph analysis, explainable visualization, and federated learning achieved performance levels comparable to centralized models (AUC > 0.97), while meeting the privacy, latency, and scalability requirements imposed by 5G networks.

It is concluded that it is feasible to implement a robust and efficient fraud detection model adapted to the reality of next-generation mobile networks. The contributions of this thesis reinforce the importance of explainable and distributed artificial intelligence in the protection of critical telecommunications infrastructures, by proposing concrete and evidence based solutions to tackle emerging threats in this new technological paradigm.



**Keywords:** Fraud Methods, Fraud Networks, Network Control, Bridging Centrality, Influence, 5G Fraud, Data Visualization, Complex Networks, Federated ML

# Resumo

Esta tese investiga novos métodos para a deteção de fraude em redes móveis de quinta geração (5G), centrando-se na identificação de comportamentos anómalos e estruturas maliciosas em grafos de grande escala. As redes 5G introduzem casos de uso disruptivos como comunicações ultra-fiáveis e de baixa latência (URLLC), comunicações massivas entre máquinas (mMTC) e banda larga móvel melhorada eMBB e alteram o perfil dos utilizadores, agora maioritariamente composto por dispositivos da Internet das Coisas (IoT), cujos padrões de comunicação diferem significativamente dos utilizadores humanos. Este novo ecossistema cria uma superfície de ataque alargada, com vetores de fraude distribuídos e volumes de dados que inviabilizam abordagens centralizadas tradicionais.

A principal questão de investigação prende-se com a capacidade de detetar eficazmente entidades fraudulentas e padrões organizados num contexto distribuído, dinâmico e sujeito a restrições de privacidade. Para responder a este desafio, foi proposta uma arquitetura baseada em aprendizagem automática distribuída, suportada por processamento em tempo real na periferia da rede (edge computing), garantindo a privacidade dos dados.

A metodologia desenvolvida inclui quatro eixos principais: deteção multietapas de dispositivos comprometidos com malware, análise topológica para identificação de entidades com posições estruturais estratégicas em redes fraudulentas organizadas, deteção de padrões anómalos de tráfego e conectividade em redes dinâmicas de grande escala, e visualizações interativas concebidas para facilitar a explicabilidade dos resultados e permitir a análise pormenorizada de subgrafos suspeitos. Adicionalmente, foram implementados modelos de aprendizagem federada, capazes de operar sobre dados descentralizados sem comprometer a precisão preditiva.

Os resultados obtidos demonstram elevada eficácia na deteção precoce de fraudes e na identificação de entidades com papéis estruturais relevantes em redes maliciosas. A combinação entre análise de grafos, visualização explicável e aprendizagem federada permitiu alcançar níveis de desempenho comparáveis a modelos centralizados (AUC > 0.97), respeitando os requisitos de privacidade, latência e escalabilidade exigidos pelas redes 5G.

Conclui-se que é viável implementar um modelo robusto e eficiente de deteção de fraude adaptado à realidade das redes móveis de nova geração. Os contributos desta tese reforçam a importância da inteligência artificial explicável e distribuída na proteção de infraestruturas críticas de telecomunicações, propondo soluções concretas e sustentadas para enfrentar ameaças emergentes neste novo paradigma tecnológico.

# Resumo desenvolvido

Esta tese teve como objectivo investigar a identificação de comportamentos suspeitos e fraude em redes de telecomunicações, especificamente dentro do contexto actual da quinta geração de redes móveis. Com o objectivo de ligar tudo e todos, e com características disruptivas associadas a novos padrões de uso, modelos de conectividade e tecnologias, as redes 5G representam uma mudança de paradigma não apenas na engenharia de redes, mas também na superfície de ataque e nas oportunidades de exploração por parte de agentes maliciosos.
A evolução massiva no número de dispositivos conectados e nos volumes de tráfego requer novas formas de observar, compreender e detectar desvios comportamentais sobretudo aqueles que escapam aos mecanismos tradicionais de detecção baseados em regras.
Um dos exemplos centrais nesse novo cenário é a evolução da fraude tradicional para formas de fraude inteligente (*smart fraud*), desenhadas para actuar de forma distribuída, adaptativa e furtiva, ajustando-se dinamicamente ao ambiente para evitar ser detectada.
Esta tese defende a necessidade de novas arquitecturas e métodos resilientes, escaláveis e descentralizados para a detecção de comportamentos suspeitos e fraudulentos, alinhados com os requisitos técnicos e operacionais das redes móveis de quinta geração.
As redes 5G introduzem requisitos exigentes de latência e desempenho, que só são possíveis com a adopção de soluções como o *Multi-Access Edge Computing* (MEC), onde o processamento ocorre na periferia da rede em vez de centros de dados centralizados.
Embora esta abordagem traga benefícios claros de desempenho e disponibilidade, ela também introduz novos desafios de detecção e controlo, especificamente: a descentralização dos dados, a limitação de recursos computacionais no edge e os constrangimentos de privacidade, tornando obsoleto muitos dos métodos convencionais de análise centralizada.
Assim, é fundamental desenvolver sistemas eficazes, distribuídos e capazes de funcionar sob condições práticas, com respeito pela privacidade, latência e capacidade de resposta.
O Capítulo 1 apresenta o contexto técnico e os principais desafios abordados nesta tese, introduzindo a arquitetura proposta para a deteção de fraude em redes 5G.
Exploram-se as fundações da utilização de técnicas de Inteligência Artificial (IA) aplicadas à análise de grafos de chamadas em larga escala, assim como o uso de aprendizagem federada para assegurar privacidade e escalabilidade no processamento distribuído.
Este capítulo também antecipa a ideia central que atravessa a tese: a fraude em redes 5G não deve ser encarada como uma sequência de eventos anómalos isolados, mas como o reflexo de estruturas organizadas que emergem na rede.
O Capítulo 2 apresenta o estado da arte na detecção de comportamento anómalo em redes, cobrindo metodologias que vão desde a análise de subgrafos e propagação de classificações até à detecção de padrões latentes e anomalias em grafos temporais. São discutidos algoritmos supervisionados, semi-supervisionados e não supervisionados, com particular atenção aos limites práticos de cada abordagem face à escala e complexidade das redes móveis.

Adicionalmente, este capítulo destaca que, apesar das várias abordagens existentes, a maior parte da literatura trata os comportamentos maliciosos de forma individualizada, raramente endereçando a detecção de estruturas maliciosas organizadas. Neste contexto, embora existam trabalhos relevantes sobre a propagação e utilização de malware, continua a verificar-se uma escassez de abordagens que tratem de forma sistemática aquele que se assume como o vector de fraude mais crítico para o 5G e gerações futuras: malwares especificamente concebidos para explorar vulnerabilidades em redes de telecomunicações, com impacto alargado sobre as indústrias que dependem dos seus serviços.

Com base nesta analise os Capítulos 3 a 7 desenvolvem o conteúdo principal da tese, no qual se propõe, implementar e validar um conjunto de métodos e ferramentas alinhados com a arquitectura apresentada, cada um abordando uma dimensão crítica do problema.

No Capítulo 3, abordamos a detecção de botnets móveis utilizando o sistema *Mondeo* (Multistage Botnet Detection). Esta escolha justifica-se pelo facto de, nas redes 5G, a infecção remota de terminais móveis por agentes maliciosos representar o principal vector de ataque. Ao controlar ilicitamente os dispositivos dos utilizadores, os atacantes transformam-nos em *zombies* ao seu serviço, capazes de executar qualquer tipo de fraude, desde originação de chamadas até acções coordenadas em larga escala com risco praticamente nulo para os atacantes, que independentemente da sua localização geográfica conseguem alcançar qualquer dispositivo de um utilizador.

O sistema *Mondeo* foi projectado para detectar *malware* do tipo *trojan* em múltiplas etapas, lidando com os desafios das redes móveis de próxima geração, como a elevada velocidade de propagação de ataques e a grande capacidade de adaptação dos botnets.

A solução proposta destaca-se pela sua capacidade de integração com os sistemas de segurança já implementados pelos operadores de rede, oferecendo uma abordagem escalável e ajustável à complexidade das redes 5G. Este capítulo explora as dinâmicas de fraude que envolvem múltiplos dispositivos infectados e como essas entidades podem ser identificadas precocemente através da monitorização de padrões de tráfego anómalo e anular qualquer tentativa de fraude destes equipamentos.

A introdução de novas metodologias de detecção, como a identificação de fontes de ameaça distribuídas e a análise do lucro ilícito gerado pelas actividades fraudulentas, são exploradas com maior profundidade nos Capítulos 4, 5 e 6, que detalham as técnicas utilizadas para mitigar tais ameaças.

O capítulo 4 apresenta a abordagem ***StarBridge***, desenvolvida para detectar estruturas fraudulentas (Fraud Rings) em redes de telecomunicações através da análise topológica da rede. Esta abordagem parte do pressuposto de que a fraude moderna, sobretudo em contextos como o 5G, raramente é executada de forma isolada, pelo contrário, envolve grupos organizados com papéis distintos, coordenação operacional e estrutura interna. Através de métricas como *Bridging Centrality*, *Influence* e *Control*, o método permite não só identificar nós individuais com elevada probabilidade de envolvimento fraudulento, mas também inferir

o grau de organização e interligação das redes suspeitas, distinguindo estruturas em anel (Fraud Rings)e padrões de distribuição hierárquica (Ring Masters). Esta capacidade de inferir papéis funcionais e segmentar grupos distintos é crucial para compreender o número, a dimensão e a sofisticação das entidades criminosas em actividade, apoiando não só a detecção como também estratégias de contenção e resposta.

O Capítulo 5 aprofunda os métodos utilizados para detectar comportamentos anómalos em grafos dinâmicos, com a aplicação do ***TgraphSpot***. Este método não supervisionado é projectado para detectar anomalias e actividades fraudulentas em redes compostas por milhões de vértices. Através da extracção de características estáticas como PageRank e Coreness,e dinâmicas, como o Inter-arrival Time (IAT) que mede os tempos entre chamadas sucessivas, o método consegue identificar padrões de tráfego atípicos que indicam a presença de comportamentos fraudulentos.

A identificação desses padrões é feita em grafos dinâmicos compostos por vastas quantidades de dados de chamadas telefonicas, o que representa um desafio significativo, devido à maldição da dimensionalidade (Curse of Dimensionality) e à necessidade de visualizar milhões de pontos simultaneamente. O ***Tgrapp***, uma versão optimizada do ***TgraphSpot***, foi desenvolvido para resolver esses problemas de escalabilidade e sobrecarga visual. Através de técnicas como mapas de calor e filtragem de nós irrelevantes, o ***Tgrapp*** consegue reduzir o espaço de características, aumentando a eficiência e clareza dos resultados.

Além disso, foi necessário desenvolver uma ferramenta interactiva, ***CallMine***, que permite aos analistas explorar subgrafos induzidos por vértices suspeitos, identificando padrões de fraude de forma mais dinâmica e visual. O ***CallMine*** facilita a análise de redes de biliões de vértices e arestas, permitindo identificar fraudes sofisticadas e camuflagem de tráfego, como chamadas para números inexistentes, mas com duração significativa.

O Capítulo 6 aborda a importância da visualização interactiva na análise de grandes grafos dinâmicos e como ela pode ser utilizada para apoiar a explicabilidade dos métodos de detecção de fraudes. Com o uso de técnicas de visualização como layouts *spring-embedded* e matrizes de adjacência com reordenação, torna-se possível identificar padrões ocultos em dados complexos. A visualização não só facilita a interpretação dos resultados pelos analistas, mas também ajuda a interpretar as relações entre vértices suspeitos e a caracterizar a estrutura das redes fraudulentas.

A combinação de visualização com as técnicas de detecção do *Tgrapp* e *TgraphSpot* provou ser eficaz na identificação de anomalias que poderiam ser facilmente mascaradas pela grande quantidade de dados. Este capítulo demonstra como as técnicas de visualização avançadas podem servir como uma ferramenta fundamental para os analistas de fraude, permitindo-lhes explorar os dados em tempo real e tomar decisões informadas sobre as acções de mitigação necessárias.

No Capítulo 7, a pesquisa concentra-se na aplicação de **aprendizagem federada** para a detecção de fraudes em ambientes distribuídos, como os que caracterizam as redes 5G. A

aprendizagem federada foi escolhida devido às suas capacidades de treinamento distribuído sem a necessidade de compartilhar dados entre diferentes partes da rede, respeitando assim as questões de privacidade dos utilizadores. Este capítulo apresenta uma comparação detalhada entre modelos federados e centralizados, com ênfase na performance do modelo de aprendizagem automática federado, que superou o modelo tradicional de Random Forest em vários testes.

Os resultados apresentados demonstram que a aprendizagem federada, além de ser eficiente em termos de desempenho (com uma AUC superior a 0.97), consegue operar de forma robusta em ambientes descentralizados, sendo capaz de escalar para redes de grande dimensão e complexidade. Este avanço é crucial para a adaptação de sistemas de detecção de fraude às características únicas das redes 5G, onde a dispersão dos dados e a necessidade de processar informações localmente exigem novas abordagens para garantir a segurança.

Em síntese, esta tese investigou e propôs uma série de métodos avançados para a detecção de fraude em redes de telecomunicações 5G, com foco na análise de grafos dinâmicos e na visualização interactiva de grandes volumes de dados. As soluções apresentadas combinam análise topológica e aprendizagem federada, com destaque para o sistema ***Mondeo***, para a detecção de botnets em múltiplas fases, que se integra de forma eficaz nas redes 5G. Ao utilizar técnicas de análise de grafos foi possivel detectar padrões complexos de fraude que surgem a partir da interacção de múltiplos dispositivos, alguns comprometidos, elevando a precisão na identificação de ameaças distribuídas.

A pesquisa demonstrou que é possível integrar essas metodologias de forma a garantir a eficácia da detecção de fraude, preservando a privacidade dos dados e mantendo o desempenho necessário para o ambiente dinâmico das redes 5G. A contribuição desta tese vai além da detecção de fraude, propondo um modelo de segurança que equilibra a escalabilidade, a privacidade e a capacidade de resposta ágil frente aos novos tipos de ataques cibernéticos que emergem neste novo contexto tecnológico.

**Palavras-Chave:** Fraud Methods, Fraud Networks, Network Control, Bridging Centrality, Influence, 5G Fraud, Data Visualization, Complex Networks, Federated ML

# Contents

# List of Figures

# List of Tables

# Acronyms

**4G** Fourth Generation Mobile Network

**5G** Fifth Generation Mobile Network

**6G** Sixth Generation Mobile Network

**AI** Artificial Intelligence

**AIDA** Adaptive, Intelligent and Distributed Assurance Platform

**C2** Command and Control

**CAPTCHA** Completely Automated Public Turing test to tell Computers and Humans Apart

**CFCA** Communications Fraud Control Association

**DGA** Domain Generation Algorithm

**DoS** Denial of Service

**eMBB** Enhanced Mobile Broadband

**EU** European Union

**FL** Federated Learning

**FMS** Fraud Management System

**FPR** False Positive Rate

**GCN** Graph Convolutional Network

**IAT** Inter-Arrival Time

**IoT** Internet of Things

**IRSF** International Revenue Share Fraud

**MAPE-K** Monitoring, Analysis, Planning, Execution, and Knowledge

**MEC** Multi-access Edge Computing

**ML** Machine Learning

**mMTC** Massive Machine-Type Communications

**MONDEO** Multistage Mobile Network Detection and Tactics

**RAN** Radio Access Network

**SRF** Shared Responsibility Framework

**TPR** True Positive Rate

**URLLC** Ultra-Reliable Low-Latency Communications

# Glossary

**5G edge computing** A paradigm that brings computing resources from centralized cloud or data center environments closer to end users and devices through the 5G network

**bridging centrality** A node metric defined as the product of betweenness centrality and the bridging coefficient, capturing both global and local structural importance

**control** The ability to steer a system from an initial state to a desired final state within finite time

**federated learning** A machine learning approach in which a model is trained across multiple decentralized devices or servers holding local data samples, without exchanging raw data

**influence** A graph-related metric used to quantify the ability of a node to affect other nodes or the overall network structure

]

# Chapter 1

# Impact Definition

## Research Foundations of the Thesis

This thesis is underpinned in the work developed mostly in the context of the AIDA project and a collection of peer-reviewed research papers, in which I was either the primary author or a co-author. These works provide the foundation for the thesis, contributing key methodologies, experimental results, and theoretical insights. Rather than a direct reproduction, this thesis integrates, extends, and adapts their content, incorporating additional refinements and analyses not included in the original publications. Each chapter highlights the relevant paper from the list below that has influenced its development, with varying degrees of adaptation depending on the specific topic.

By building upon these works while incorporating new insights and extended analyses, this thesis presents a comprehensive and structured approach to fraud detection in modern telecommunications networks. It addresses the full spectrum of the problem, starting from the proliferation of malware that enables fraudulent activities, to the identification and mitigation of fraud through advanced detection techniques. Furthermore, this thesis explores a federated learning approach that allows multiple telecom operators, who are often restricted from sharing data due to internal policies or government regulations, to collaboratively benefit from a collective intelligence.

The core contributions of this thesis build upon my prior work in fraud detection within telecommunications networks, leveraging machine learning, graph-based anomaly detection, and federated learning. The methodologies and findings presented in the following papers, which I have authored or co-authored, serve as the foundation for the research presented in this thesis:

- **MONDEO-Tactics5G: Multistage Botnet Detection and Tactics for 5G/6G Networks**

(Elsevier, Computers & Security), doi:10.1016/j.cose.2024.103768

- **Star-Bridge: A Topological Multidimensional Subgraph Analysis to Detect Fraudulent Nodes and Rings in Telecom Networks** (IEEE Big Data), doi:10.1109/BigData55660.2022.10020714
- **TgraphSpot: Fast and Effective Anomaly Detection for Time-Evolving Graphs** (IEEE Big Data), doi:10.1109/BigData55660.2022.10020898
- **TgrApp: Anomaly Detection and Visualization of Large-Scale Call Graphs** (AAAI Conference on Artificial Intelligence), doi:10.1609/aaai.v37i13.27062
- **CallMine: Fraud Detection and Visualization of Million-Scale Call Graphs** (ACM CIKM 23), doi:10.1145/3583780.3614662
- **A Federated Machine Learning Approach to Detect International Revenue Share Fraud on the 5G Edge** (ACM SAC 22) doi:10.1145/3477314.3507322

At the end of each chapter, the specific contributions of each author in the respective publication are explicitly acknowledged, delineating the individual efforts.

## 1.1 Introduction

Telecom fraud significantly impacts operator revenues, with losses estimated at $28.3 billion (USD) annually, as reported by the CFCA in 2021 [24]. To combat this, operators deploy Fraud Management Systems FMS aimed at reducing these losses. However, the effectiveness of these systems is questionable, with more than 28% of telecom operators experiencing a False Positive Rate (FPR) as high as 90%, and 14% reporting an FPR of 97%. On the brighter side, concerning the True Positive Rate (TPR), 62% of operators have seen a TPR of 80%, though 7% report a lower TPR of 60% [24]. As the telecom industry transitions to 5G networks, expected to offer substantial improvements such as higher speeds, reduced latency, and increased device connectivity, the complexity and potential for fraud also rise. This evolution presents a dual challenge: an expanded attack surface and enhanced sophistication of attacks, making the fight against telecom fraud even more complex. Like previous mobile evolutions, the initial immature state of 5G technology creates fertile ground for unknown fraud types and methods. As operators are still developing and implementing security measures for 5G networks, fraudsters are likely to exploit vulnerabilities in areas such as network slice, edge computing, and massive IoT deployments. The increased number of connected devices and virtualized network functions presents new attack vectors that traditional fraud detection systems may not be equipped to handle. This technological shift demands adaptive fraud prevention strategies that can evolve alongside emerging threats.

The heart of the Fraud theory was initially developed in the context of occupational fraud and

described as a triangle of three different factors , Pressure, an incentive or a motive leading to unethical behaviour; Opportunity, a weakness or lack of monitoring that makes fraud possible; and Rationalism, a moral justification for unethical actions. Later Capability [5] was added as the fourth element of fraud, complementing the previous prerequisites by adding that, to perpetrate fraud, it is necessary to have the set of skills, traits, and resources to do it.

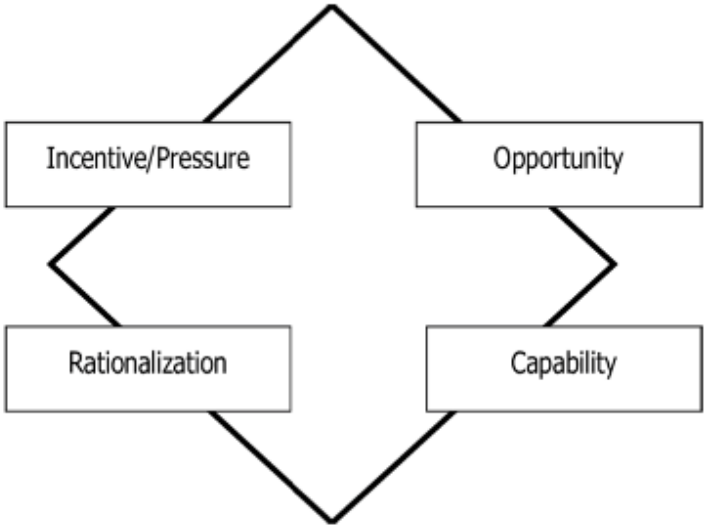



**Figure 1.1:** The Fraud Diamond, Capability is added as a fourth dimension.

This fourth dimension changed the concept from a triangle to a diamond [53]. The aim of this thesis is to explore the capability dimension by designing malware propagation and abnormal behaviour algorithms, and propose a 5G edge computing environment where the associated algorithms can be deployed across distributed base stations using local data, enabling privacy-preserving fraud detection while minimizing computational overhead and latency.

Finally, the effectiveness of these algorithms was tested in a novel federated machine learning framework, using IRSF (International Revenue Share Fraud) fraud detection, that successfully processes high-volume data streams through 5G edge computing, enabling scalable distributed fraud detection across base stations while preserving data privacy and optimizing resource usage, with experimental results demonstrating comparable performance to centralized approaches.

## 1.2 Context and outline of the present thesis

In the current technological landscape, especially within the telecommunications sector, the European Union is increasingly concerned about mitigating risks and threats in 5G networks. In January 2020, the European Union published a Coordinated Risk Assessment Report on 5G [72], underscoring the significant security risks and emerging threats expected to intensify

with the deployment of 5G networks. Both the report and the subsequent Council Conclusions emphasized the evolving security challenges inherent to the 5G landscape. These challenges are related to the availability, integrity, confidentiality, and privacy of the networks, the key innovations and wide range of services enabled by 5G, and the complex role of suppliers in the construction and operation of the 5G network. The report emphasizes the necessity of reassessing current policy and security frameworks to address these new security paradigms and the importance of implementing appropriate mitigating measures at both national and European levels. The AIDA project (Adaptive, Intelligent, and Distributed Assurance Platform) is set to play a crucial role in addressing these security challenges, specifically risk scenarios related to the modus operandi of major threat actors, such as the exploitation of 5G networks by organized crime targeting end-users, and the risks posed by IoT devices and smart devices in 5G environments.

This project represents a concerted effort by industry leaders and a consortium of academic partners from diverse fields, aiming to tackle the challenges of 5G fraud and distributed platform components. With a grant of USD 2.1M (€1.8 million) from the EU, AIDA aims to efficiently scale fraud management and network security programs as telecom operators progress in their digital transformation. The focus is on distributing platform components using edge computing and 5G, exploring federated machine learning techniques, testing resilience to intrusion or tampering, and ensuring data privacy and confidentiality. The primary goal is to provide a comprehensive 5G-ready fraud management platform that can effectively handle the security and analytics demands of the 5G era.

In this context, the thesis presents a significant advancement in the application of mixed and composite algorithms, alongside machine learning, for fraud detection in the telecommunications sector. By introducing innovative methodologies such as federated learning, anomaly detection in time-evolving graphs, and multistage botnet identification this work addresses key challenges related to data privacy, scalability, and model performance. The proposed visualisation techniques further enhance interpretability and facilitate analyst interaction, making the systems more accessible and operationally effective.

Moreover, the methods developed not only yield strong results in detecting fraud and identifying critical nodes within fraud networks, but also offer valuable insights into the behavioural patterns underlying these activities. The broader contribution of this thesis lies in its potential to inform the design of more robust and intelligent fraud detection systems, while also highlighting ongoing challenges. These include the need for enhanced privacy safeguards, scalable architectures, improved visualisation of large-scale data, reliable performance in federated environments, and more advanced botnet detection capabilities suited to 5G and emerging 6G networks.

This thesis is structured into three main sections. The initial section, which encompasses Chapter 1 and Chapter 2, delves into impact analysis and provides an overview of the current state of the art.
The second section, which forms the core of the thesis, spans from Chapter 3 to Chapter 7. It begins with Chapter 3, focusing on the identification and strategic response to rogue devices infected with malware, identified as a primary enabler of fraud. Recognising that malware prevention is essential but not always possible due to devices already infected before joining operators network, Chapters 4, 5, 6 focus on static and dynamic algorithms designed to detect abnormal behaviour and fraud within large-scale network graphs, notably in who-calls-whom networks.
Chapter 7 shifts the focus to a distributed approach, leveraging edge computing and federated learning architecture, which is capable of handling the massive data volumes of 5G networks while maintaining privacy and not compromising accuracy of fraud detection algorithms when compared to centralised models. The thesis concludes with Chapter 8, which provides a conclusive overview and directions for future research.

### 1.2.1 Mobile Malware Fraud Enabler

The increasing threat of mobile malware in the telecommunications sector is a critical issue, particularly since it is increasingly becoming a facilitator of various telecom fraud activities. This type of malware, encompassing viruses, trojans, and worms, can stealthily penetrate mobile devices, leading to a spectrum of harmful consequences. These include the theft of sensitive personal and financial data and the commandeering of the device's communication functions. Such capabilities make mobile malware a formidable instrument in the hands of fraudsters, enabling them to perpetrate telecom fraud schemes such as unauthorized account access or account takeover, interception of communications, and premium-rate service scams.
The spread of mobile malware is driven by increasingly sophisticated attack techniques, the availability of malware-creation tools on the Dark Web, and the prevalent use of mobile devices in financial activities. With the ongoing evolution of the telecommunications industry, especially with the advent of 5G and IoT technologies, the threat of mobile malware as an enabler of telecom fraud is expected to intensify.
To address the growing challenges of phishing scams, the Shared Responsibility Framework SRF [96] has been introduced, revolutionizing the role of financial institutions and telecommunication companies from passive to active participants in fraud prevention. This framework demands that financial institutions and telecom operators adopt a proactive stance in combating these scams, assigning them specific and actionable responsibilities. Furthermore, the SRF establishes a mechanism to compensate victims, ensuring accountability and forcing these institutions to diligently fulfill their newly defined roles in safeguarding against such

fraudulent activities.
By detecting and limiting the impact of these devices, the spread of malware and fraud can be significantly reduced. The current increase in phishing scams and rogue mobile apps is identified as major fraud facilitators, this is especially relevant in the 5G context [91]. Chapter 3 delves into the detection of Command and Control (C2) servers and botnets, as the key mark to recognize infected devices, illustrating how their identification can preempt potential threats and facilitate mitigation strategies. We propose a comprehensive strategy for addressing malware threats using the MONDEO (Multistage Botnet Detection and Tactics) architecture and the Rainbow [22] framework. Detection is split into four stages: whitelisting/blacklisting, query rate analysis, domain generation algorithm (DGA) detection, and machine learning-based anomaly and pattern recognition. Upon detecting malware, the system, through the MONDEO architecture, evaluates the appropriate response strategies using Rainbow's self-adaptive framework. This involves the MAPE-K loop (Monitoring, Analysis, Planning, Execution, and Knowledge) [15], which helps in assessing the system's state and deciding on mitigation actions. The mitigation tactics considered include isolating the affected mobile device, redirecting the C2 server to a harmless IP address, and prompting users to complete a CAPTCHA when a suspicious connection is detected. To simulate real-world scenarios, malware-infected applications were tested on an isolated system, using samples from sources like MalwareBazar [6] and Koodous [54], on a device emulating the Pixel 4 with Android API 29.

### 1.2.2 Static Topological Multidimensional Subgraph Analysis

The intelligence and masking capabilities of malware cannot be underestimated, as secure protocols and virtual private networks may hide suspicious traffic. Moreover, devices could already be infected by previous users or networks, with dormant malware not exhibiting activity at the time but still enabling fraudulent operations. Lastly, even though malware and rogue applications are the key enablers of fraud, there are many other attack vectors, such as subscription fraud, account hijacking, and internal fraud, to name a few. Algorithms to identify abnormal behaviour and detect fraud represent the next level of detection. In Chapter 4 we propose Starbridge, a multidimensional static topological subgraph analysis to detect fraudulent nodes and rings in telecom networks.
Fraud evolved from individual actions to complex group operations,commonly refereed as fraud rings requiring better ways to identify suspicious nodes in these networks. Our method ***Starbridge*** introduces a novel approach that focuses on categorizing nodes within a network based on their functional role and level of influence. By examining the layout and arrangement of nodes within a network, we can uncover patterns of connections, strengths, and unique characteristics typically linked to certain network types. Fraud is a composite force driven by a combination of technical strategies and human behaviour. Within such networks, strategically

positioned nodes are pivotal in exerting control, remaining concealed, and maintaining the network's stability and success. The significance of each node and its role can be assessed through local and global network metrics. To this end, we propose the integration of Bridging Centrality, Control, and Influence as key measures within the network. The two initial metrics depend on the network's structure, while the third assesses a node's influence relative to its neighbors. Together, these measures assign a score to each node based on its role, local impact, and central position, thus providing a comprehensive evaluation of node significance from multiple perspectives. Furthermore, our research demonstrates that a combination of various factors, such as a node's role and influence score, is highly effective in identifying nodes with a greater likelihood of engaging in fraud. This approach has been corroborated using real data from a fraud management system, underscoring its practical applicability. Notably, nodes ranked within the top 25 percentile in terms of influence were found to be more prone to fraudulent behaviour, confirming the relevance of these metrics in detecting and understanding fraud within networks.

### 1.2.3 Dynamic Analysis of Fraud Networks in Time-Evolving Graphs

While static analysis has proven effective in identifying fraudulent nodes and patterns and reveal modus operandis within a network's snapshot, they may not fully address the challenges posed by abnormal behaviour. There is a growing need for continuous improvement in dynamic network analysis algorithms, given that relying solely on static analysis may not adequately address the complexities of rapidly changing fraud enablers and methods. In addition, analysts face challenges in detecting and interpreting anomalies within dynamic networks, especially when fraudsters use camouflage techniques to hide suspicious behaviour. These two concepts are the focus of chapter 5.

The dynamic analysis of fraud in time-evolving graphs introduces a paradigm shift from conventional static methods. This shift is crucial because traditional approaches often do not adapt to the rapid changes and complexities inherent in modern telecommunication networks. Our approach delves into scenarios without predefined labels, analyzing millions of phone call records, each marked with essential details such as source, destination, timestamps, and call duration. The aim is to not only locate anomalous activities and identify potential fraudsters but also to adjust to the network's temporal changes, enhancing the effectiveness and accuracy of fraud detection.

Two significant evolutions in the realm of Dynamic Analysis of Fraud in Time-Evolving Graphs are the development of ***TgraphSpot*** and ***CallMine***. ***TgraphSpot*** serves as an advanced method specifically tailored to navigate and analyze the complexities of dynamic graphs, providing a robust framework for detecting fraud in evolving networks. This method is structured into three interconnected steps: feature selection, an interactive high-level summary of the data, and a deep-dive analysis focusing on suspicious nodes. Similarly, ***CallMine*** rep-

resents an evolution from ***TgraphSpot***, designed to excel in dynamic graph analysis, even with imperfect labels. Its robustness in supervised settings distinguishes it from traditional methods, providing a more resilient approach to the sophisticated camouflage techniques used by fraudsters.

These advances underscore the importance of dynamic analysis in the current telecommunications landscape. They offer innovative tools and strategies for detecting and understanding various anomalies within large-scale dynamic 'who calls who' graphs. By providing methodologies that not only identify current fraudulent activities but also predict and adapt to future changes in network structure and behaviour, ***TgraphSpot*** and ***CallMine*** set the stage for a more comprehensive exploration into dynamic analysis methods suitable for time-evolving graphs. This shift towards dynamic analysis is crucial in maintaining the effectiveness of fraud detection strategies, particularly in scenarios where deceptive practices are employed to evade traditional detection methods.

### 1.2.4 Leveraging Edge Computing and Federated Machine Learning for Fraud Detection

5G networks are expected to handle significantly more data compared to 4G. According to Opensignal, 5G smartphone users consume between 2.7 and 1.7 times more mobile data than 4G users [76]. This increase is attributed to the much faster speeds and additional capacity provided by 5G, which allows operators to access higher frequency bands previously unused for mobile services. 5G is capable of supporting about one million devices per square kilometer compared to 4G's support for approximately 4,000 devices, enabling more extensive and uninterrupted use of telecom services. However, these advancements present their own set of challenges in the context of fraud detection.

The escalation in data volumes brought by 5G networks emphasizes the need for a distributed architecture, which offers enhanced capabilities for content processing and storage, operating in close proximity to mobile users with minimal latency. Additionally, multiple reasons can enforce distributed architectures mainly related to user privacy laws or government restrictions that limit any share of data between edges.

This decentralization of data processing through multiple edges, raises a concern on the performance of fraud detection algorithms especially when compared with traditional centralized models (ML). This is the last part of this thesis, and the focus of chapter 7, propose a framework which is able to run our algorithms using decentralized data processing and preserve user privacy, yet ensure the same performance has centralized models.

We propose the ***OptiEdge*** framework which leverages Edge Computing and employs Federated Learning to train modules across edges. Edge computing offers the potential to host applications proximal to users, thereby reducing communication latency and augmenting application performance and energy efficiency. Particularly, Multi-access Edge Computing

(MEC) integrates Cloud Computing and IT capabilities within the Radio Access Network (RAN), situating these services near mobile subscribers. MEC technology thus provides a distributed environment for application and service provisioning, as well as content processing and storage, in close proximity to mobile users. By integrating Machine Learning (ML) applications with Federated Learning, we aim to train a universal ML model across all network edges. A critical aspect of this framework is its respect for privacy; it enables individual fraud systems to detect fraudulent calls autonomously, without the need for data sharing among systems, and with minimal computational resource consumption. Results show that the proposed Federated ML framework can deal with the training and deployment of ML models within a MEC scenario using decentralized data and lighter ML algorithms. Our findings indicate that the decentralized model, employing Federated Learning, achieved a remarkable level of class discrimination, consistently exceeding 0.97. Furthermore, this model successfully sustained its performance throughout two distinct rounds of analysis.

### 1.2.5 Conclusions

In Chapter 8, we bring this thesis to a close by emphasizing the key contributions of the methodologies we have developed. We highlight the significance of our findings and how they contribute to the broader field. This chapter also outlines the expected impact of our work and proposes future pathways for investigation and research, laying the groundwork for further academic exploration in this area.

## 1.3 Overall Contributions and Impact

This thesis addresses the problem of fraud detection in modern telecommunications networks, with a particular focus on the structural, behavioural, and systemic challenges introduced by fifth-generation (5G) mobile environments. As 5G networks significantly increase connectivity, data volume, and service diversity, they simultaneously expand the attack surface and enable more sophisticated, distributed, and adaptive fraud schemes. Traditional fraud detection approaches, typically based on centralized processing, static rules, and isolated event analysis, are increasingly inadequate in this new paradigm.

The central contribution of this thesis is the design and validation of a holistic, multi-layered fraud detection framework that redefines how fraud is modelled, detected, and operationalised in large-scale telecom systems. Rather than treating fraud as isolated anomalous events, this work frames fraud as an emergent property of complex systems, arising from the interaction of compromised devices, coordinated actors, and evolving behavioural patterns within massive communication graphs.

To address this, the thesis proposes an integrated pipeline spanning four complementary dimensions.

First, it introduces a multistage malware and botnet detection strategy (MONDEO), targeting the primary enabler of telecom fraud: compromised devices. By modelling malware activity across multiple stages, from initial infection to command-and-control communication and fraud execution, the approach enables early detection and proactive mitigation. This contribution shifts fraud detection upstream, preventing fraudulent activity before it manifests at scale.

Second, the thesis advances graph-based structural analysis through the StarBridge methodology, which identifies fraudulent entities not only by their individual behaviour but by their functional role within the network topology. By combining metrics such as bridging centrality, influence, and control, the approach uncovers organized fraud structures, including fraud rings and hierarchical coordination patterns. This represents a shift from node-level classification to role-based detection in complex networks, enabling a deeper understanding of fraud organisation and modus operandi.

Third, the work extends fraud detection into the temporal domain through dynamic anomaly detection in time-evolving graphs (TGraphSpot and CallMine). These methods capture behavioural patterns that unfold over time, allowing the identification of short-lived, bursty, or camouflaged fraud strategies that static approaches fail to detect. By leveraging robust statistical features and scalable visual analytics, the thesis provides tools capable of analysing millions of nodes and interactions while maintaining interpretability for analysts.

Fourth, the thesis proposes a distributed fraud detection architecture based on edge computing and federated machine learning, enabling collaborative intelligence across network edges without sharing raw data. This approach directly addresses the constraints of 5G environments, including data decentralisation, privacy requirements, and latency constraints. Experimental results demonstrate that federated models achieve performance comparable to centralized approaches (AUC > 0.97), validating the feasibility of scalable, privacy-preserving fraud detection in real-world deployments.

Beyond individual contributions, the thesis delivers a unified architectural vision that integrates these components into a coherent system aligned with the operational realities of telecom networks. It demonstrates how malware detection, graph analytics, temporal analysis, and distributed learning can be combined into a continuous detection loop, capable of adapting to evolving threats while maintaining scalability and explainability.

From a scientific perspective, the thesis contributes to the fields of complex networks, graph mining, and anomaly detection by introducing new methodologies for role-based analysis, temporal feature engineering, and large-scale graph exploration. From an engineering and

operational perspective, it provides practical, deployable solutions for telecom operators, addressing key industry challenges such as high false positive rates, limited visibility into coordinated fraud, and the need for real-time decision-making at scale.

Ultimately, this work demonstrates that effective fraud detection in 5G networks requires a shift from static, centralized models to adaptive, distributed, and system-aware approaches. By combining insights from network science, machine learning, and telecom systems engineering, the thesis establishes a foundation for next-generation fraud detection systems capable of operating in increasingly complex and adversarial environments.

# Chapter 2

# State of the Art

## 2.1 Background and Related Work

### 2.1.1 Mobile Malware Attacks and Characteristics

Mobile malware has emerged as a significant enabler of fraud, presenting a multifaceted threat landscape that combines high technical sophistication with the strategic capabilities inherent in the fraud diamond framework. In the context of fraud, capability refers to the possession of the necessary characteristics, abilities, or skills to conduct fraudulent activities. It represents the point at which a fraudster identifies specific opportunities for fraud and possesses the means to execute them effectively. The supporting components of capability, including position, intelligence, coercion, deception, business and domain knowledge as well as technical skills, play crucial roles in enabling fraudulent activities. However, not all individuals with motive, opportunity, and rationalization possess the capability required to successfully execute fraud. This distinction is especially relevant in the context of mobile malware in telecommunications, where fraud facilitation depends on the convergence of advanced technical expertise and strategic acumen. This chapter presents the current state-of-the-art methods for defending against mobile malware, detecting abnormal behavior, and identifying telecom fraud.

Mobile malware, characterized by malicious software specifically targeting mobile devices, has witnessed a staggering surge in sophistication and frequency in early 2022, with attacks escalating by 500% in February alone [79]. These malicious entities pose multifaceted threats, not only aiming to breach privacy and disrupt organizational operations but also enabling various forms of fraud, notably within the telecom domain. Among the fraudulent activities facilitated by mobile malware are International Revenue Share Fraud (IRSF), Application to People (A2P) Bypass, SMS Bypass, Banking Fraud, and Click Fraud, underscoring the diverse range of illicit activities enabled through compromised devices. The proliferation of smart-

phones in modern society has revolutionized communication and accessibility to information, yet this increased reliance on mobile devices has made them prime targets for cybercriminal exploitation. Consequently, the development of robust detection and prevention mechanisms becomes paramount, not only to counter the spread of mobile malware but also to mitigate the risks of fraud perpetrated through compromised devices. The consequences of mobile malware infections extend beyond individual device compromise, as compromised smartphones serve as powerful tools for cybercriminals to orchestrate identity theft, financial fraud, and unauthorized access to sensitive information. Therefore, proactive measures must be implemented to effectively combat the evolving threat landscape posed by mobile malware and safeguard users digital security and financial integrity.

Bots and botnets play a significant role in enabling fraudulent activities, amplifying the impact of mobile malware threats. Cybercriminals utilize bots for various malicious purposes, including SIM card registration, call and message routing, phishing, hijacking one-time passwords, and more. Complicating matters further, bots can emulate human behavior patterns, making detection of suspicious activity challenging. Cybercrime-as-a-service (CaaS) has emerged as a favored option for individuals and organizations seeking to launch sophisticated digital attacks against enterprises. With CaaS causing $6 trillion [57] in damage last year alone and experts predicting a breach of some 33 billion accounts in 2023 [57], the threat landscape continues to escalate. Organized through online platforms and marketplaces, both on the internet and the dark web, CaaS offers services such as malware creation, exploit kits, malicious bots, and other automated attack tools for sale. This accessibility to 'ready-made' attack tools has democratized the use of bots to target company websites, putting customer data at risk and potentially leading businesses into financial distress, while also increasing the likelihood of compromised customer data being utilized in subsequent attacks, leveraging its credibility to further enhance the success rates of cybercriminal endeavors and exacerbating financial losses. As cyberattacks become increasingly sophisticated, the prevalence of CaaS is expected to persist and thrive, posing significant challenges to cybersecurity across various sectors, including telecommunications, finance, and e-commerce.

While infected mobile devices have common characteristics such as battery drainage, increased data usage, appearance of uninstalled apps [34], frequent crashes, and unauthorized payments [127], which can alert end users to potential malware infection, the reality is that malware is becoming increasingly sophisticated, employing new evasion techniques and mechanisms to remain concealed and execute its functions without detection. Furthermore, mobile malware enables attackers to make calls or send SMS from infected devices, exploiting additional monetized fraud opportunities through voice calls or SMS. This rising threat is underscored by a report showing that in the last 12 months of 2021, three-quarters of consumers received scam emails, 66% via text message, 58% via phone, and 15% through messaging apps, illustrating the extensive exposure of the subscriber base [38]. A notable example is the Exodus Spyware discovered in 2019, which targeted iPhones and Android phones, gaining

root permissions and acquiring complete control over the devices [43].
As the numbers of mobile malware attacks and associated fraudulent activities continue to rise unabated, it becomes increasingly evident that current solutions are insufficient in addressing the evolving threat landscape. Despite efforts to combat fraud, fraudsters persistently find ways to bypass existing security measures, while a lack of user awareness further exacerbates vulnerabilities. In light of these challenges, there is a pressing need for telecom-level solutions capable of identifying and mitigating botnets, safeguarding not only end-users but also telecom operators and other partners from potential fraud.

### 2.1.2 Mobile Malware Detection Techniques

The detection of malware is paramount in the contemporary landscape of mobile cybersecurity, especially with the incessant rise in mobile device usage. Among the myriad of malware types, the focus here lies predominantly on identifying bots and trojan horses, given their prevalence in facilitating telecom fraud. While various methodologies exist, a fundamental dichotomy emerges between static and dynamic analysis, each offering distinct advantages and limitations. Most research work propose a hierarchical approach, wherein static and dynamic analysis serve as subcategories to signature and anomaly-based techniques, providing a structured framework for understanding malware detection methodologies. Figure 2.1 illustrates the multiple types of Malware and mobile malware detection techniques.

To address this challenge, latest researcher have employed static analysis approaches aimed at identifying and classifying malicious applications based on their characteristics and behavior. These methods have undergone significant evolution, evolving into sophisticated approaches with diverse strengths and weaknesses. Conversely, dynamic analysis techniques entail the execution of mobile applications within a controlled environment to scrutinize their behavior in real-time. Although dynamic analysis holds promise in detecting sophisticated malware variants, challenges persist, particularly in identifying evasive techniques employed by cybercriminals. This is the main reason why Machine Learning techniques have gained prominence in malware detection, not only due to their ability to analyze large datasets and identify complex patterns indicative of malicious behavior, but also by leveraging features extracted from both static and dynamic analysis, an hybrid approach using machine learning models which can effectively classify mobile applications as benign or malicious. This dichotomy sets the stage for a comprehensive exploration of contemporary malware detection strategies, underscoring the necessity of adopting a multifaceted approach to combat evolving threats effectively.

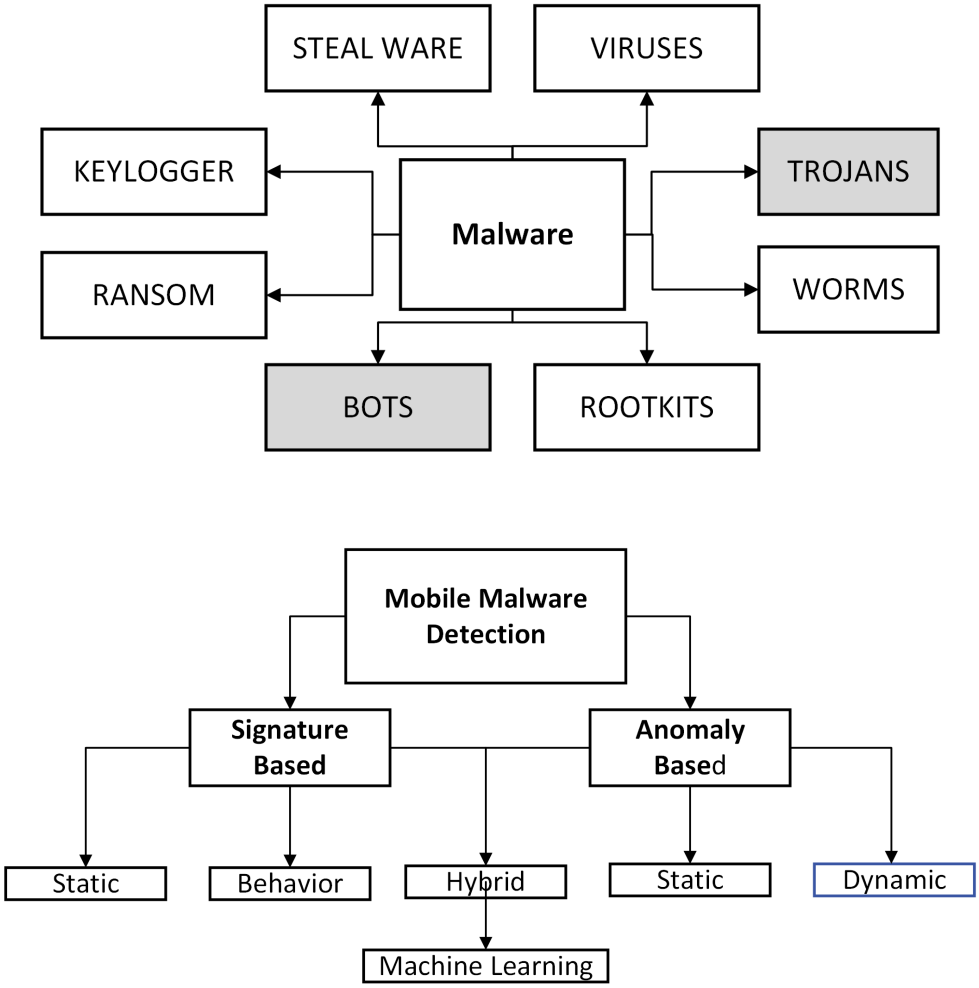



**Figure 2.1:**
Types of malware (top) and Malware Detection Techniques (Bottom)

### 2.1.3 Signature-based Detection

Signature-based detection techniques involve the identification of known malware patterns or signatures within mobile applications. These methods rely on predefined signatures or patterns extracted from known malware samples to detect malicious behavior.

Behavior signature-based detection methods analyze the runtime behavior of mobile applications to detect malicious activities. These techniques observe system calls, network traffic, and other runtime activities to identify patterns indicative of malware infections

#### Static Malware Detection

Static analysis techniques in mobile malware detection involve examining the characteristics and behavior of applications without executing them. These methods offer valuable insights into the code structure, permissions requested, and other static attributes of the applications. Despite their effectiveness in identifying known malware variants, static analysis approaches have certain limitations, particularly in detecting novel and zero-day threats. This section

explores various static analysis methods employed in Android malware detection, highlighting their strengths and weaknesses in combating telecom fraud.

Among the prominent methods, Akram et al. [12] proposed an innovative approach that integrates attacker information into static analysis, resulting in improved accuracy rates of up to 98%. Similarly, Faruki et al. [35] utilized syntactic foot-printing techniques to classify apps as malware or benign, achieving a poor accuracy rate of 60%. Song et al. [98] introduced a comprehensive static framework with multi-layer filtering techniques, achieving remarkable accuracy levels of nearly 99

Additionally, Akhtar et al. [8] presented a permission-based malware detection system employing machine learning classifiers, which demonstrated an impressive accuracy rate exceeding 94%. Wu et al. [113] introduced a novel approach utilizing permission and API data to classify apps as normal or malicious, achieving a high accuracy rate of 97.87%. Furthermore, Talha et al. [101] developed a permission-based detection system with an accuracy rate of 88%, showcasing the effectiveness of static analysis in identifying potentially harmful applications.

While these static analysis approaches offer promising results, challenges remain, particularly in the detection of new and unknown malware variants. Signature-based methods, although effective in identifying known malware families, struggle to detect novel threats. In contrast, permission-based approaches demonstrate a lower false positive rate but may overlook sophisticated malware variants that exhibit deceptive behavior. Static analysis methods also tend to rely heavily on the availability of signatures or predefined rules, limiting their effectiveness against zero-day attacks and polymorphic malware.

One of the drawbacks of static analysis approaches is their limited ability to protect telecom subscribers from getting their devices infected and becoming victims of telecom fraud, as they primarily focus on identifying and classifying malware based on predefined patterns or rules, rather than actively preventing infections and fraudulent activities.

**Behavior Signature-based Detection**

In this approach, signatures are acquired during the analysis of malware source code decomposition in static signature-based techniques, while dynamic behavior-based techniques acquire signatures after executing malicious code. Chen et al. [21] proposed a detection approach that identifies threat patterns by analyzing function invocation and data flow to detect malicious behaviors in Android devices, achieving a 91.6% detection rate over 252 malicious samples.

### 2.1.4 Anomaly-Based Detection

Anomaly-based methods adopt a less stringent approach by observing a device's normal behavior over a certain period and using those metrics as a baseline to detect deviations. Both static and dynamic methods are employed in the analysis. Static analysis dissects an app before installation, while dynamic analysis occurs during app execution, gathering data such as system calls and events. Anomaly-based detection comprises training and detection phases, with the former observing a non-infected system operating normally and the latter identifying anomalies from the training period model.

#### Static Anomaly-based Detection

Static anomaly-based detection methods analyze the code and structure of mobile applications to identify suspicious patterns or behaviors without executing them. By comparing an application's characteristics against a baseline of normal behavior, these techniques can detect deviations indicative of malware.

Wu et al. [112] developed DroidMat, offering malware detection through manifest and API call tracing. They achieved high accuracy in detecting mobile malware using a static analysis approach.

#### Dynamic Malware Detection

Dynamic analysis techniques involve the execution of mobile applications in a controlled environment to observe their behavior at runtime. By monitoring system calls, network traffic, and other runtime activities, dynamic analysis methods can detect malicious behavior indicative of malware infections. However, dynamic analysis approaches may face challenges in identifying sophisticated malware variants that employ evasion techniques to evade detection. This section delves into dynamic analysis methods used in Android malware detection, emphasizing their role in mitigating telecom fraud risks while acknowledging their limitations in addressing advanced threats.

Authors in [64] proposed a machine learning-based approach to identify malware using dynamic analysis to extract features of system functions. They employed J48 decision tree and Naïve Bayesian classifiers to train and test the classifiers. Results showed a classification accuracy of 90% in detecting malware Android applications with a low false positive rate. In [36], a novel model comprising dynamic and static analysis, machine learning, and local and remote host components is utilized to identify Android malicious operations. The model exhibits high accuracy in malware detection due to its efficiency in terms of storage consumption and power, achieving a 99% accuracy rate. Back propagation Neural Network is employed by authors in [36] to identify illegitimate applications based on the application system call sequence. A static Markov Chain replicates the system call sequence, with transi-

tion probabilities differing between illegitimate and normal applications. The results showed an F-score of 0.982773. Yerima et al. [119] employed ensemble learning for Android malware detection, providing robustness and resilience to code obfuscation, resulting in a detection accuracy of 97

While dynamic analysis techniques offer advantages such as the ability to detect zero-day attacks and polymorphic malware, they also have limitations. Dynamic analysis can be resource-intensive and may impact device performance, especially on resource-constrained mobile devices. Additionally, dynamic analysis may struggle to capture complex malware behaviors that require prolonged observation or interaction with specific environmental conditions. This limitation may hinder the ability to protect telecom subscribers from becoming victims of telecom fraud, as the dynamic analysis may not be able to detect and prevent sophisticated fraudulent activities in real-time.

### 2.1.5 Hybrid Anomaly-based and Signature-based Detection

Hybrid anomaly-based detection methods combine both static and dynamic analysis approaches to enhance detection accuracy. By leveraging both code analysis and runtime behavior monitoring, these techniques can effectively identify sophisticated malware variants.

Hanlin et al. [42] presented ScanMe mobile, a cloud-based Android malware analysis service that performs both static and dynamic analysis on app package files. The service allows users to compile a comprehensive report and share it via a web interface, effectively combining static and dynamic analysis.

Hybrid signature-based detection methods combine both static and dynamic analysis approaches to improve detection accuracy. By leveraging both code analysis and runtime behaviour monitoring, these techniques can effectively identify a wide range of malware variants.

Enck et al. [31] proposed Kirin, a security service for the Android Operating System (OS), which certifies an app at installation using a set of security rules. Kirin evaluates the app's security configuration against predefined security rules, effectively combining static and dynamic analysis.

### 2.1.6 Machine Learning Approaches

Machine learning techniques have gained prominence in malware detection due to their ability to analyze large datasets and identify complex patterns indicative of malicious behavior. Leveraging features extracted from static and dynamic analysis, machine learning models can effectively classify mobile applications as benign or malicious. While machine learning-based

approaches offer high accuracy rates in malware detection, they may struggle with detecting zero-day threats and evolving malware variants. This section explores the use of machine learning algorithms in Android malware detection, highlighting their potential in combating telecom fraud while recognizing their susceptibility to advanced evasion techniques employed by cybercriminals.

Yuan et al. [123] proposed an approach using Deep Learning for malware identification in Android phones. Deep learning, a burgeoning aspect of machine learning research, is employed to extract over 200 features from both static and dynamic analyses of each Android application, achieving an accuracy of 96.60%. Hasegawa and Iyatomi [44] proposed a lightweight Android malware detection method utilizing one-dimensional convolutional neural networks. Illegitimate applications can be identified with an accuracy of 97%. Authors in [124] proposed an approach using Artificial Immunity based on detector set artificial immune system (MAIS) for the discovery of illegitimate applications in mobile computing devices. Results showed 93.33% accuracy with reduced false negative rates. Li and Jin [61] presented an approach to identify Android Malware Detection based on Feature Codes, achieving high accuracy and a low false positive rate.

While machine learning approaches offer the potential for high accuracy in malware detection, they also have limitations. Machine learning models may require extensive training on large datasets, which can be time-consuming and resource-intensive. Additionally, machine learning models may be susceptible to adversarial attacks or evasion techniques employed by sophisticated malware authors. Furthermore, machine learning models may struggle to generalize well to new and unseen malware variants, requiring frequent updates and retraining to maintain effectiveness. This limitation may impact the ability to protect telecom subscribers from becoming victims of telecom fraud, as the machine learning models may not be able to adapt quickly to emerging fraudulent activities and evolving malware threats. Table 2.1 compares the mobile malware detection techniques across various studies.

**Table 2.1:** Comparison of Mobile Malware Detection Techniques across Various Studies.

| Reference | Sig-St | Sig-Bh | Sig-Hy | Ano-St | Ano-Dy | ML-S | ML-A | Loc-Scale |
|---|---|---|---|---|---|---|---|---|
| Akram et al. (2013) | | | | | | | | Ext-Mob |
| Faruki et al. (2013) | | ✓ | | | | | ✓ | Dev-Mob |
| Song et al. (2014) | Y | | | | | | | Ext-Mob |
| Akhtar et al. (2017) | | | | Y | | Y | | Dev-Mob |
| Wu et al. (2016) | Y | | | | | Y | | Dev-Mob |
| Talha et al. (2016) | Y | | | | | Y | | Dev-Mob |
| Li and Jin (2018) | | Y | | | | | Y | Dev-Mob |
| Faruki et al. (2015) | Y | | | | | | | Dev-Mob |
| Yerima et al. (2019) | | | | | Y | | Y | Dev-Mob |
| Yuan et al. (2017) | | Y | | | | | Y | Ext-Tel |
| Hasegawa et al. (2018) | | Y | | | | | Y | Dev-Mob |
| Zhang et al. (2018) | Y | | | | | Y | | Dev-Mob |
| Li (2019) | Y | | | | | Y | | Dev-Mob |
| Hanlin et al. (2020) | Y | Y | Y | | | Y | Y | Ext-Tel |
| Enck et al. (2010) | | Y | Y | | Y | | | Dev-Mob |
| MONDEO-Tactics5G | | | | | Y | | Y | Ext-Tel |

**Legend:** Ref.: Reference; Sig-St: Signature-Static; Sig-Bh: Signature-Behaviour; Sig-Hy: Signature-Hybrid; Ano-St: Anomaly-Static; Ano-Dy: Anomaly-Dynamic; ML-S: Machine Learning (Signature); ML-A: Machine Learning (Anomaly); Loc-Scale: Execution Location — (Ext)ernal vs (Dev)ice and (Tel)co-scale vs (Mob)ile scope.

After surveying current literature and technological approaches in the realm of carrier-grade cybersecurity solutions (Loc-Scale = Ext-Tel) in table 2.1 , it becomes apparent that there exists a gap in effectively managing and mitigating advanced network threats in emerging 5G environments. Traditional methods, such as those discussed in papers by Yuan et al. (2017) [123] and Hanlin et al. (2020) [42] , focus primarily on leveraging deep learning and hybrid analysis techniques. However, they often lack the comprehensive integration of real-time, multistage detection and response strategies that are crucial for modern networks.

MONDEO-Tactics5G presented in chapter 3 stands out by offering a multilayered detection framework specifically designed for the complexities of 5G network architectures. It not only employs machine learning algorithms for anomaly and signature-based detection but also integrates tactical response mechanisms that can dynamically adapt to the evolving threat landscape. This approach is particularly necessary for modern telecom environments where the scale and variety of potential security threats require a robust, scalable, and adaptive security infrastructure.

In essence, MONDEO-Tactics5G fills this critical gap by providing a solution that not only detects a wide range of sophisticated threats but also manages and mitigates them through proactive measures. Such a comprehensive solution is vital for ensuring the security and integrity of carrier-grade networks, particularly as they evolve towards more open and interconnected 5G and B5G implementations. This makes MONDEO-Tactics5G a necessary advancement over existing solutions, positioning it as a pivotal technology in the future of network security.

## 2.2 Graph Based Anomaly detection

The 2023 CFCA Global Fraud Loss Survey reveals a significant increase in telecommunications fraud, with a reported 12% rise from previous years. This increase has led to an estimated loss of $38.95 billion in 2023, accounting for about 2.5% of total telecommunications revenues [23]. The survey indicates that the main drivers of this uptick in fraudulent activities include sophisticated methods such as subscription fraud and identity theft, exacerbated by changes in customer engagement processes and the adoption of new technologies by both service providers and fraudsters. Detecting telecom fraud within the constantly evolving landscape of telecommunications presents significant challenges.First, modern telecom fraud spans beyond traditional voice calls and SMS, extending into arenas involving social media, specialized hardware, and customized applications. Second, advanced camouflage techniques make fraud detection hard to distinguishing between genuine customer activities and deceptive ones. And last, fraud mechanisms have evolved from isolated actions performed by single individuals to complex criminal networks. These networks not only possess sophisticated technological capabilities but also significant financial resources that enable them to orches-

trate elaborate fraud schemes. Their ability to fund these operations and sustain them over time presents an ongoing challenge for telecommunications companies.
The necessity for enhanced detection capabilities in complex data environments, where fraud typically manifests through inherent interconnectivity and relational dynamics, has led to various analytical techniques which have been organized into five major groups, each encompassing a range of specific methods tailored to address the complexities of detecting fraudulent activities.

**Graph Analysis:** Essential for detecting complex interconnections within fraud networks, focusing on subgraph analysis to identify localized patterns of fraudulent activity.

**Machine Learning-Based Techniques:** Ranging from supervised to unsupervised learning, these methods are crucial for predictive analytics and identifying unknown patterns, with semi-supervised learning providing a balanced approach in label-sparse environments.

**Spectral and Dimensionality Reduction Techniques:** Simplify complex data, making it more interpretable, with techniques like PCA highlighting crucial features for anomaly detection.

**Probabilistic and Statistical Models:** Analyze statistical distributions to identify deviations indicative of fraud.

**Deep Learning Approaches:** Utilize advanced neural networks, such as CNNs and RNNs, to learn and detect complex patterns within large data sets.

Each of these categories supports a variety of sub-techniques that are tailored to address the specific challenges posed by the evolving landscape of telecom fraud. It is important to highlight that sometimes there are fraud detection methods that combine aspects of more than one technique. A method could use subgraph analysis to first isolate sections of a telecom network that are susceptible to fraud based on certain criteria, such as high call volumes or unusual connectivity patterns. Then, machine learning models like decision trees, time series, neural networks, or clustering algorithms could be applied to these subgraphs to detect anomalies or classify the node behavior. This approach leverages the strengths of both domains structural insight from graph analysis and predictive power from machine learning. A Graph Neural Network (GNN) primarily functions as a Machine Learning-Based Technique, only applicable within the subset of graph analysis methods that incorporate machine learning principles to process graph structures. However, due to its utilization of neural network architectures, GNN can also be distinctly categorized under deep learning approaches. This dual categorization is pivotal, as GNN leverages both the structural analytical capabilities of graph analysis and the advanced pattern recognition power of deep learning. This intersection

is essential for the discussions in the upcoming chapters, where the integrated application of GNN across these domains will be further explored.

Although the above techniques have demonstrated effectiveness in fraud detection, graph analysis stands out due to its superior capability for addressing the challenges in fraud detection. The primary reasons why graphs are particularly well-suited for this task are detailed below [11]:

**Inter-dependent Nature of Data:** communication, or transaction networks, often exhibit complex interdependencies. Graphs excel in representing these relationships, where entities are interconnected and influence each other.Detecting anomalies often requires understanding how entities are interlinked, as fraudulent activities or failures in one part of the network can have repercussions throughout the network.

**Powerful Representation:** Graphs provide a natural framework to represent complex datasets with multiple interdependencies. By defining nodes and edges with specific attributes, graphs capture not just the entities and their interactions but also the nuanced relationships and correlations that might extend over several degrees of separation. This capability is crucial for understanding the layered structures within data, especially in telecommunication networks, enabling more precise anomaly detection.

**Relational Nature of Problem Domains:** Many anomalies or fraudulent patterns manifest through relationships within the data. For instance, fraud can spread through networks via connections among individuals (opportunistic fraud) or can be orchestrated by closely-knit groups (organized fraud). Graphs allow for modeling such relational anomalies effectively, making it possible to detect complex fraud schemes that might not be visible through individual data analysis alone.

**Robustness Against Adversarial Actions:** Graphs offer a robust way to detect anomalies in environments where fraudsters might manipulate their behavior to avoid detection. For example, while fraudsters can alter behavioral cues like log-in locations or times, it's significantly more challenging for them to alter their position or connectivity within a network without being detected. Graphs provide a holistic view of the network, making it harder for fraudsters to conceal their activities without a deep understanding of the entire network's dynamics.

### 2.2.1 Graph Analysis Based Fraud Detection

This section outlines the various approaches and methods used in graph anomaly detection, categorized into several key domains that facilitate the identification and analysis of anomalies within graph-based data[11].

Graph anomaly detection techniques can be broadly divided based on their approach to handling data, the type of graph data they analyze, and whether the data is labeled.

### 2.2.2 Graph Classification in Anomaly Detection

In the field of anomaly detection, graphs are primarily categorized based on two main dimensions: temporality and node/edge characteristics. Initially, graphs are classified by their temporality into two fundamental types:

1. **Static Graphs**: These graphs have a fixed structure where both the structure and the attributes of the nodes and edges do not change over time. They are used to model systems or networks where relationships and entities remain constant, providing a snapshot of the system at a specific point in time.
2. **Dynamic Graphs**: Unlike static graphs, dynamic graphs capture the evolving nature of systems by allowing changes in their structure and/or attributes over time. This adaptability makes them suitable for representing networks where relationships and entities evolve, such as social networks or transactional systems.

Following the classification by temporality, both static and dynamic graphs are further divided based on the characteristics of their nodes and edges:

- **Plain Graphs**: These are the simplest form of graphs, consisting only of nodes and edges without any additional attributes. They are useful for studying the fundamental structure of a network.
- **Attributed Graphs**: In these graphs, nodes and/or edges possess attributes that provide additional information beyond mere connectivity. This additional layer of information allows for more nuanced analyses and can help identify patterns based on attributes of the entities involved.
- **Heterogeneous Graphs**: These graphs are characterized by having multiple types of nodes and/or edges. They represent complex systems where different kinds of entities and relationships coexist, such as in a multifaceted marketplace with various types of interactions and transactions.

This hierarchical classification system not only helps in structuring the approach towards analyzing graphs but also in selecting appropriate methods and algorithms for anomaly detection specific to the nature and complexity of the graph being studied as outlined below.

### 2.2.3 Detailed Techniques and Sub-Methods

Each major category encompasses various techniques designed to tackle specific aspects of anomaly detection:

- **Graph Anomaly Detection (GAD):**
  - **Static Graphs**
    - Structure-Based Methods for Plain Graphs.
    - Community-Based Methods for Attributed Graphs.
    - Motif-Based Methods for Plain Graphs.
    - Meta-path based anomaly detection.
    - Multi-modal fusion techniques to integrate different types of data and relationships.
  - **Dynamic Graphs**
    - Feature-based, Decomposition-based, and Community-based detection methods.
    - Window-based techniques to monitor changes over set intervals.
    - Temporal attentive mechanisms to detect changes in multi-typed entities and relationships.
    - Dynamic meta-path based anomaly detection, adapting to the temporal evolution of the graph structure.
- **Graph-Anomaly Detection with Deep Learning (GADL):**
  - Deep Neural Networks (DNN),Autoencoders and deep neural networks for learning data representations and detecting anomalies
  - Graph Convolutional Networks (GCN), to learn node embeddings that help identify structural and attribute-based anomalies.
  - Generative Adversarial Networks (GAN), modeling normal data distributions and identifying anomalies as deviations.

- Network Representation, using Dynamic Meta-Path Based Anomaly Detection and latent space representations.

- Reinforcement Learning, to learn strategies for identifying anomalous nodes and edges.

- Temporal Attentive Mechanisms, identifying significant temporal changes in graphs and attention mechanisms to highlight important parts of the graph for anomaly detection.

### 2.2.4 Graph Anomaly Detection (GAD)

**Anomalies in Static Plain Graphs**, techniques are employed to detect structural anomalies in graphs that do not feature node or edge labels.These methods focus primarily on *structural anomalies* that deviate from typical graph patterns. Techniques such as **Structural Anomaly Detection** identify nodes or subgraphs with unusual connectivity patterns. By examining the topological peculiarities of the graph, these methods focus on detecting structural outliers that do not conform to the common patterns of the graph. Techniques such as subgraph mining are pivotal, where the aim is to find unusual subgraphs that could indicate malicious activities or faults [11]. Another approach, **Motif Analysis**, involves searching for over- or under-representation of small subgraphs (motifs) indicating structural deviations. Primarily, these methods transform graph anomaly detection into an outlier detection problem. An additional method is the use of graph kernels, which measure the similarity of graphs and subgraphs to typical patterns, helping to pinpoint anomalies [74]. A variant technique is **Community Analysis** which is employed to detect anomalies by analyzing the community structure, identifying nodes that do not fit into any community or anomalously bridge communities.Community-based methods are particularly effective in networks where nodes are expected to exhibit similar behaviors within their communities or clusters. Anomalies are detected based on deviations from these expected behaviors, such as nodes having connections outside their typical community ties or exhibiting different properties compared to their peers [40]. Reference work in plain graphs often use methods that identifies patterns in graphs, with focus on ego-nets and induced subgraphs from its neighbours. They look for "strange nodes" in a social perspective either because they are highly or poorly connected, or reveal "strangeness" in the direction and the weight of its connections, either full inbound links (Blackholes) or full outbound links (Volcanos) or unbalanced weight distribution (heavy links) [65]. This is the case of OddBall [9] algorithm for detecting anomalies in weighted graphs based on patterns such as weight, rank, and eigenvalues in neighborhood subgraphs. The focus is on identifying violations of expected patterns which could indicate anomalies, leveraging traditional statistical methods.
GraphScope [99] employs a parameter-free approach, making it suitable for real-world ap-

plications where pre-defining parameters can be complex and impractical. The method leverages the concept of graph snapshots, capturing the state of the graph at different time intervals, and uses these snapshots to identify substantial structural changes over time. This approach enables the detection of critical events, such as the emergence of new communities or sudden changes in connectivity, providing valuable insights into the dynamic behavior of large-scale networks. The **Anomalies in Attributed Graphs**, focus on graphs with labeled nodes and edges. This includes methods such as *Attribute-Assisted Structural Anomaly Detection* integrating structural properties with node and edge attributes, like *Clustering-Based Methods* leveraging clustering algorithms considering both the attributes and graph topology, and *Inference-Based Techniques* applying probabilistic models to estimate the likelihood of anomalies based on attributes and their positions in the graph [90].

Other relevant method using plain and attributed graphs is Label propagation, which assigns information from labelled nodes to unlabelled nodes through a graph. From an initial subset of labelled nodes, this method constructs a graph based on node-to-node similarity. In the case of nodes connecting to multiple other nodes with different labels, multilevel graphs can be built based on their connected attributes. This class of algorithms aims to identify suspicious nodes based on their connectivity and synchronized behaviour. Real and complex networks tend to have community structures and label propagation methods can find communities by analysing each single entity under the assumption that nodes that are connected in the graph are likely to share the same semantic label. Thus, label propagation algorithms can also infer fraudulent from a priori legitimate nodes based on their suspicious clique membership and relationships to known fraudsters in the network [51] [106] [27] [47] .

Finally, Latent Factor Models (LFM's). Here, latent refers to a group of variables that are inferred from other variables that can be observed. Latent features are computed from observed features using matrix factorization. In a graph context, the observable interactions between a set of nodes can provide information that is useful to predict unobserved links. Latent models are frequently coupled with machine learning and Bayesian methods in order to identify the latent structure associated to the observable variables [84].

Detecting anomalies in **heterogeneous graphs** involves navigating their complex structures, which include multiple types of nodes and edges, with encode diverse relationships and interactions within the data.Traditional anomaly detection methods often fall short when applied to heterogeneous graphs because they fail to account for the rich semantic and structural diversity present in these graphs [11]. Four different detection techniques stand out in terms of effectiveness, textbfMeta-path Based Anomaly Detection, Multi-modal fusion techniques, Temporal Attentive Mechanisms and Dynamic Meta-path Based Anomaly Detection.

**Meta-path Based Anomaly Detection** utilizes predefined paths in the heterogeneous graph, which connect different types of nodes through a sequence of relations. These paths, known as meta-paths, are used to extract and learn the complex and hidden relationships between various entities in the graph. By analyzing these paths, the model can detect anomalies that

deviate from the normal patterns expected in these meta-paths. This approach is particularly effective in graphs where relationships between nodes can be more meaningfully interpreted through their types and connections, relevance (HeteSim) and similarity (PathSim) [93] [100]. On the other hand, **Multi-modal fusion techniques** [80] involve integrating different types of data and relationships within a heterogeneous graph. This method leverages various data modalities (such as node attributes, edge types, and external semantic information) to enrich the graph model. By fusing these modalities, the technique enhances the graph's representational learning, providing a more comprehensive view of the entities and their interactions. This enriched representation helps in better identifying discrepancies and anomalies in the graph by capturing a broader context. A key example in this field is StreamSpot [70] which introduces a significant advancement in the real-time anomaly detection within heterogeneous graph streams, crucial in cybersecurity for detecting advanced persistent threats at the host level, achieving in tests over 95% accuracy. Unlike traditional methods, StreamSpot efficiently processes the complexities of diverse node and edge types as they stream in, utilizing a novel similarity function that evaluates graphs through the frequency of localized substructures. This approach allows for the transformation of intricate graph data into concise vector representations, optimizing both computational speed and memory use.
While it's conceivable to use basic forms of multi-modal fusion without relying on machine learning or deep learning, the fact is that due to the complexity and variety of data types they integrate, is not common or generally effective to implement multi-modal fusion without advance machine learning or deep learning techniques.The same applies to Temporal Attentive Mechanisms and Dynamic Meta-path Based Anomaly. For this reason, these techniques will be addressed in 2.2.5.

### 2.2.5 Graph-Anomaly Detection with Deep learning (GADL)

Graph anomaly detection has become a critical research area due to the increasing complexity and size of graph-structured data in various domains. Various studies, and survey's [62], have examined the efficacy of deep learning methodologies in detecting anomalies within graphs. These studies highlight the remarkable effectiveness of deep learning techniques in identifying anomalies, showcasing the significant impact and potential of Graph-Anomaly Detection with Deep Learning (GADL) in advancing this field. In fact, Deep learning techniques have significantly advanced the field of Graph anomaly detection by providing powerful tools to model and analyze the complex structures and dynamics of graphs. These methods leverage different aspects of neural network architectures, such as deep neural networks, graph convolutional networks, generative adversarial networks, network representation learning, reinforcement learning, and attention mechanisms, each offering unique advantages in uncovering anomalous patterns in complex and large-scale graph data.
**Deep Neural Networks (DNN)** based techniques are foundational in this context. These

methods, such as autoencoders and deep neural networks, are used to learn representations of the data. An exemplary approach is the DONE model [59], which utilizes separate autoencoders for the structure and attributes of the graph to detect global, structural, and community anomalies. This model measures anomaly scores based on the attribute similarity across communities and the structural connections, identifying deviations from the norm [63].

In a different category OCAN (One-Class Anomaly Detection on Attributed Networks via One-Class Cross-Entropy)[126], detects anomalies in attributed networks using a generative adversarial network (GAN) approach. The GAN is trained to generate normal data distributions, and deviations from this learned distribution are flagged as anomalies. More specifically OCAN integrates a deep neural network architecture that combines both the structural and attribute information of the nodes. The model uses a generator to create data samples that mimic normal instances and a discriminator to distinguish between real normal instances and the generated samples. The training objective is to make the generator produce data that the discriminator cannot differentiate from real data, effectively learning the distribution of normal instances. Another prominent category is **Graph Convolutional Networks (GCN)** which leverage both the graph structure and node features to learn embeddings that are crucial in anomaly detection. A reference technique in this category is AddGraph model, designed to handle dynamic graphs where the structure and attributes of the graph evolve over time. AddGraph integrates an attention-based temporal GCN that dynamically adjusts the attention weights to highlight significant temporal changes and relationships in the graph. This approach enables the model to detect anomalies that arise due to sudden changes in node connections or attribute values. Additionally, the model incorporates a temporal graph embedding technique to maintain a comprehensive history of node states, which helps in identifying long-term patterns and anomalies. [115].

In a heterogeneous dynamic perspective, **temporal attention mechanisms** focus on detecting changes over time in multi-typed entities and relationships within the graph. This approach uses attention mechanisms to weigh the importance of different nodes and edges across various time steps dynamically. By focusing on how relationships evolve or how entities behave over time, temporal attentive mechanisms can pinpoint anomalies that represent sudden changes or gradual deviations from typical interaction patterns. Furthermore, **Dynamic meta-path based anomaly detection** adapts to the temporal evolution of the graph structure. Unlike static meta-path techniques that use fixed paths, dynamic meta-path methods adjust these paths as the graph evolves over time. This adaptability allows the detection models to stay relevant and effective even as the underlying data changes. Dynamic meta-path models can track and analyze the creation, modification, and dissolution of nodes and edges, identifying anomalies in evolving patterns of interaction. This is the case of HRGCN, (Heterogeneous Graph-level Anomaly Detection with Hierarchical Relation-augmented Graph Neural Networks) designed to detect anomalies in heterogeneous graph data [60]. This model

leverages hierarchical relation-augmented **Graph Neural Networks (GNNs)** to effectively capture the relationships between intra-type and inter-type relationships, allowing the model to learn the complex dependencies between different types of nodes and edges. Additionally it employs a self-supervised learning strategy, training without the need for labeled anomaly data. This is achieved by generating pseudo-anomalies and using them to train the model, making it particularly useful in scenarios where labeled anomaly data is scarce, nonexistent or cannot be trusted. The model robustness and general applicability is improved by employing a heterogeneous graph data augmentation method (HetGDA), specifically by alter the graph structure, such as adding or removing nodes and edges, or swapping them, and changing the attributes of nodes and edges, which helps the model learn to detect anomalies that manifest through unusual attribute values or patterns.

There are multiple variations of attention mechanism-based techniques, the dynamics do not apply only to the structure, they can apply to attributes. The ResGCN model (Attention-based Deep Residual Modeling for Anomaly Detection on Attributed Networks) [82]. This model uses a residual modeling through a deep neural network to capture nonlinearity, allowing the model to highlight significant deviations that indicate anomalies (discrepancies between the true network data (both structure and attributes) and the reconstructed data predicted by the model). This residual information informs an attention mechanism within Graph Convolutional Networks (GCNs), dynamically adjusting the importance of nodes and edges during the learning process. This helps in focusing on the most critical parts of the graph. For representation learning, ResGCN aggregates information from neighboring nodes, weighted by the attention mechanism to form effective node embeddings. These embeddings are then used for network reconstruction, predicting both the network structure and node attributes. The model calculates reconstruction errors for structure and attributes, using high residual errors to identify anomalous nodes. This combined approach ensures that ResGCN can dynamically adjust to and effectively highlight anomalies in complex, attributed networks.

The models based on **Dynamic Meta-Path Based Anomaly Detection** can be categorized under network representation-based techniques. This method involves encoding the heterogeneous relationships within dynamic graphs into a latent representation that captures the essential features of meta-paths over time. By representing these meta-paths dynamically, it can effectively detect anomalies based on evolving interaction patterns within the graph [62]. Methods like NetWalk [121] utilize deep representation learning to update node embeddings in dynamic graphs, and clustering algorithms to identify outliers. This technique is particularly useful for handling the temporal dynamics of graphs and ensuring that the learned embeddings are up-to-date with the latest graph structure. A notable example is the DeepWalk model [83], which uses random walks to generate sequences of nodes. These sequences are then fed into a Skip-gram model, a neural network model typically used for natural language processing, to learn latent representations of vertices in the graph. This method captures intricate relationships and structures within the graph, making it useful

for tasks like classification and clustering. Table 2.2 compares the Graph Anomaly Detection Techniques across Various Models.

**Table 2.2:** Comparison of Graph Anomaly Detection Techniques across Various Models.

| Feature | DONE | ResGCN | NAC | NetW | GScope | DWalk | HRGCN | AddG | OCAN | SSpot | HetS | PathS | ESpokes | CSync | GConst | GAssoc | Oddball | V&Bholes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Det. Global Anomalies | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Det. Structural Anomalies | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Det. Community Anomalies | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Uses Autoencoders | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Uses Dynamic Embedding | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Uses Graph Convolutional Network | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Uses Attention Mechanisms | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Uses Residual Modeling | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Uses Reinforcement Learning | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| GRAPE Integration | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| MaxG Integration | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| GraphL Integration | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Vineyard Integration | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Handles Structural Data | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Handles Attribute Data | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Handles Dynamic Data | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Unsupervised Learning | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Reinforcement Learning | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Supervised Learning | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Precision Evaluation | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Recall Evaluation | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| F1-score Evaluation | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Reconstruction Errors Evaluation | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Residual Errors Evaluation | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

**Note:** This table provides a comprehensive comparison of various graph anomaly detection techniques across multiple models. The features compared include the ability to detect different types of anomalies (Global, Structural, Community), the use of various techniques (Autoencoders, Dynamic Embedding, Graph Convolutional Networks, Attention Mechanisms, Residual Modeling, Reinforcement Learning), integration capabilities with different graph processing systems (GRAPE, MaxGraph, Graph-Learn, Vineyard), and the handling of different types of data (Structural, Attribute, Dynamic). Evaluation metrics such as precision, recall, F1-score, reconstruction errors, and residual errors are also included.

## 2.3 Proposed Approach and Methods for Anomaly Detection

The evolution of fraud tactics, now more complex and networked than ever, poses a challenge for traditional fraud detection methods. To address this, fraud detection algorithms should focus on topological and multidimensional subgraph analysis, which allows them to capture complex structures and connectivity patterns within the network. This approach provides a more comprehensive understanding of the network's topology, making it possible to detect sophisticated fraud patterns that might be missed by traditional node-level or edge-level analyses.

Many algorithms are designed to be generic, but it is necessary to adapt them to specific use cases to achieve optimal effectiveness. In telecom networks, for instance, specialized capabilities are required to detect fraudulent nodes and rings. This targeted focus, especially on identifying fraudulent rings, addresses a critical need in telecom fraud prevention. These rings often represent organized and sophisticated fraudulent activities that generic anomaly detection methods may overlook. By understanding the roles of nodes within the network and their relationships through control and influence, it becomes possible to pinpoint nodes that are central to fraudulent schemes.

Additionally, scalability and efficiency are crucial to handling large telecom datasets without significant performance degradation. A parameter-free design simplifies deployment and minimizes the risk of configuration errors, enhancing the system's robustness and adaptability to various network conditions. By leveraging influence and control metrics, algorithms can rank nodes based on their importance within the network. This approach helps in providing actionable insights into key nodes involved in fraudulent activities, particularly those with the highest capability to commit fraud. This role-based understanding ensures that the most critical nodes are effectively identified and monitored.
These challenges and requirements are specifically addressed by the STARBRIDGE model, which will be presented in detail in Chapter 4. STARBRIDGE leverages topological and multidimensional subgraph analysis, tailored to detect complex fraud patterns in telecom networks. It emphasizes the identification of influential and controlling nodes, ensuring robust and efficient detection of sophisticated fraudulent activities.
One significant gap in fraud detection landscape is the inadequacy of existing models to handle dynamic network data effectively. Many traditional approaches are static, failing to account for the time-evolving nature of real-world networks where relationships and behaviors change continuously. While some dynamic algorithms exist, such as NetWalk [121] they often lack the ability to capture the full temporal complexity of evolving networks. These algorithms may focus on specific types of temporal changes or rely heavily on periodic snapshots, which can miss subtle yet critical shifts in behavior. NetWalk, for example, employs random walk-based techniques on dynamic graphs to identify anomalies, but it typically

relies on discrete time steps or intervals. This method can fail to detect anomalies that occur in between these intervals or those that evolve gradually over time. Similarly, other Dynamic Graph methods may update its models periodically based on batched data, which introduces delays in anomaly detection and can overlook incremental changes that are significant over a continuous timeline. Moreover, these algorithms might prioritize specific types of temporal changes, such as sudden spikes or drops in activity, while missing more subtle, long-term trends and patterns. The reliance on periodic snapshots means that there is often a trade-off between the granularity of temporal data and the computational complexity involved in processing it. As a result, critical shifts in behavior that occur at a finer temporal resolution might go unnoticed, leaving gaps in fraud detection capabilities.

One significant gap in the current landscape of fraud detection is the need for advanced feature selection that integrates both static and temporal data. Traditional models often focus on either static snapshots or simple time-based changes, failing to capture the complex dynamic patterns that evolve over time. A comprehensive approach that includes both types of features is essential to detect intricate fraud schemes that might not be evident when considering static data alone. Combining these features, models can significantly enhance its overall detection capabilities, providing a more accurate and detailed analysis of network behaviors and relationships.

Another critical gap is the lack of robust visualization tools within many existing fraud detection systems. Effective visualization is essential for analysts to delve deeply into the data, understand the underlying patterns, and make informed decisions. While some current models offer basic visualization capabilities, they often fall short in terms of interactivity and detail. Traditional static charts and graphs do not provide the level of insight needed to fully understand complex fraud schemes, especially in large and intricate networks.

The absence of interactive and intuitive visualization tools hampers the ability of analysts to interpret complex data outputs and act upon them efficiently. Advanced visualization techniques, such as those incorporating interactive dashboards, real-time data exploration, and multidimensional views, can significantly enhance the usability and effectiveness of fraud detection systems. These tools should enable analysts to zoom in on specific areas of interest, filter data dynamically, and visualize relationships and anomalies in ways that reveal hidden patterns and connections.

Incorporating advanced visualization techniques would allow for a more intuitive understanding of network behaviors and anomalies. For instance, heat maps could illustrate the intensity of suspicious activities across different network segments, while temporal graphs could show how fraud patterns evolve over time. Such tools not only aid in the detection process but also enhance the ability to communicate findings to non-technical stakeholders, thereby facilitating quicker and more effective decision-making.

By addressing these gaps—dynamic data handling and advanced visualization—future fraud detection models can significantly improve their effectiveness and usability. Models that can adapt to evolving network conditions and provide rich, interactive visualizations will be better equipped to identify and respond to complex fraud schemes. These enhancements will enable a more proactive and comprehensive approach to fraud prevention, ultimately safeguarding networks more effectively against sophisticated fraudulent activities. To address these challenges, chapter 5 details TgraphSpot which enhances the detection of dynamic network data by integrating both static and temporal features, allowing it to capture complex evolving patterns and relationships that static models might miss. This continuous, real-time analysis ensures timely identification of anomalies and supports scalability for large datasets, crucial for industries like telecommunications. On the other hand, CALLMINE provides advanced visualization tools, enabling analysts to perform deep-dive analyses and interpret complex data through interactive dashboards and summary visualizations. It supports both supervised and unsupervised learning, making it adaptable to various detection scenarios. Together, these models fill critical gaps by offering comprehensive feature analysis, real-time detection, and robust visualization tools, significantly improving the effectiveness and usability of fraud detection systems.

## 2.4 Federated Learning and Multi-access Edge Computing (MEC)

### 2.4.1 Federated ML

The development of fifth-generation (5G) broadband cellular networks has the potential to revolutionize various industries by enabling almost instantaneous data communication and high transmission rates, low communication latency, and massive connectivity. This facilitates applications such as autonomous vehicles, smart cities, smart homes, and Industry 4.0 [49]. To meet the demands of these new applications, the Edge Computing paradigm has emerged, which extends cloud capabilities to the edge of the network, allowing computationally-intensive tasks and data storage to occur closer to the end-user equipment and within the Radio Access Network (RAN), thus enhancing the quality of service [4],[7].

Several technologies have emerged to support Edge Computing and to redefine the Cloud Computing paradigm, shifting away from centralized architectures to alleviate the constraints faced by centralized cloud systems [7]. Multi-access Edge Computing (MEC) is particularly relevant in the telecommunications sector, offering benefits such as low latency and high bandwidth for Mobile Network Operators (MNO), Application Service Providers (ASP), and end-users. MEC allows MNOs to provide RAN access to third-party vendors for deploying applications and services, ASPs to benefit from scalable infrastructure, and end-users to

experience improved performance through offloading techniques [32].

In recent years, studies have explored ML in Edge Computing and MEC scenarios for optimizing architecture planning (e.g., resource allocation, offloading management) [45],[88],[48], addressing security issues [29],[77],[78], and developing operational intelligence [39].These studies typically employ centralized approaches where ML models are trained at a single location using data aggregated from edge devices. More recently, Federated Learning concepts have been applied to train models without sharing data between edge devices. Works like [109],[118],[122] utilize Federated Learning to develop ML applications from decentralized data. Other studies propose new Federated Learning frameworks using Reinforcement Learning [87],[109],[120] or innovative training and aggregation techniques [33],[118].

Based on the review of current advancements and performance gaps, several research directions are identified:

## Identified Performance and Research Gaps

- **Integration with FML**: Integrating MEC with FML frameworks remains complex. Effective coordination between edge nodes and central servers is necessary to ensure seamless model updates and data synchronization [120].
- **Resource Management**: Efficiently managing computational resources and energy consumption at the edge is a critical challenge. [88] highlighted the need for advanced resource allocation strategies to optimize MEC performance.
- **Security and Privacy**: Ensuring the security and privacy of data processed at the edge is paramount. Techniques to protect data integrity and prevent unauthorized access need further development [77],[78].
- **Enhanced Hardware Utilization**: Research should focus on optimizing FML algorithms to work efficiently on resource-constrained edge devices. This includes developing lightweight models and efficient training techniques.
- **Robustness to Adversarial Attacks**: Future studies should investigate robust FML frameworks that can handle adversarial attacks and non-IID data distributions, ensuring consistent performance across diverse edge environments.
- **Scalable Aggregation Techniques**: Developing scalable model aggregation techniques that can handle a large number of edge devices without compromising performance is essential for large-scale telecom networks.
- **Seamless Integration of FML and MEC**: Research should explore seamless integration strategies for FML and MEC, ensuring effective coordination and synchronization

between edge nodes and central servers.

- **Advanced Resource Management**: Innovative resource management strategies are needed to optimize the use of computational resources and energy consumption in MEC environments.
- **Enhanced Security Protocols**: Developing advanced security protocols to ensure data integrity and privacy at the edge is crucial for the adoption of MEC in sensitive applications like telecom fraud detection.

Federated Machine Learning and Multi-access Edge Computing hold significant promise for enhancing telecom fraud detection. However, most existing studies do not account for the hardware constraints of edge devices during the training or inference phases of ML models. Few works explicitly mention the resource-constrained nature of applications utilizing Federated ML [109]. Similarly, few studies address the MEC architecture specifically, focusing instead on other Edge Computing paradigms [120]. To the best of our knowledge, there is no research addressing Federated Learning within the telecommunications domain and MEC architecture.
Despite significant advancements, detecting telecom fraud remains a persistent challenge, further complicated by perpetrators who engage in coordinated fraudulent activities while seamlessly blending with legitimate users. Traditional methods, which focus on identifying anomalies in call patterns, struggle to detect the increasingly sophisticated and collaborative fraud schemes that are emerging, However, the shift toward data-rich, graph-based representations of telecom interactions opens new possibilities for GNN-powered fraud detection strategies.

This state-of-the-art review explores the evolving landscape of telecom fraud detection, emphasizing the urgent need for innovative and adaptable solutions capable of unraveling the intricate web of factors that enable fraud in the telecom sector.

# Chapter 3

# Malware - Multistage Botnet Detection and Tactics

## 3.1 Introduction

Mobile Malware refers to malicious code designed specifically to compromise mobile devices with the goal of enabling various types of fraud, either by compromising sensitive information or by causing disruption to organizations and businesses.
Statistics point out that 3.5 million malicious installation packages were detected in 2021, which lead to 46.2 million attacks worldwide [52]. While there is an overall declining trend in the number of attacks, mobile malware continues to evolve, including new capabilities. Attacks are more complex and harder to spot and malware can often be downloaded from official application stores with a much higher success rate in tricking end users. Unsurprisingly,the Fraud Loss Survey reports that fraud losses increased 28%, or approximately $11.6 Billion USD in 2021 when compared to 2019 [24]. The Short Message System (SMS), despite being deprecated, is one of the main spread vectors used in mobile malware. Indeed, the SMS Phishing/Pharming is the 3rd top fraud method in 2021, being responsible for financial impacts of around $2.03 Billion USD [24].

These attacks can cause substantial financial loss, identity theft, and privacy violations for end users. They also negatively affect the services Mobile Network Operators (MNOs) provide. Since many malware campaigns spread through SMS, customers often hold MNOs accountable and call for stronger protection. [38].

FluBot is one of the most recent threats in terms of malware that uses the Domain Name System (DNS) to interact with Command and Control (C2) servers having an impact on end devices and network operators. Through FluBot, the malware deployed in mobile devices

can be instrumented via the C2 channel to act as a botnet enabling coordinated attacks like Distributed Denial of Service (DDoS). Other types of botnets exist like Mirai, Reaper and variants that appear from time to time [111].

Several botnet detection techniques have been proposed, including honeypot-based analysis, communication signatures (e.g., whitelists and blacklists), deep learning approaches based on neural networks or reinforcement learning, statistical analysis, distributed methods, as well as hybrid strategies that integrate different detection mechanisms [114]. In addition, recent defence mechanisms such as Moving Target Defense (MTD) can be leveraged to improve resilience against botnet attacks [13, 108].

Most of these approaches rely on deploying monitoring agents on mobile devices and/or within the network infrastructure, for example honeypot agents. Their detection accuracy is typically influenced by how effectively heterogeneous signals and methods are combined [110].

Botnet detection may also leverage information beyond DNS traffic, such as the number of packets per second (PPS) or the average payload size. Based on these features, clustering techniques, including k-Nearest Neighbour (kNN), can be applied to identify anomalies in communication flows (e.g., excessive request patterns).

Network operators cannot depend on installing software on mobile phones or IoT devices to secure their networks. This approach is ineffective for roaming users, who fall outside the operator's administrative control, and operators have no authority to install software on devices they do not own. In addition, many IoT devices do not even support the installation of third-party software in the first place. Furthermore, detection and mitigation mechanisms must operate in a coordinated manner so that appropriate tactics can be selected based on the current security risk (for example, the number of infected devices).

A central question arises at this stage: how can mobile network operators strengthen their security posture while overcoming the limitations of existing techniques? To address this challenge, we introduce MONDEO-Tactics5G, a multistage botnet-detection and mitigation framework designed for 5G and future networks. The system is capable of identify FluBot-type malware, known for its significant impact, by combining efficient detection methods with operator-defined tactics that can be incorporated directly into existing network controls. MONDEO-Tactics5G is guided by three fundamental design principles. First, it avoids any requirement for deploying software agents on user devices; no code needs to run on subscribers' smartphones or IoT equipment. Second, it is engineered to integrate seamlessly with core operator services, such as DNS infrastructure. Third, it enables the use of customizable policies and mitigation tactics that can be aligned with the operator's security mechanisms, including Access Control Lists (ACLs).

For the botnet-detection module of MONDEO-Tactics5G, designed to identify FluBot in-

fections, we adopt a four-phase pipeline: (1) whitelist and blacklist filtering, (2) analysis of DNS query rates, (3) detection of Domain Generation Algorithm (DGA) behaviour, and (4) a machine-learning–based evaluation stage. This sequence enables the system to pinpoint compromised devices and constrain their activity within the network. Once a botnet presence is detected, it becomes necessary to determine the corresponding command-and-control (C2) servers. Among the large volume of DNS traffic generated by FluBot, the few queries that return valid IP resolutions are the key indicators, as they may reveal the addresses of the C2 infrastructure. Following this stage, FluBot-infected devices typically proceed to initiate HTTP communication with the C2 server, a pattern that further confirms the infection.

The tactics available in MONDEO-Tactics5G are varied, ranging from isolating or quarantining compromised devices and servers to deploying CAPTCHA mechanisms that help differentiate human users from automated bots. The specific tactic selected must take into account the system's operational context,for instance, the workload on customer-service teams who may need to support affected subscribers, as well as the overall utility considerations and business objectives defined by the operator.

The contributions of MONDEO Tactics5G lie in establishing a solid basis for botnet detection and for the use of mitigation tactics. First, the system provides efficient botnet detection capabilities that can be integrated with the User Plane Function in 5G networks or with DNS servers. Second, it enables the identification of C2 servers, even when these are hidden within large volumes of DNS traffic. Third, MONDEO Tactics5G offers several mitigation actions, including quarantining infected devices, blocking the IP addresses associated with C2 servers, and applying CAPTCHA mechanisms at the connection level. These tactics are selected according to a utility function that takes into account both operator and customer interests, and they can be deployed in 5G environments through the Policy Control Function (PCF).

The remainder of this chapter is organized as follows. Section 3.2 introduces the background and motivation for this work, Section 3.3 details the design of MONDEO-Tactics5G. Section 3.4 delineates the evaluation methodology, while Section 3.5 provides evaluation results.

## 3.2 Mobile Malware in Telecom Fraud

Unlike previous generations, 5G rethinks and re-architects how networks are designed and managed by introducing new use cases and business models that impact not only consumers but also enterprises and industrial sectors. In 5G, most subscribers will no longer be traditional consumers; instead, a significant portion of connections will involve IoT devices with requirements that differ substantially from those of human users (e.g., smart meters, environmental sensors). Even within the IoT domain, device behaviour can vary widely depending on the use case, for example when comparing a smart meter with a connected vehicle. 5G can be broadly

categorised into three main use cases, two of which are closely related to IoT: Ultra-Reliable Low-Latency Communications (URLLC) and massive Machine-Type Communications (mMTC). The third, enhanced Mobile Broadband (eMBB), is primarily associated with general-purpose services and remains largely oriented towards human subscribers. This combination of heterogeneous devices and diverse use cases expands the definition of suspicious behaviour and gives rise to new fraud models. Banking and gaming trojans are illustrative examples of how malware already targets specific business domains, and how future trojans may evolve to exploit particular 5G use cases.

### 3.2.1 5G Impact on malicious activities

5G technology is expected to significantly impact fraud, as already highlighted in reports involving the participation of Mobile Network Operators [24]. This impact is driven, in part, by the use of Artificial Intelligence to perpetrate fraud and evade detection, a phenomenon commonly referred to as Smart Fraud. In addition, the massive increase in the number of connected devices further amplifies the effectiveness of Distributed Denial of Service (DDoS) attacks. The New Radio (NR) enhancements introduced with 5G enable support for high-density connectivity, allowing a substantially larger number of connections per unit area and higher data rates when compared to previous generations such as 4G [104].

An overall rise in fraud incidents is not only enabled by the capabilities introduced with 5G networks but is also driven by several additional factors. These include the growing use of cross-industry social engineering schemes, the expansion of financial services impersonation attacks that rely on stolen credentials from data breaches, and the increasing availability of faceless transaction portals that present themselves as legitimate websites and facilitate Subscription Fraud and Account Takeovers.

The common factor in all of the above is compromised credentials and sensitive personal data obtained through data breaches and mobile malware leading to identity theft. While the financial impact is extremely high, there are other major impacts to the MNOs, such as: customer complaints, low customer loyalty and trust, churn, dispute costs, artificial increase of traffic (SMS/Voice/Data), impact on network performance and interconnect costs, among others.

Another concerning aspect is the wide range of additional fraud scenarios that can be enabled by infected mobile devices, many of which have not yet been fully exploited by fraudsters, together with the sheer volume of scam messages capable of compromising user devices. Within the telecommunications domain in particular, most malware already obtains sufficient privileges on infected devices to place voice calls or send SMS messages. This capability enables the execution of fraud schemes such as International Revenue Share Fraud (IRSF) and A2P bypass, as well as other fraud types that are monetised through voice or messag-

ing services. Furthermore, recent figures are alarming: during the last 12 months of 2021, approximately three-quarters of consumers reported receiving scam emails, 66

### 3.2.2 Infection and Spreading

SMS is frequently used as the primary attack vector in smishing campaigns. Attackers often send text messages that mimic legitimate government programmes, such as COVID-19 vaccination notifications. Other common examples include fake gift offers, missed delivery alerts, fraudulent banking warnings, invoice or order confirmations, and supposed customer support messages. These texts typically contain a call to action and a malicious link that redirects the victim to a convincing imitation of the organisation being impersonated. Once on that webpage, the victim may be prompted to enter personal information such as credit-card details, login credentials, or date of birth, or to download a malicious application disguised as a legitimate one. Installing this software enables several forms of attack. After compromise, the device becomes part of a botnet and begins executing commands issued by the C2 servers. Attackers initially obtain target phone numbers from leaked datasets, such as previously exposed Facebook information, and then harvest additional contacts from infected devices, allowing the botnet operators to scale their attacks further.

While blocking access to known fraudulent websites seems like a common-sense approach to protect subscribers, it is in many ways ineffective. Not only it is easy to avoid blocklists, but also cybercriminals simply need their malicious URL running for a few hours (i.e., lower than 13 hours) for an effective bulk campaign [30]. Attacks happen quickly and use cheap website domains that were registered the same day or within days of the attack itself, not allowing enough time to verify if a site is safe or dangerous, since nearly a million new domains are registered every day. In practice, all security vendors take 2-3 days average to investigate and block new dangerous domains.

We anticipate in 5G an increase in smishing campaigns targeting specific businesses (e.g., financial). These campaigns will be similar to what Google defines as boutique campaigns, designed just for a few individuals in a company, where the malicious URLs last up to 7 minutes [30], making it more difficult to detect and mitigate.

### 3.2.3 Botnets and DNS

A botnet is a network of devices that have been hijacked through some form of remote code installation and are subsequently controlled by an attacker. Once compromised, these devices execute commands issued by the botmaster, who communicates with them through a command-and-control infrastructure. This infrastructure may use centralized or distributed C2 servers, peer-to-peer overlays, or other communication channels such as IRC or HTTPS that enable instructions to be delivered to the bots. To avoid attribution, botmasters usually

conceal their identity by routing their traffic through proxies or the TOR network, making it difficult for investigators or law-enforcement agencies to trace their true IP address. Botnets can grow to enormous scales, in some cases involving tens of millions of infected devices, as observed with Zeus, Storm, and Mariposa. Different botnet families are designed for distinct criminal activities, including click fraud, banking credential theft, distributed denial-of-service attacks, and cryptocurrency mining.

While control of the infected devices can be done using different protocols, 85% of malware uses the DNS protocol for malware delivery, Command and Control (C2), or data exfiltration [73]. The ubiquitous nature of DNS, high traffic volume and often overlooked attack surfaces, are the primary reasons for malware to use DNS to hide malicious activity. Three different attacks stand out:

- **Malware Using DNS for C2**. Once a device is infected, the system sends a DNS request back to the attacker's control server. In this way, the infected device becomes part of the botnet. Depending on the malware installed on the device, it will receive and execute commands associated with fraud or cyberattacks.
- **Malware Using Domain Generation Algorithms (DGAs)**. DGAs randomly generate a high number of distinct domain names, which do not need to be registered. In these cases, attackers may use just one bypassing traditional security controls like block lists or web reputation filtering.
- **DNS Tunneling** attackers encode their payloads (e.g., data theft or C2) in small chunks within DNS requests to bypass security controls. Once a device is compromised, it sends a request inside the DNS traffic to a DNS server (controlled by the cybercriminal), which is instructed to connect to the cybercriminal server, opening a channel through which data is transmitted.

### 3.2.4 Fraud Realization

Malware on infected devices acquires permissions which allow them to have full control of most system features, and be able to perform almost any task on behalf of cybercriminals. Figure 3.1 illustrates the case of FluBot, a Banking Trojan, and the associated permissions once installed, as well as the possible fraud scenarios that can be performed with the associated permissions, as summarised in Table 3.2.

From the above it is possible to observe multiple fraud scenarios that can be executed by the fraudster, once the mobile malware is installed and the infected device is part of a botnet. To notice that the permissions (*) associated with identifying the apps on the infected device, allows not only the frausters to detect crypto wallets and banking apps, but also to remove any apps that detect and prevent them from running, like antivirus software.


**Table 3.1:** FluBot Permissions on Android Phones.

| Fraud Type | Permission |
|---|---|
| Account Take Over (OTP) | RECEIVE_SMS |
| SMS Spamming | SEND_SMS |
| IRSF | CALL_PHONE |
| Artificial Voice Traffic | CALL_PHONE |
| Steal Banking Credentials | QUERY_ALL_PACKAGES* |
| A2P Bypass | SEND_SMS |
| Steal Crypto Wallets | QUERY_ALL_PACKAGES* |
| Click Fraud | INTERNET |

**Table 3.2:** Permissions and Potential Fraud Types.

Additionally permissions like WAKE LOCK, FOREGROUND SERVICE, MODIFY AUDIO SETTINGS and IGNORE BATTERY OPTIMIZATIONS, allow fraudsters to run totally unnoticed to the end user. Figure 3.1 represents the multiple phases of FluBot Malware, from infection to fraud execution, a process in many ways similar to most mobile malwares.

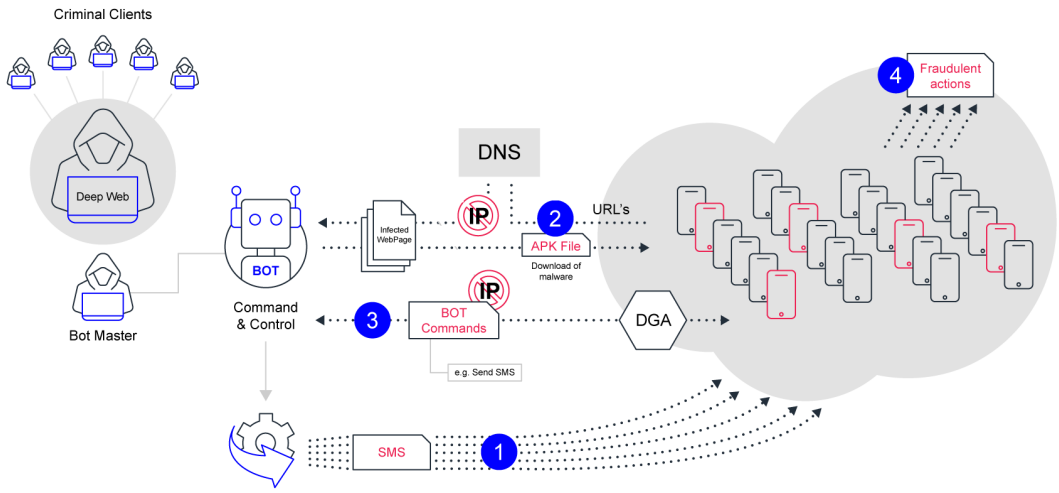



**Figure 3.1:**
(1) Smishing Campaing (2) FluBot Infection (3) BotNet Commands (4) Execution of Fraudulent actions.

Mobile devices are compromised via a smishing campaign ***(1)***, which in this case targets users of specific banking applications in certain countries. The SMS message prompts the user to install an application ***(2)*** impersonating a package delivery provider (e.g., FedEx, DHL), supposedly to track or reschedule a delivery. Once installed, the device becomes infected with the FluBot malware and communicates with its command-and-control (C2) server using DNS tunnelling over HTTPS (DoH). FluBot ***(3)*** employs a domain generation algorithm (DGA) to maintain communication with its C2 infrastructure. In more recent versions, additional seeds are downloaded from the C2 server to generate further domains. The malware is commonly propagated via SMS messages sent to the contacts stored on an infected device ***(4)***, and subsequently carries out fraudulent activities on behalf of the fraudster.

In 5G, we anticipate new and enhanced versions of current malwares, able to explore specific scenarios associated with the business and type of devices in different network slices, as exemplified in Figure 3.2, introducing a considerable number of new fraud types currently unmonitored by Mobile Network Operators. It is critical to identify potential devices that can be the source of fraud, and minimize their impact.

### 3.2.5 Malware detection requirements

The requirements for an efficient detection of malware in 5G and beyond networks include:

1. Perform flexible data collection in existing systems of mobile network operators;

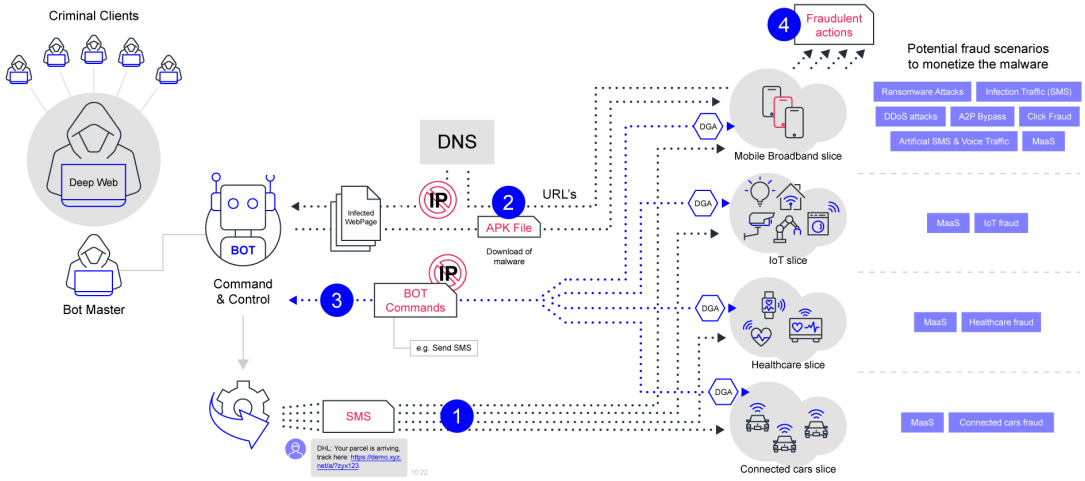



**Figure 3.2:** Mobile Malware affecting different network slices in a 5G network.

2. Avoid the installation of agents, software in end user equipment;

3. Enable distinct tactics according to the probability of a security event or severity.

4. Support multiple decision layers considering performance issues. For instance, support blacklist(s) for immediate denial.

## 3.3 Multistage Botnet Detection and Tactics for 5G/6G networks

This section describes MONDEO-Tactics5G - a Multistage Botnet Detection and Tactics approach that can be implemented in 5G and beyond networks.

### 3.3.1 Overall Architecture

The architecture depicted in Figure 3.3 illustrates the different components of the multistage botnet detection and tactics.

The detection module, referred to as MONDEO, communicates with the Domain Name System through the *M.1* interface and with the User Plane Function (UPF) via the *M.2* interface. Through *M.1*, the DNS server supplies MONDEO with the necessary information about DNS queries and their corresponding replies. The link to the UPF is essential for identifying traffic flows that may be directed toward a C2 server. The *M.R* interface enables MONDEO to transfer information to TACTICS whenever malware is detected, allowing the appropriate mitigation actions to be triggered. This may include details about the originating device (the User Equipment) and the characteristics of the C2 server that has been identified.

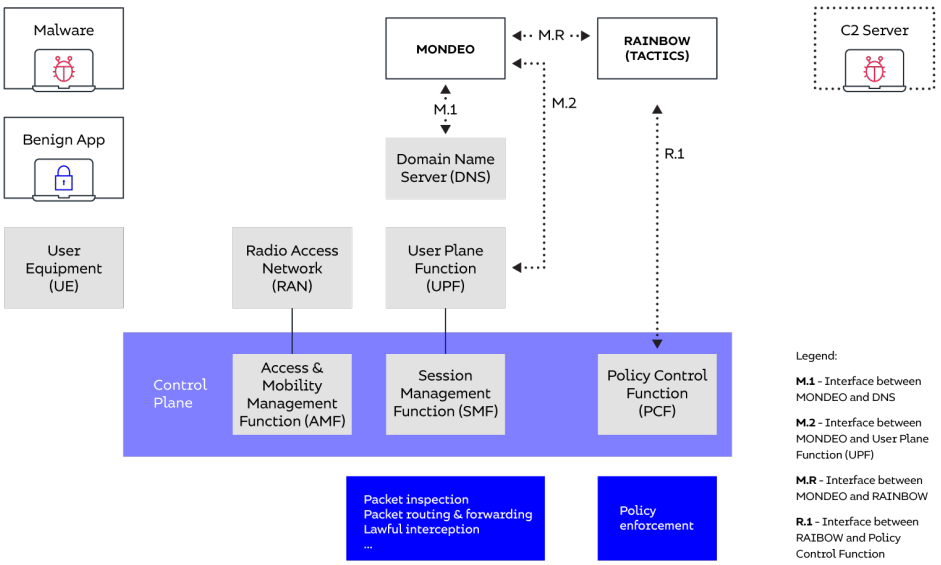



**Figure 3.3:** MONDEO Architecture.

The detection software runs in a microservice-based architecture, providing the detection service through a RESTful API. Through the API it is possible to exchange the required data, where each request is treated individually and processed efficiently through parallelization by leveraging the support of multiple threads in Python Flask.

### 3.3.2 Botnet Detection

This section provides details regarding the implementation of MONDEO for botnet detection.

The general view of MONDEO covers the detection workflow and the practical decisions taken to build a functional and deployable proof of concept.

The botnet detection process in the multistage MONDEO pipeline is organised into four phases (Fig. 3.4): 1- Whitelisting and Blacklisting, 2- Query Rate Analysis, 3- DGA Evaluation, and 4- Machine Learning Evaluation. Each phase produces an outcome of either Benign (passed) or Infected (flagged), and the next phase is only triggered when no classification is made. In phase 2, the Query Rate Analysis marks as Infected those DNS requests that exhibit abnormal ratios or behaviour. The results from the DGA and ML phases enable a feedback mechanism, which can be used to update whitelist and blacklist entries according to the classification outcome, adding benign entries to the whitelist and infected ones to the blacklist.

The design of MONDEO takes into account concerns related to deployment, accuracy and efficiency. From a deployment perspective, MONDEO performs its multistage analysis directly on DNS requests, while other approaches focus on DNS replies or answers [97]. This is important for mobile network operators and ISPs, since they manage the DNS infrastructure

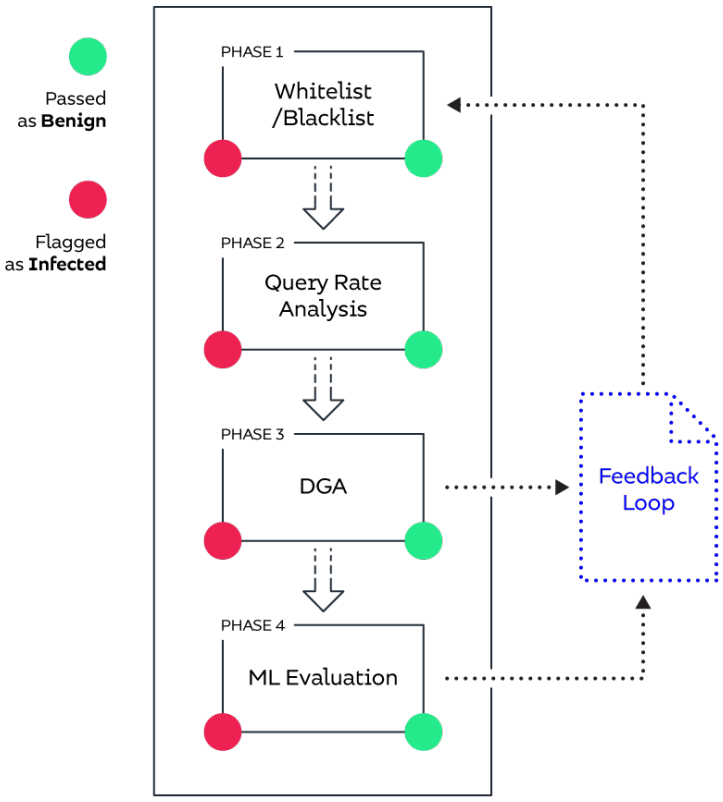



**Figure 3.4:** MONDEO overall stages and feedback loop

within their own networks, allowing botnet detection to run in parallel with DNS core services. Efficiency considerations motivated optimisations in the analysis pipeline so that processing is accelerated and packets can be accepted or discarded as early as possible, ideally during the first phase of the pipeline.

### Phase 1 - Whitelisting/Blacklisting

White and black lists are basic data structures that can store simple representations of relevant information.

The methods described above assume a full one-to-one direct match. It is also possible to adopt a similar approach using only partial matches. By analysing only a portion of the domain name, following the concept of a Free Level Domain (FLD), the lists can be made much shorter, though with some loss of precision and increased processing time. For instance, instead of keeping separate entries for every Google application, one may simply whitelist all domains ending in ".google.com".

In our proof-of-concept implementation, we opted for a straightforward solution: since the whitelist is relatively small, a traditional linear search with complexity $O(n)$ was sufficient.

The Whitelist and Blacklist mechanisms in MONDEO can be further improved by making use of the feedback loop, which incorporates the output from phase 3 (DGA evaluation) and phase 4 (ML model). This feedback enables phase 1 to be refined by adding or removing domains dynamically. Such updates may be performed manually, with a human reviewing potential candidates for the lists, or automatically within the subsequent stages of the pipeline.

**Phase 2 - Query Rate Analysis**

While examining botnet traffic manually, one observation stands out: to mimic legitimate communication with the real C2 server, an infected device generates a very large number of DNS queries. Although this behaviour benefits the attacker, it can also be leveraged by defenders to strengthen detection mechanisms.

Designing a method to identify unusually high query rates may appear straightforward, but scalability must be considered, especially if the technique is intended for deployment within the core networks of operators. A viable option would be an event-driven structure built around a doubly linked list. New packets are appended to one end of the structure while older packets are removed from the other, ensuring that only the packets within a specified time window $w$ are kept. The use of a doubly linked list is essential, as it allows efficient insertion and removal operations.

In the proof of concept implementation, we followed a simpler, yet effective approach. Instead of keeping a list with all the packets that circulate in the network for a given time window, we instead measure the time difference between every DNS query for each individual device. For example, if packet 1 arrived at timestamp $t1 = 1$ and packet 2 arrived at $t2 = 2$ then they differ in 1 time unit. With this approach, we can use parameters, to calibrate the sensitivity of the algorithm:

- $\Delta F$ which details the maximum allowed time unit interval;
- $K$ which specifies how many packets can disrespect $\Delta F$ before being quarantined.

$\Delta F$ is used to calibrate the interval sensitivity, where $\Delta F = 0$ is the shortest interval possible, and therefore the smallest possible number of packets are caught. $K$ is used to limit packet capture even if the $\Delta F$ fails. For example, in situations where a legitimate service makes 5 queries under the designated $\Delta F$ time, nothing will be reported if the $K > 5$.

**Phase 3 - DGA Detection**

Any packet that is not flagged by the query rate analysis stage, or that corresponds to a normal request, is subsequently processed by a detector that identifies domains generated

through DGA techniques. In our proof-of-concept implementation, we employed the Intel DGA detector, which is available as an open-source project on GitHub [69].

The outcomes from the evaluation performed in this phase includes a value between 0 and 1 indicating whether a domain could be or not be DGA generated, where 0 is a regular domain and 1 is a DGA-Generated Domain. With this phase, it should be once again possible to calibrate the acceptance/rejection criteria from the DGA-generator. In experiments we defined the lower bound as 0.1, which meant immediate acceptance of the packet, and upper bound as 0.9, which meant immediate packet discard. All other $0.1 \leq x \leq 0.9$ will be evaluated in the next step of the pipeline. Such threshold values were based on the sensitivity analysis that was conducted when gathering knowledge of the FluBot malware.

**Phase 4 - Machine Learning Detection**

Finally, the last stage of the pipeline is the machine learning evaluation. Only a small subset of packets should reach this phase, as they correspond to the most difficult DNS requests to classify and consequently require more processing time. This stage outputs a binary decision, where 0 indicates a non-infected packet and 1 designates an infected one. The construction of the model involves both feature selection and model training, as described next. An initial step in developing the ML model concerns the identification of relevant features. This selection was carried out based on the preliminary study of FluBot, presented in section 3.2. The chosen features, which mostly rely on fields extracted from DNS requests, are summarised in Table 3.3.

**Table 3.3:** Selected Feature Set

| Feature ID | Description | ML Data Type |
|---|---|---|
| IP Src | Source IP performing the request | Bit Conversion |
| IP Dst | Destination of DNS request | Bit Conversion |
| Length | Size of the Payload | Integer |
| DNS Flag | Info Regarding Flags | Boolean |
| DNS Questions | Number of requests in a DNS message | Integer |
| Query Type | Query Type (A, AAAA, CNAME, PTR) | Integer |
| Query Name Null | If DNS name is NULL or not | Boolean |
| Timestamp | Indication to when the packet was created | Integer |

It should be mentioned that several fields were converted into numeric representations to improve processing efficiency. In particular, IP addresses were transformed into their bit form by encoding each decimal octet of the IPv4 address. These features are not exclusive to FluBot; they can be applied to other malware families that rely on DNS traffic to discover and communicate with a C2 server. As shown in Table 3.3, the selected attributes for botnet detection are not tied to the specifics of FluBot. They are derived from standard IPv4 fields and DNS protocol elements, such as *DNS QueryType* and *DNS Questions*, which makes them suitable for identifying a broader range of malware.

Each model employs a straightforward but effective training procedure. The dataset used for

training consisted of 10,000 samples, divided evenly between synthetic packet data generated from the Alexa Top 1 Million domain list [102] and malware traffic collected in a controlled laboratory environment. This results in a balanced dataset, with an equal proportion of infected and non-infected packet examples. The trained model is evaluated using an 80/20 train–test split, where 80% of the samples are used for training the ML algorithm and the remaining 20% are used for assessing its accuracy. The implementation is carried out in Python using the scikit-learn library [103], relying on the *RandomForest* and *IsolationForest* classifiers, which demonstrated the best trade-off between accuracy and execution speed when compared with alternatives such as KNN.

### 3.3.3 C2 Server Detection

To identify the Command and Control (C2) server, it is necessary to revisit the behaviour exhibited by the malware. As noted earlier, the malware issues a large volume of DNS queries due to its use of a DGA, whose purpose is to obscure communication with the C2 infrastructure. While this behaviour is effective for detecting infected devices, it provides little direct information for identifying the active C2 servers themselves. Observations of infected devices show that an HTTP handshake takes place between the device and the C2 server to confirm the validity of the connection. This occurs once the DNS queries begin, and the DNS flood stops almost immediately after a successful connection is established. When this handshake is captured, it enables us to narrow down the thousands of possible DGA-generated domains to those that are registered and actively in use. The detection of the C2 server therefore relies primarily on efficiently capturing this handshake once infected traffic has been flagged.

#### Capturing the handshake

To successfully capture the handshake with the C2 server, a packet filter was first deployed into the network, capturing HTTP connections that match a common format, described below:

$$http \quad \&\& \quad tcp.port == 80 \quad \&\& \quad http.request.method == \text{“}POST\text{"}$$

By applying the described filter, it becomes possible to isolate all HTTP POST communications associated with potential C2 servers. The endpoints identified through this process often correspond to domain names that appeared in earlier DNS queries. To improve the precision of the detection, the Uniform Resource Identifier (URI) within the HTTP packets is examined [69]. When the predicted score exceeds a user-defined threshold, the traffic is classified as infected. Fig. 3.5 illustrates the request URI found in a filtered packet directed to the C2 server `tofelumrdyrgwhg.ru`, which can be readily recognised as malicious. This straightforward check significantly reduces the likelihood of false positives. Alternative methods may

additionally inspect the HTTP response status code, where a value of 200 indicates that the request was successfully processed.

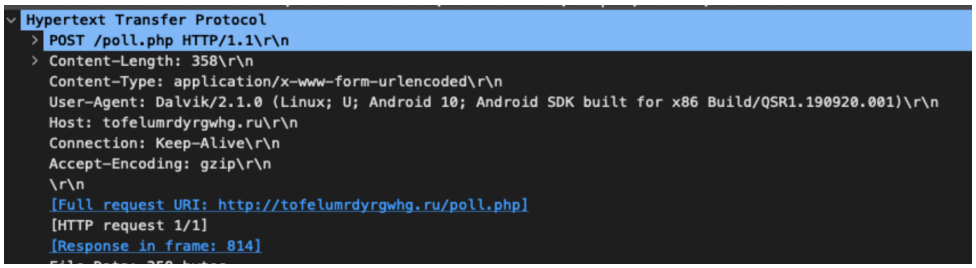



**Figure 3.5:** Wireshark detail on possibly infected packet

**Implementation of detection mechanisms in MONDEO**

Considering a microservice architecture, and that packet capture occurs within other processes/approaches, for instance using the UPF in a 5G network, the mechanism works as follows:

1. Based on the DNS analysis (Section 3.3.2), the system establishes a list of possibly infected devices (time-based, such that false-positives are not triggered).
2. Whenever a packet arrives , it is analyzed with regards to the source, if it is present in the infected list - *phase 1* the observed query ratio (Section 3.3.2), and with regards to the name/URI requested in DNS request (i.e., if DGA based or not). If meeting any of the criteria in the diverse phases it is considered infected (Section 3.3.2).
3. Otherwise the final evaluation is carried out, considering:
   - Infected, if both conditions in phase 3 are true (Section 3.3.2);
   - Not infected, otherwise.
4. Perform the detection of the C2 server.
5. Based on this evaluation, a JSON response is produced and provided through the REST API.

The implementation adopts a microservice architecture, enabling efficient parallel processing for botnet detection and the retrieval of the information required to apply mitigation tactics. This architectural choice also simplifies deployment in 5G and 6G environments, where network functions are commonly delivered as microservices. It is important to note that the identification of the C2 server is guided by behavioural patterns observed in FluBot samples. The detection mechanism in MONDEO was evaluated using six distinct FluBot variants, each generating a high volume of DNS requests during C2 discovery, which allowed us to refine and

improve the accuracy of the detection process, as described in Section 3.4. Once a C2 server is detected, MONDEO communicates this information to the tactics component together with an associated probability value, allowing the appropriate risk-mitigation strategy to be selected, as detailed in the following section.

### 3.3.4 Tactics

Once malware is detected and this information is communicated through the *M.R* interface in the MONDEO architecture (Fig. 3.3), it becomes necessary to determine which mitigation tactics should be applied to address the associated risk. To support this decision process, we employ Rainbow [22], a self-adaptation framework capable of reacting to the information supplied by MONDEO. In brief, self-adaptation relies on a control loop composed of four core activities: **M**onitoring, **A**nalysis, **P**lanning, and **E**xecution, all of which operate using shared **K**nowledge regarding the system and its surrounding environment. Together, these activities form the well-known MAPE-K loop [15]. In this loop, the monitoring phase updates the Knowledge component with information about the system and its environment, represented through a model. This model incorporates data from MONDEO, including the current state of infected devices and any identified malicious C2 servers. The analysis phase processes this information to derive higher-level insights. In our case, MONDEO also supplies probability values associated with suspicious behaviours. Rainbow's analysis stage evaluates these inputs to determine whether an adaptive response should be triggered.

Rainbow uses the information from the detection of MONDEO to maintain a model of the system and its environment. From this information, it can decide if the system is not behaving as desired and develop or choose a plan to fix/mitigate identified problems. For such purpose Rainbow uses the utility theory [22]. A planner within Rainbow uses this information and predicts the effects of various tactics on future system utility. The tactic(s) that maximize the utility are the ones that Rainbow then selects and executes. These tactics are discussed below.

The interface between MONDEO and Rainbow occurs through the *M.R* interface, as depicted in Fig. 3.3. The data that is exchanged includes:

- Timestamp of the identification event;
- Identification of infected device (IPv4 address);
- Identification of C2 server (IPv4 address and FQDN);
- Level of certainty in the detection.

The integration between MONDEO and Rainbow is achieved through the use of *probes*. In this configuration, MONDEO functions as a probe (or monitor) within Rainbow, periodically send-

ing updates that are used to refresh the system model. When Rainbow receives information about infected devices and identified C2 servers, it applies the necessary policies to mitigate the detected threat. In our approach, we focus on three representative tactics for responding to malware. These include quarantining the affected mobile device, “blackholing” the C2 server by resolving its DNS entry to a benign IP address, and issuing a CAPTCHA challenge to the user whenever a suspicious connection attempt is identified.

Quarantining operates at the level of individual mobile devices, with each device isolated separately. This tactic does not scale efficiently and carries a higher risk of false positives, as legitimate devices may be unnecessarily restricted. In contrast, blackholing the IP address of the C2 server is performed centrally by the mobile network operator and affects all devices that rely on the operator’s DNS servers. This makes blackholing more scalable and easier to enforce as a global policy. It should be noted that DNS security mechanisms, such as DNSSEC, may complicate the effective use of blackholing, although these challenges fall outside the scope of this work. Whereas quarantining acts at the device level and blackholing at the network level, CAPTCHA challenges operate at the connection level. They temporarily interrupt the connection until the user demonstrates, through interaction, that the request originates from a human rather than malware or automated software.

At the same time, we model three impacts that influence the overall utility: the impact on customer service department utilisation for the mobile network operator, the intrusiveness to the customer experience, and the effectiveness of attack containment.

Details of tactic implementation can affect the impacts on the utility. For example, if the quarantine uses a list of known-bad sites to prevent malicious connections, the impact to the end user will be minimized while the effectiveness will also be diminished versus a quarantine approach that uses a list of known-good sites. The known-good sites approach is more limiting and will result in higher impacts to the mobile user and mobile network operator while also likely being more effective.

### 3.3.5 5G/6G Networks

The three tactics are implemented differently within a 5G or beyond network environment. Among them, the blackhole tactic is the simplest to deploy. It involves sending an update to the DNS server instructing it either not to resolve a particular domain or to resolve it to a benign address. This mechanism can be supported through the cooperation of 5G network functions such as the UPF or the PCF, as illustrated in Figure 3.3.

The quarantine of a device can be implemented using technologies like Network Function Virtualization (NFV) to create a network slice for quarantined devices. These devices can have a separate DNS service or have their connections mediated through a user plane function

(UPF) that limits to known-good (or not known-bad) connections depending on the specific implementation.

The implementation of a CAPTCHA-based tactic is more complex than the others. It requires components within the 5G core network that are capable of identifying the flows that should be temporarily interrupted, presenting the CAPTCHA challenge to the user, and, upon successful completion, maintaining the necessary state to reopen the connection and allow subsequent reconnections. The UPF and the Session Management Function (SMF) are suitable 5G core functions for performing this type of connection tracking and managing the CAPTCHA workflow. However, this approach may face scalability limitations, particularly given the large number of devices concurrently connected to the network and their heterogeneous capabilities, such as cases where devices do not support the HTTP protocol.

From a network operator's standpoint, botnet detection must be integrated with the 5G core functions that handle mobile device traffic in the data plane. The UPF is a strong candidate for this purpose, as it can support packet capture or deep packet inspection capabilities needed to identify botnet activity, as outlined in Section 3.3.2. Such traffic analysis is essential not only for detecting infected devices but also for determining the corresponding C2 servers. The inspection process may also take advantage of existing 5G features such as network slicing, in which a UPF instance can be deployed within each slice.

When focusing solely on detecting botnet traffic, capturing DNS requests is generally sufficient. Other malware families, however, may require full packet inspection. The detection process is further simplified when operators manage their own DNS infrastructure or rewrite DNS responses with their own results.

## 3.4 Evaluation

This section describes the evaluation methodology.

### 3.4.1 Datasets

To be as realistic as possible, we configured a DNS server using ISC BIND, where we collected information regarding DNS queries. The collection included the DNS logs and the capture of DNS packets with tshark tool. The information in the datasets included the DNS packets that were captured from regular DNS clients, of several volunteer participants that configured their devices to use the configured DNS server.

To produce infected traffic, we executed an isolated virtual machine where malware-embedded applications (APKs) could be deployed safely, as summarised in Table 3.4. The malware samples were obtained from public repositories such as MalwareBazar [6] and Koodous [54]. The

emulated device profile corresponded to a Pixel 4 running Android API 29. This device was selected because its computational characteristics are representative of commonly used smartphones worldwide. It has also been adopted in other studies that analyse mobile device behaviour [92, 105]. Each emulated device instance was activated at different times, and the resulting malware traffic — referred to as $\mathrm{infected}$ — was captured in individual tshark trace files. It is important to note that during the emission of malicious requests, $\mathrm{Benign}$ DNS traffic from normal operation was also being generated.

**Table 3.4:** FluBot Malware sample information

| Name | Malware File(s) | Description |
|---|---|---|
| Correos FluBot | Correos1 | Application that mimics Correos app |
| FedEx FluBot | FedEx1, FedEx2 | Application similar to FedEx app for tracking |
| UPS FluBot | UPS1 | Application that mimics UPS app tracking features |
| DHL FluBot | DHL1, DHL2 | Application similar to DHL app for tracking |
| VoiceMail FluBot | VoiceMail1 | Application similar to VoiceMail app for voice mailing |

### 3.4.2 Botnet Detection

The evaluation of MONDEO-Tactics5G includes the performance characterization of each phase in the MONDEO data pipeline, as documented in section 3.3.2. Performance is assessed in terms of the time required to process a packet in each phase and the overall number of packets that are processed.

**Table 3.5:** Tests Information in the Evaluation of Data Pipeline

| Test Type | File(s) | Description |
|---|---|---|
| #1 Infected | FedEx1, FedEx2, UPS1, Correos1, DHL1, DHL2, VoiceMail1 | Lab With samples of malware |
| #2 Benign | 23 | Only with regular DNS requests |

All the tests rely on the DNS samples that were collected in the DNS Experimental Setup, summarized in Table 3.5 and that were collected using the methodology documented in section 3.4.1. In this evaluation we focus on testing the impact of the malware, especially with regards to the HTTP handshake (discussed in the next section). From this list, a data set with the features summarized in Table 3.3 was built to assess the performance of MONDEO and the application of tactics. The metrics employed to assess the performance of the MONDEO Data Pipeline are summarized in Table 3.6.

**Table 3.6:** Performance Metrics in the MONDEO Data Pipeline

| Metric | Unit | Description |
|---|---|---|
| Packets Processed | % | Ratio of packets processed in each phase, considering the total of captured packets |
| Processing Time | ms | Time required to process a packet in each phase |
| Classification | n/a | Final classification of MONDEO, if packet is flagged as infected or as a regular/benign reques |

### 3.4.3 C2 Server Detection

To evaluate the performance of the C2 detection, we consider the ratio of HTTP requests analyzed, in terms of passed or flagged as infected (recall Fig. 3.4). Resource consumption is also considered, in terms of CPU usage, memory usage ratios and the amount of information that is exchanged and measured in bytes.

### 3.4.4 Tactics

To evaluate the effectiveness of tactics, we model the system and then use statistical model checking to show which combinations of tactics would be most effective in each state. We take this approach because we did not have access to a real 5G network, nor a simulator of it. In this evaluation approach, however, we can show that in all cases tactics would improve the overall utility of the network.

#### Model

We model our system as a Discrete-Time Markov Chain (DTMC) using the PRISM probabilistic model checker and its modelling language [56]. To avoid the complexity and scalability challenges that would arise from modelling each individual device and server, we instead organise them into device and server *tiers*. Elements within the same tier are assumed to share a comparable probability of being compromised, and therefore the tactics in our framework are applied at the tier level rather than at the granularity of individual devices or servers. For clarity of presentation, rather than providing the full PRISM code, we illustrate in Figure 3.6 the interaction dynamics among the processes (referred to as *modules* in PRISM terminology) that represent the behaviour of our system's main components: the device and server tiers, the connectors between them, and the customer service element. Each device tier contains a state variable compromised, ranging from 0 to 100, which expresses the probability that the tier is affected by malware. Server tiers maintain a corresponding variable c2, which indicates the likelihood that a server tier is functioning as a C2. The customer service module includes a state variable utilization, also between 0 and 100, representing the load on customer service, that is, whether agents are occupied assisting users who report service disruptions.

Each device tier is associated with a maximum number of attempts,MAX_REQUEST_ATTEMPTS to communicate with the server. For every attempt, if the device tier is not currently quarantined, it may issue a request to a server through the connector. Once the connector receives this request, it forwards it to the appropriate server tier provided that the CAPTCHA mechanism for the originating device tier is not active. When the CAPTCHA is enabled, the request is forwarded only with a probability pCaptcha, which is inversely related to the compromised value of the device tier sending the request (that is, the greater the likelihood of compromise, the smaller the probability that the request passes the CAPTCHA). After the server tier generates a response, the connector checks whether that server tier is blacklisted; if it is not, the response is returned to the corresponding device tier. When the device tier receives the response, the following updates occur:

- The value of compromised is updated by adding to it the c2 of the server that sent the

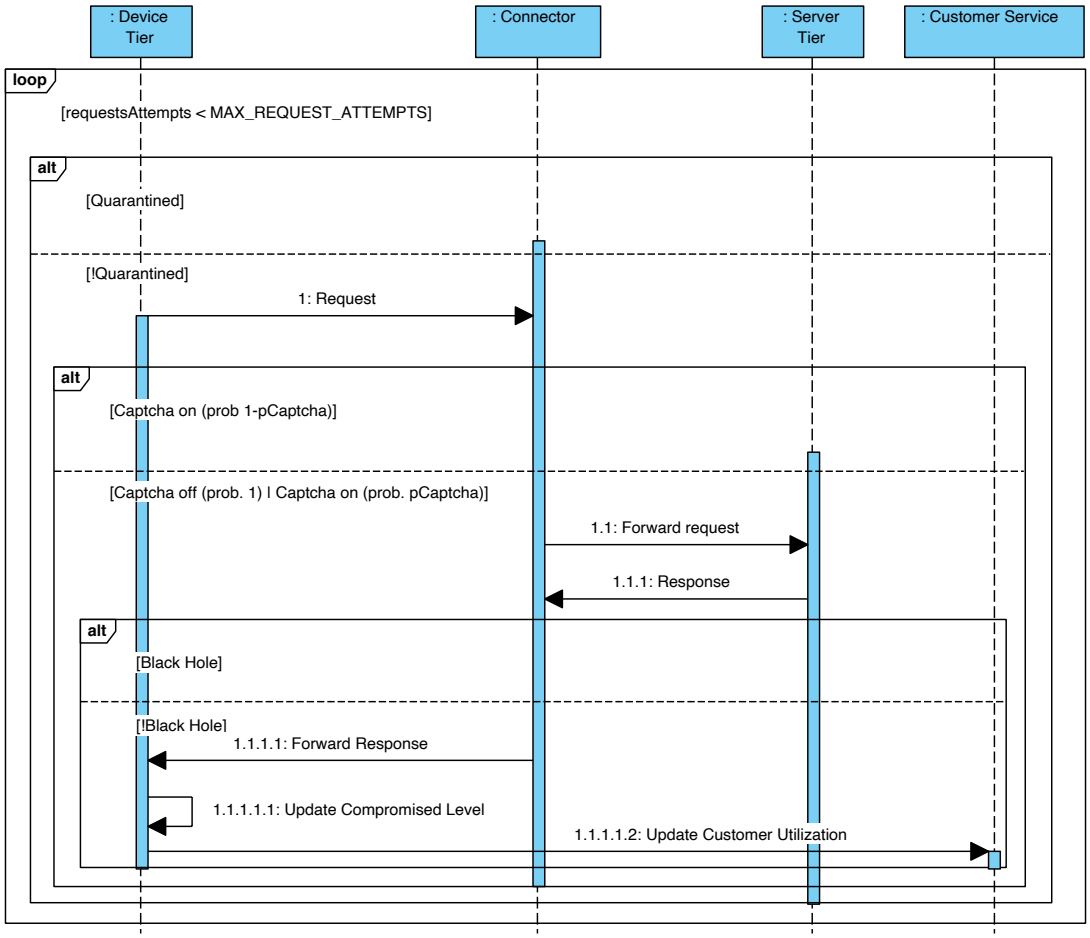



**Figure 3.6:** Sequence diagram describing interactions of the PRISM model.

response.

- The value of utilization is updated in the customer service to reflect the new levels of customer service utilization. Updates are:
  - Directly proportional to compromised and c2 levels.
  - Inversely proportional to an accuracy in malware detection parameter designated by $\alpha$.

The magnitude of the updated value depends on which tactics are enabled. Concretely, blacklisting, captcha, and quarantining have multiplicative factors that capture the increasing level of disruption in services that these tactics can introduce (with quarantining being the most disruptive):

$$utilization' = \sum_{t \in T} (1 - \alpha) m_t * \sum_{j=1}^{n} compromised_j * e_t i \tag{3.1}$$

In the expression above, $T = \{captcha, quarantine, blacklist\}$ is the set of available tactics, $m_t$ is a multiplicative factor that models the disruption of different tactics, $n$ is the number of device tiers, and $e_t i$ is 1 if tactic $t$ is enabled in device tier $i$, and 0 otherwise.

To measure the value provided by the system during execution, we consider a utility function $U : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ defined as a linear combination of two terms that correspond to the level of utilization of the system, and the effectiveness of the adaptation tactics (i.e., in terms of minimizing the likelihood of devices being compromised):

$$U(u, e) = w_u * u_u(u) + w_e * u_e(e) \tag{3.2}$$

In the expression above, the utilization utility function $u_u(*)$ returns an output between 0 and 1 that is inversely proportional to the utilization value provided as input, whereas the effectiveness utility function $u_e(*)$ takes as input an effectiveness value that corresponds to the mean of the compromised values across all device tiers, and returns a value between 0 and 1 that is inversely proportional to it.

Our model incorporates a reward structure that enables storing information about accrued utility. During execution, an amount of utility equivalent to the result of Expression 3.2 is accrued on the reward structure at the end of every cycle of the loop depicted in Figure 3.6. We designate the amount of accrued utility during the execution of a scenario as $u_{mau}$ (mean accrued utility).

**Analysis**

To evaluate the effectiveness of the tactics, we take an approach similar to the evaluation in [20], where we analyze the system using the statistical model checking engine of the PRISM probabilistic model checker. We model check the system to understand the effect of various strategies on the utility of the system, comparing versions of the system with and without adaptation.

To quantify the utility that each strategy yields, we make use of Probabilistic Reward CTL (sPCTL) [14], which extends the probabilistic temporal logic PCTL [17] with reward-specific operators aimed at the specification of performability measures over DTMC models. Specifically, our technique enables us to statically analyze a particular region of the state space, which first has to be discretized to check PRCTL properties. Obtaining the results of the analysis for each state in the discrete set requires an independent run of the model checker in which model parameters are instantiated with variable values that correspond to that state. In our case, the discrete set we consider corresponds to pairs $(\alpha, w_u)$ in the range $[0, 1]$, where the discretization step for accuracy is $\mu_\alpha = 0.1$ and the one for $w_u$ is $\mu_{w_u} = 0.05$.

For each independent run of the model checker, we analyze a PRCTL property that employs the reward operator $\mathsf{R}\{\mathsf{r}\}_{=?}[\mathsf{F}\phi]$, which enables the quantification of the accrued reward r along paths in a model that eventually reach states that satisfy the reachability predicate $\phi$. Concretely, we analyze the property $\mathsf{R}\{\mathsf{utility}\}_{\mathsf{max}=?}[\mathsf{F\ done}]$, where utility is the name of the reward structure in our PRISM model, and done is a label that corresponds to an expression over state variables that captures states in which devices can no longer perform requests (because MAX_REQUEST_ATTEMPTS has been reached).

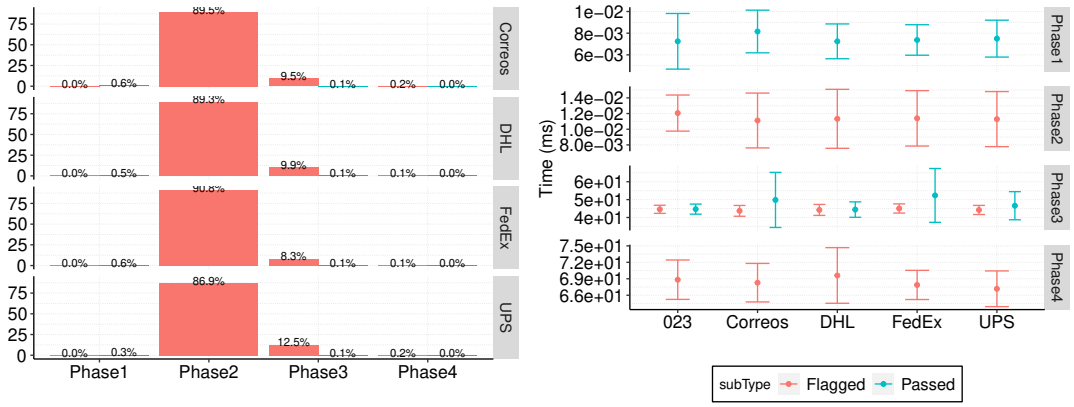
## 3.5 Evaluation Results

This section summarizes the results, considering the evaluation methodology previously described.

### 3.5.1 Botnet Detection

MONDEO-Tactics5G uses a multistage and feedback loop to detect packets from FluBot malware. As illustrated in Fig. 3.7-a the majority of the detection, for the used samples is performed at phase 2, which assesses the query rate. In this phase the majority of requests is flagged as malware due to the high number of requests per second. In the evaluation results, the feedback loop was not used, as one can see in the Fig. 3.7-b since there is no flagged time in phase1 and the passed is almost zero. The phases with more impact, in terms of processing time are phases 3 and 4, which use the DGA algorithms and the ML models, respectively. In

these phases the processing time is in the order of 400 ms, either to flag or to allow a packet to pass.




**(a)** Detection Rate **(b)** Detection Time

**Figure 3.7:** Detection rate and time in each MONDEO phase

### 3.5.2 C2 Server Detection

The C2 server detection is assessed in terms of the detection accuracy regarding the HTTP requests towards the C2 server. Fig. 3.8 depicts the results of HTTP identifying the HTTP

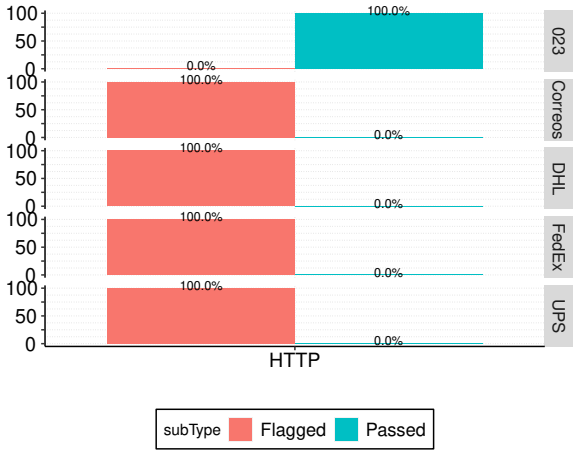
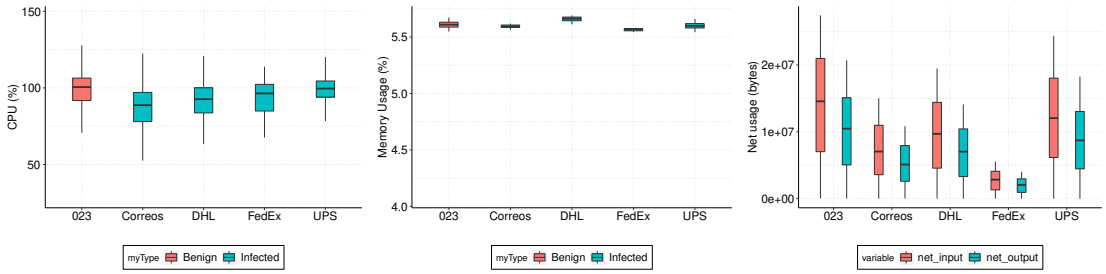



**Figure 3.8:** Successful identification of HTTP requests to the C2 server(s)

requests towards the C2 server. In the 023 dataset all the requests correspond to legitimate HTTP requests, while in the datasets infected with FluBot there are HTTP requests to the C2 server(s) and benign HTTP requests, which correspond to browsers requests. The majority of the HTTP requests in the datasets with malign samples are for the C2 server(s). Either the botnet detection and the C2 server detection lead to resource consumption, as illustrated in Fig. 3.9, in terms of CPU, memory and network I/O. The results report the resources consumed by MONDEO which is implemented as a microservice with APIs for botnet detection and HTTP analysis. MONDEO has an impact in terms of CPU usage due to the required analysis; nonetheless the impact on the memory usage is low. In addition, the microservice exchanges

a small amounts of data, where the input is higher, since it contains information of the DNS and HTTP packets. The output is provided in JSON format, having significantly lower volume. The amount of information exchanged in normal conditions - case 023 is superior, since DNS packets contain legitimate requests.




**(a)** CPU usage ratio **(b)** Memory usage ratio **(c)** Network I/O

**Figure 3.9:** Resource usage ratio

### 3.5.3 Tactics

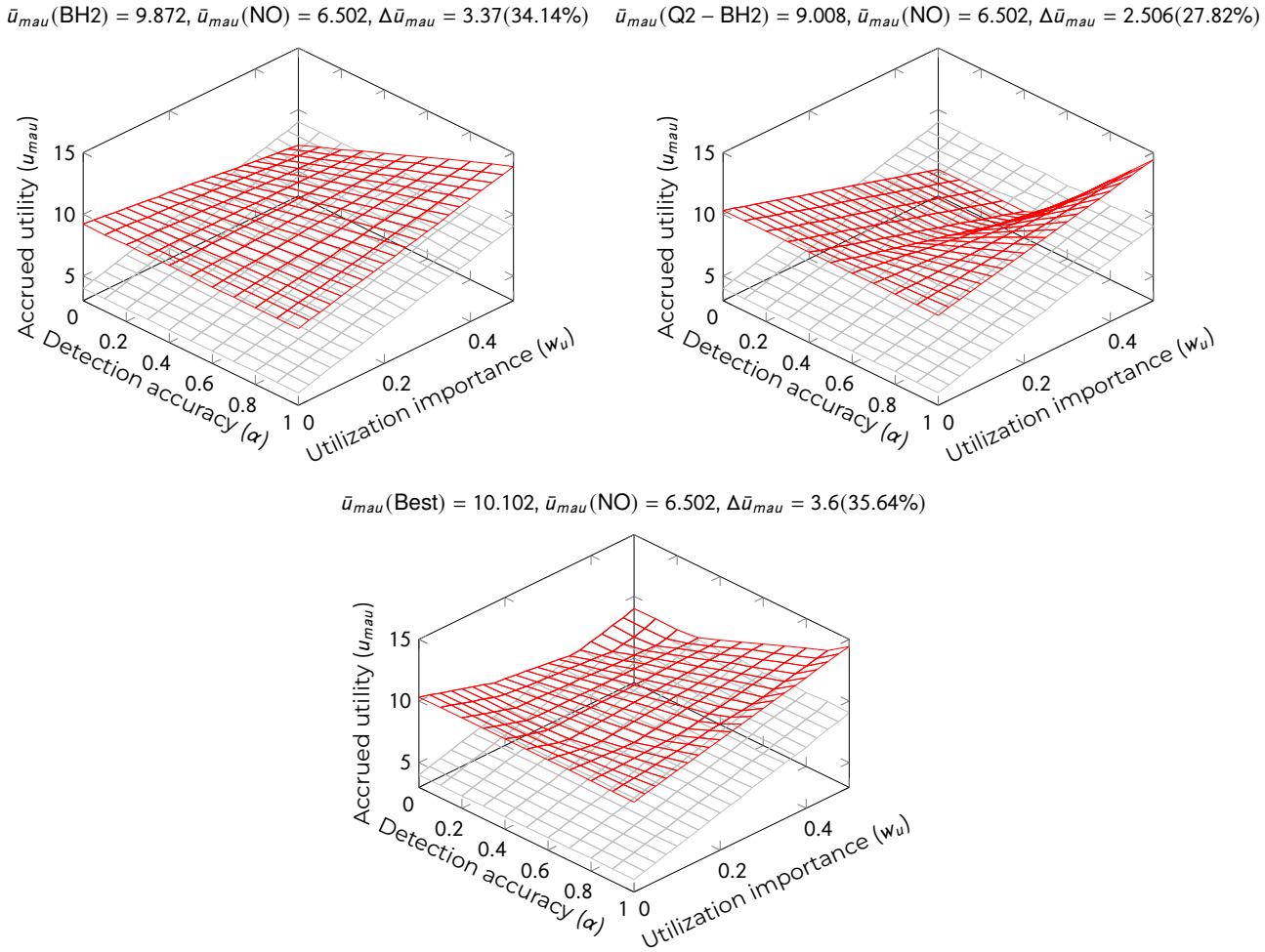



**Figure 3.10:**
Experimental results comparing accrued utility with and without adaptation strategies: Blacklisting (top, left), blacklisting combined with quarantining (top, right), and best strategy (bottom).

Figure 3.10 presents the experimental results for a scenario where we compare the total utility accumulated by the system during execution, both with adaptation enabled and without it. The system consists of four device tiers and four server tiers, each defined by an increasing probability of being compromised by malware (for example, the first device tier ranges from 0 to 25%, the second from 25 to 50%, and so forth). The horizontal axes in the plots span the interval from 0 to 1 and represent the accuracy with which the system identifies the likelihood that a device is compromised ($\alpha$, assumed the same for all device tiers in this experiment) and the weight assigned to the utilization component of the utility function $w_u$. The vertical axis reports the accumulated utility $u_{mau}$ for each pair of values of $\alpha$ and $w_u$. Each plot contains grid points that illustrate the performance of the system operating without adaptation (in gray) and with adaptation (in red). The plots in the figure correspond to strategies that use different combinations of tactics from the following set:

- Q2 Quarantines the two top tiers of devices, i.e., those with a likelihood of being compromised by malware ranging between 50 and 100%.
- C2 Captchas the two top tiers of devices.
- BH2 Blacklists the two top tiers of servers.
- Q1 Quarantines the top tier of devices (75-100%).
- C1 Captchas the second from top tier of devices (50-75%).

Each strategy is labeled by the combination of tactics it uses. In the Figure, NO refers to a strategy that does nothing (i.e., it does not execute any tactics), and Best corresponds to a strategy that picks at each point ($\alpha$, $w_u$) the best of all available strategies (including NO). The set of strategies analyzed corresponds to C2-BH2, C2, Q2, BH2, Q2-BH2, Q1-C1, Q1-C1-BH2. However, we only represent in the Figure strategies that are optimal at some point of the space (Q2-BH2 and BH2).

All adaptation strategies (including those not shown in the figure) achieve, on average, higher accumulated utility than the non-adaptive version of the system. The improvement in mean accrued utility ($\Delta\bar{u}_{mau}$) is consistently positive, ranging from 1.62When examining the configuration without adaptation (the gray grid, which remains identical across all plots), it becomes clear that lower values of utilization weight are associated with reduced utility, whereas higher utilization importance leads to larger utility values. This behaviour is expected because, in these regions, the utility originates predominantly from the effectiveness term of the utility function, and device tiers that are highly compromised contribute very little to this component. Turning our attention to the adaptive scenarios (red grids), a consistent trend emerges across all plots: higher detection accuracy generally leads to increased accrued utility, particularly when combined with larger values of $w_u$. This is a consequence of fewer legitimate

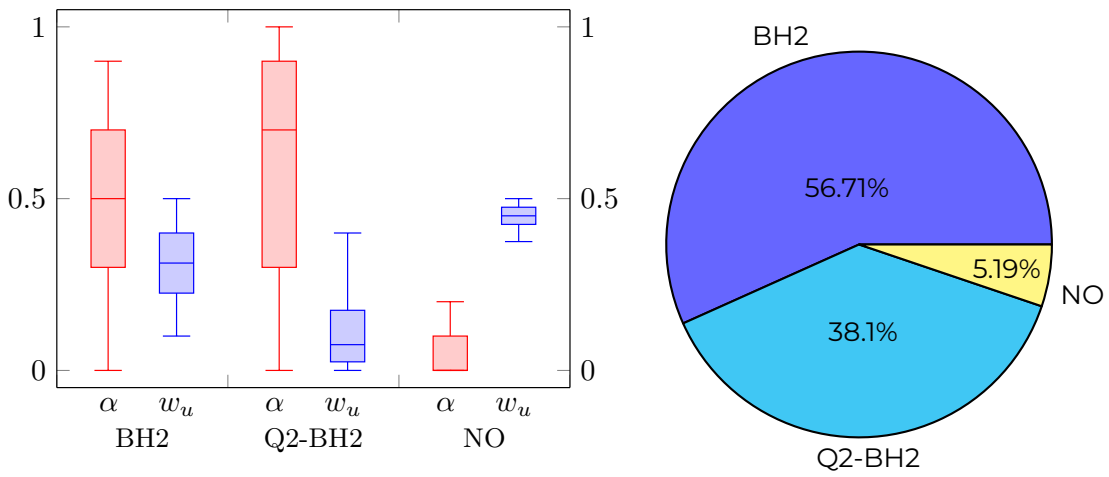



**Figure 3.11:**
Accuracy and utilization importance conditions and coverage: Use context of best adaptation strategies: accuracy and utilization importance conditions (left) and coverage (right).

users being disrupted when accuracy is high, which in turn keeps customer service utilization lower, thereby increasing the utility derived from the utilization term. It is also notable that when utilization weight is high but detection accuracy is low, the utility becomes significantly reduced, as more legitimate users experience disruptions and customer service becomes overloaded. In fact, the Best strategy (bottom plot) shows that when utilization importance is very high ($w_u \simeq 0.5$) and detection accuracy $\alpha$ is very low, the optimal choice is to avoid applying adaptation tactics altogether, as this minimises disruptions to legitimate users.

Figure 3.11 depicts the conditions under which different adaptation strategies achieve the highest performance. The left side of the figure presents a boxplot summarising the average, maximum, and minimum values of detection accuracy and utilization importance for which each strategy outperforms the others. From this representation, it is evident that strategies BH2 and Q2-BH2 dominate across a broad portion of the parameter space. When utilization importance is very low, Q2-BH2 tends to outperform Q2, which is consistent with the fact that, in scenarios where utility is derived almost entirely from the effectiveness component of the utility function, adding quarantining to server blacklisting more effectively keeps device compromise probability low. The boxplot also indicates that for high utilization importance combined with low accuracy, avoiding adaptation tactics altogether is the best course of action. This observation aligns with the behaviour shown in Figure 3.10. The right side of the figure provides a pie chart illustrating the proportion of the parameter space in which each mitigation strategy is optimal. The chart shows that in approximately $\simeq 5\%$ of the evaluated configurations, the best decision is not to execute any tactic, corresponding to the region characterised by high utilization importance and low detection accuracy described earlier. Additionally, strategies that include BH2 clearly dominate the parameter space, yielding the best performance in more than 94% of the cases considered.

### 3.5.4 Credit authorship contribution statement

**Paper:** MONDEO-Tactics5G: Multistage botnet detection and tactics for 5G/6G networks
**Doi:**https://doi.org/10.1016/j.cose.2024.103768
Elsevier, Computers & Security Volume 140, May 2024, 103768

**Pedro Fidalgo**: Investigation, Supervision, Writing – original draft.
**Bruno Sousa**: Investigation, Supervision, Writing – original draft, Writing – review & editing.
**Duarte Dias**: Investigation, Software, Writing – original draft.
**Nuno Antunes**: Investigation, Writing – original draft.
**Javier Cámara**: Investigation, Writing – original draft.
**Ryan Wagner**: Investigation.
**Bradley Schmerl**: Investigation, Writing – original draft.
**David Garlan**: Investigation, Methodology, Supervision.

# Chapter 4

# Static topological multidimensional subgraph analysis to detect fraudulent nodes and rings in telecom networks

## 4.1 Introduction

Every network has a purpose and a goal. The structure and node organization inside the network unveils link distribution, strengths and peculiar features often associated to specific types of networks. Fraud networks, like covert networks try to hide its leaders and be resilience to disruption. On one hand, it protects the most important nodes of the network, not only by avoiding direct exposure, but also possible prosecution if any of these nodes are compromised. Distinct positions in the network drives to different judicial outcomes. Nodes with high degree centrality are more likely to be arrested, on the opposite of nodes with high betweenness centrality [71]. Snake like shapes, where nodes are 9 degrees away from each other are also described as a strategy that minimizes damage to the network if one of the nodes is exposed [55]. Typically, perpetrators or executors of malicious actions, have the highest chances of being caught, they are peripheral on the network with limited opportunities to increase their degree. In essence the network limits their tie opportunities protecting the most important nodes [81]. Secrecy and security are strategies taken into consideration in the formation of fraud networks, since they increase resilience to attacks and exposure. Sparse and thin networks with weak ties ensure network survivability, on the opposite of dense interconnected networks with high group centrality which are more exposed and vulnerable [89]. In this

aspect, number of ties and level of centrality can indicate the role of the node in the network. Informally, we have the following research problem:

### 4.1.1 Problem Definition

- **Given**
  - Who calls whom network
  - Fraud labels for some of the nodes
- **Find**
  - Suspicious fraud nodes and networks
  - Patterns of ring associations
  - A group of features that characterize fraud nodes and networks

### 4.1.2 Contributions

The contributions of STARBRIDGE are as follow:

- **Scalable**, it scales linearly with the network size
- **Ranked Out**, it provides a ranked score for each node in the network
- **Parameter Free**, requires no parameter tuning
- **Novel Discoveries**, STARBRIDGE led to the identification of nodes with association to fraud networks and network patterns that reveal ring associations.

The rest of the chapter is organized as follow. Chapter 4.2, provides a brief review of related work. Chapter 4.3, describes the proposed method, STARBRIDGE. Chapter 4.4, experiments with the method and analyses the results obtained and finally Section 4.5 conclusion and overview.

## 4.2 Related Work

Figure 4.1 was obtained from a Telecom Fraud Management System. It contains the network of all nodes that where marked as suspicious fraud. It is possible to identify an emerging pattern in the network: bridge nodes interconnecting stars, a backbone network that interconnects the major hubs of fraud.

Previous work [81] notes that closeness reflects how easily a node can reach others in the network, corresponding to a supervisory capability, while betweenness captures the extent

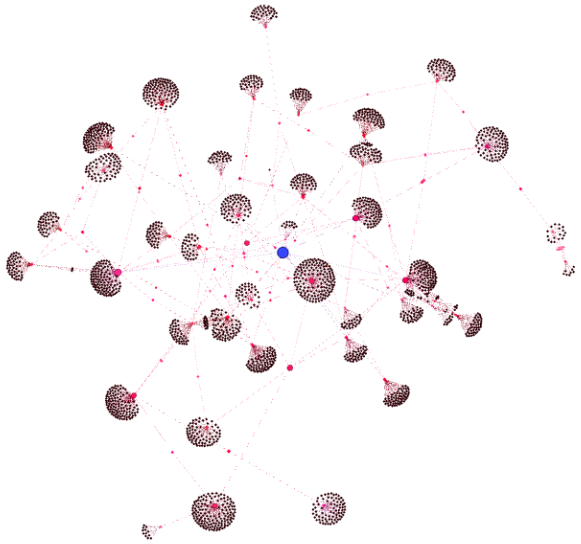


**Figure 4.1:** Suspicious Fraudulent Nodes and Edges [Star-Bridge] pattern.

to which a node governs information flow, acting as a broker. Nodes situated around major hubs often occupy positions of power and reveal hierarchical divisions within the network. Additionally, certain nodes may not exhibit high degree or bridge distant regions, yet they occupy strategically advantageous locations near hubs or dense clusters, giving them privileged access to information and network resources. For these reasons, centrality measures such as degree, closeness, and betweenness have become some of the most widely used approaches for estimating node influence. However, centrality metrics have well-known limitations. They often overlook the influence of non-hub nodes [18], fail to quantify differences between nodes with similar profiles [16], and are strongly affected by network topology [95]. Centrality measures that perform well for identifying influential nodes at a global network scale may be poor at capturing local structural context, where metrics such as eigenvector-based approaches (e.g., PageRank, Katz Centrality) are more suitable. Indeed, more than 200 centrality measures have been proposed [75], each sensitive to specific network dynamics and structural assumptions. Moreover, many centrality measures are highly correlated, making it difficult to detect meaningful strategic patterns related to node position or function [19]. With respect to network structure, three primary methodological families are traditionally used for modelling communities and structural patterns in graphs: **Subgraph Analysis**, **Label Propagation**, and **Latent Factor Models**.

### 4.2.1 Subgraph Analysis

Subgraph Analysis is a method that identifies patterns in graphs, with focus on ego-nets and induced subgraphs from its neighbours. They look for “strange nodes” in a social perspective either because they are highly or poorly connected, or reveal “strangeness” in the direction and the weight of its connections, either full inbound links (Blackholes) or full outbound links

(Volcanos) or unbalanced weight distribution (heavy links) [65] [9] [27] .

### 4.2.2 Label propagation

Label propagation assigns information from labelled nodes to unlabelled nodes through a graph. From an initial subset of labelled nodes, this method constructs a graph based on node-to-node similarity. In the case of nodes connecting to multiple other nodes with different labels, multilevel graphs can be built based on their connected attributes. This class of algorithms aims to identify suspicious nodes based on their connectivity and synchronized behaviour. Real and complex networks tend to have community structures and label propagation methods can find communities by analysing each single entity under the assumption that nodes that are connected in the graph are likely to share the same semantic label. Thus, label propagation algorithms can also infer fraudulent from a priori legitimate nodes based on their suspicious clique membership and relationships to known fraudsters in the network [51] [106].

### 4.2.3 Latent Factor Models

Latent Factor Models (LFMs) constitute the third traditional class of approaches. In this setting, latent refers to variables that cannot be directly observed but are inferred from measurable ones. These hidden factors are typically derived through matrix factorization applied to observable features. Within a graph setting, the pattern of observed interactions among nodes can reveal underlying structural properties that help anticipate unobserved or future links. Latent models are often combined with machine learning or Bayesian techniques to uncover the latent structure that best explains the observed relationships [84]. Considering the behaviour of covert or fraudulent networks, the nodes that matter most are those that balance structural positioning with local influence. Crucially, influential actors often avoid occupying visibly dominant positions, thereby reducing exposure to detection while retaining effective control. This reflects an inherent trade-off between minimizing risk and preserving operational power within the network [19]. In this context, STARBRIDGE belongs to the family of subgraph analysis methods. It evaluates each node along multiple dimensions, focusing on its role as a Bridge [86], its Control characteristics [67], and its Influence within the neighbourhood [58]. The objective is to rank nodes and identify fraudulent actors and ring structures. The first two metrics are strongly shaped by network topology, whereas the influence metric captures the strength of a node's position relative to its neighbours. Taken together, these attributes provide a multidimensional score for each node, integrating structural role, local influence, and centrality to highlight strategically important positions within the network.

## 4.3 Proposed Method

Fraud is a composite force, driven by technical schemes and human psychology. In this context, nodes with better strategic positioning that allows them to maintain control, avoid exposure and ensure resilience, are key to succeed. An interplay between roles that can we proposed to be measured assessing Bridging Centrality, Control and Influence.

### 4.3.1 Control (Driver Nodes)

To ensure a specific goal in a network the relations between the nodes must support a set of mechanisms or processes. This is often accomplished through controllability, i.e., the ability to steer a system into an arbitrary final state in a finite time [86]. Nodes involved in control, influence the evolution of a system through control strategies, characterized by controllability profiles [116]. Albeit most real systems in terms of processes are nonlinear; in many aspects they are structurally similar and governed by linear, time-invariant dynamics.

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \tag{4.1}$$

The state dynamics of a system of N nodes at a time t, is characterized by its adjacency matrix $A$, and a set of controlling nodes (or drivers), $B$, and control signals. A system is controllable if it meets Kalman's controllability rank condition. Hence, equation (4.1) needs to respect the condition on equation (4.2) where its controllability matrix, $C$, has full rank.

$$rank(C) = rank(B, AB, A^2B, ..., A^{(N-1)}B) = N \tag{4.2}$$

Typically, hubs are nodes that play a chief role in control processes and are typically identified as drivers in matrix $B$ [86]. While Structural Controllability accesses whether control is theoretically possible in a network, nodes with control reveal different properties in distinct graph models. Previous work identified different node level controllability metrics [116], which highlights how certain nodes, may be predisposed to drive the system in a multitude of states, using average, modal, and boundary controllability strategies. While this work was originally performed on brain networks [41], it was further extend to other types of graph models and consequently network structures [116] [28] [125] [107]. One important concept is the Boundary controllability [41], which describes which driver nodes (boundary controllers) can act to synchronize or desynchronize communities in the network. A high number of real world networks have a scale free topology (Barábasi-Albert model), and results comparing distinct graph models demonstrated that Barábasi-Albert models have higher boundary controllability values and variance [116], suggesting the correlation between network topology and the role of these nodes in real world networks. Procedures to detect boundary controllers

vary from bi-partitioning the original network recursively into smaller subnetworks using Fiedler eigenvector [37] and identify the nodes whose connections span both halves, or by using the modularity function from Newman and Girvan [40]. Due to the intrinsic characteristics of fraud described in section I (trade-off between risk exposure and ability to control the network), we followed the Bridging Centrality metric procedure to identify nodes serving as gatekeepers and which of those are boundary controllers.

### 4.3.2 Bridging Centrality (GateKeeper Nodes)

Nodes with high bridging centrality are very well located in the network. They have a very low degree but serve as bridges between highly connected clusters, functioning as gatekeepers between these different clusters. The bridging centrality metric,, for node, *i*, is obtained from its betweenness centrality, and bridging coefficient [86].

$$C_R(i) = B_C(i).C_B(i) \tag{4.3}$$

$$B_C(i) = \sum(s, i, t \epsilon V, s \neq i \neq t)\frac{\rho_{st}(i)}{\rho(st)} \tag{4.4}$$

$$C_B(i) = \frac{d(i)^{-1}}{\sum j \epsilon N(i) d(j)^{-1}} \tag{4.5}$$

Betweenness centrality (4.4) defines globally the node importance in terms of the ratio of shortest paths that cross it, (4.5) ; bridging coefficient, (4.3) translates into how well the node is located between high degree nodes, ultimately it gauges the relevance of its neighbourhood, N, via their degree (). The product between these two metrics assesses the characteristics of the node in the local network with its relevance in a more global scope. Bridging centrality values are arbitrary since they relate to the network topology, yet, only the top 25 percentile are relevant, since below this point the interest for these nodes rapidly drops [86].

### 4.3.3 Influence (Important Local Nodes)

Node influence metrics aim to quantify how much impact each node has within a graph. Among the commonly used approaches are accessibility measures derived from random walk theory, which evaluate the diversity of self-avoiding walks originating from a node. In fraud networks, influential actors are not always those with the highest degree or betweenness; instead, fraudulent behaviour often depends on the ability of certain nodes to reach critical areas of the network within a limited number of steps. Because this metric incorporates a parametric horizon, its values are inherently uneven: only nodes sufficiently close to major

hubs will achieve high influence scores. A key advantage of this class of metrics is their local scope. They do not require global knowledge of the entire network topology, which is valuable in settings where the network may be evolving or partially unknown. Fraudulent structures frequently introduce layers or additional distance between nodes to obscure activity, and ***influence*** provides a means to evaluate such behaviours through epidemic-style models across a wide variety of network configurations. In the simplest SIR formulation for closed and constant populations, the dynamics are described by two fundamental performance equations: the susceptible population in (4.6) and the infected population in (4.7).

$$\frac{dS}{dt} = \frac{-\beta}{N} SI + \gamma I \tag{4.6}$$

$$\frac{dI}{dt} = \frac{\beta}{N} SI - \gamma I \tag{4.7}$$

In these equations, $\beta$ is the average number of contacts made by nodes, N is the total population size and $\gamma$ is the recovery rate. The Influence Score approach considers an initial scenario of a network where there are no recovered nodes and one seed node $i$ is able to transmit to their clusters in a distance $h$ (typically, a value of 2 is considered for this spreading horizon). In this scenario the neighbours of $i$, a set with cluster sizes $d_1$, contains all first-degree nodes (i.e., those directly infected by $i$), while $d_j$ is the set of cluster sizes of all infected clusters after $j$ transmissions. As an example, if $i$ contains two neighbours, $a$ and $b$, the following clusters are formed: after one transmission: ([$i \rightarrow a$], [$i \rightarrow b$]), leading to $d_1$ = {1,1}; and after two transmissions ([$i \rightarrow a$, $i \rightarrow b$], [$i \rightarrow b$, $i \rightarrow a$]); additionally, if $a$ and $b$ share an edge, the clusters [$i \rightarrow a$, $a \rightarrow b$], [$i \rightarrow b$, $b \rightarrow a$] are also considered after two transmissions, thus $d_2$ = {2,2,1,1}. Assuming non-recovery in the nodes of the network (a fraudulent node does not recover from that state), the Influence Score of a spreading process seeded from node $i$ is a discrete random variable from $d_1$ to $d_h$, allowing for the proportionality constant equal to the transmission rate of the process. The Influence Score can be estimated (4.8) by the entropy of $d_j$ after normalization [58].

$$k(i) = \sum_{(j=1)}^{J} d_j log(d_j) \tag{4.8}$$

The ***Influence Score***, quantifies the distribution of the number of susceptible edges after two transmission events originating from a seed node in a complete susceptible network. The higher the value, the higher the diffusion effect and consequently the influence of the node in the network. Empirically it can be assumed that driver nodes with high bridging centrality have the highest ***Influence Score*** in the network.

### 4.3.4 Node Role and Influence Labels

Considering the approaches and metrics described in the previous sections, the proposed method labels nodes according to their role in the network. Nodes with control capabilities are labelled with d (driver nodes). Nodes within the top 25 percentile bridging centrality metric are labelled with t (top bridging) and whenever the node reveals both properties a dt label is assigned to the node. Nodes not revealing any of these properties are labelled as n. The Influence Score value is calculated individually to each node.If the dataset as ground proof classification of suspicious abnormal behaviour the label f is added. Consequently, the f label is assigned to nodes with reported fraud and control; fd, fraud and top bridging; ft and fraud with control and top bridging; fdt. Figure 4.2 illustrates the applicability of the proposed method.

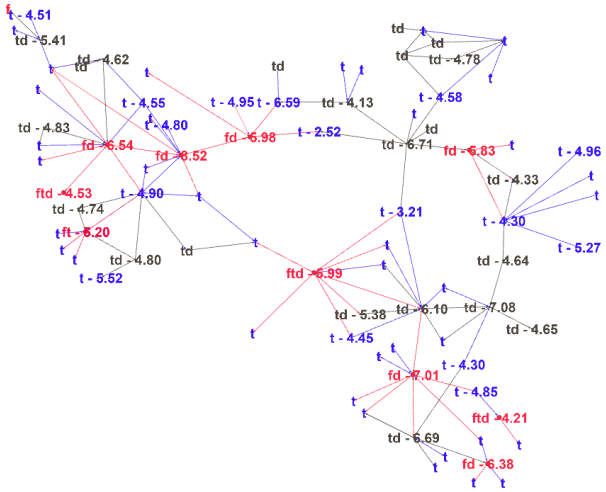



**Figure 4.2:** STARBRIDGE Role Rank.

## 4.4 Experiment

The dataset used,contains 515.755 nodes and 23.863.734 edges forming 487 connected components. This dataset corresponds to the anonymized who called who network obtained from the records of a telecommunications operator corresponding to all inbound and outbound phone calls during one month billing period. In order to preserve centrality metrics, we focus on the largest connected component in the network, a subset of 28356 nodes and 178542 edges.
In this subset Fraud was identified in 1426 nodes from running an algorithm on a Fraud Management System configured to detect telecom fraud in accordance to TM Forum GB954 [85]. Table 7.3 describes the results.

Fraud is triggered for nodes with suspicious behaviour, either by abusive usage to international premium numbers, known as (IRSF), International Revenue Share Fraud or suspicious use of a payment method (e.g. a credit card being used by multiple different subscribers or identified

**Table 4.1:** Network Node Role Distribution

| Complete node set (28356) | | | Fraud nodes by the FMS(1426) | | |
|---|---|---|---|---|---|
| **Role** | **%** | **#** | **Role** | **%** | **#** |
| n | 57.34 | 16262 | f | 13.77 | 197 |
| d | 18.37 | 5210 | fd | 50.27 | 719 |
| t | 16.81 | 4768 | ft | 10.69 | 153 |
| dt | 7.46 | 2116 | fdt | 25.24 | 361 |

as compromised). These types of fraud are known to be performed by organized groups. The influence score was applied to all nodes in the network, as well as the STARBRIDGE dimensions, complemented with the fraud labelling from the Fraud Management System. Table 1 describes these results. For each different STARBRIDGE category role, or combination of roles, the absolute number and percentage of nodes in each category is provided for the complete node set and for the nodes associated to fraud by the FMS system. It's possible to observe that over 75% of the fraud detected by the Fraud Management System was originated in nodes associated to **control** and **bridging**, highlighting the importance of these roles in fraud.

In the violin plot in figure 4.3, it is possible to compare the influence score of the nodes with and without fraud. While outliers exist on both domains, it is clear to identify a significant gap between the two in terms of influence, showing that fraud nodes have a considerable higher amount of influence than non-fraud nodes.

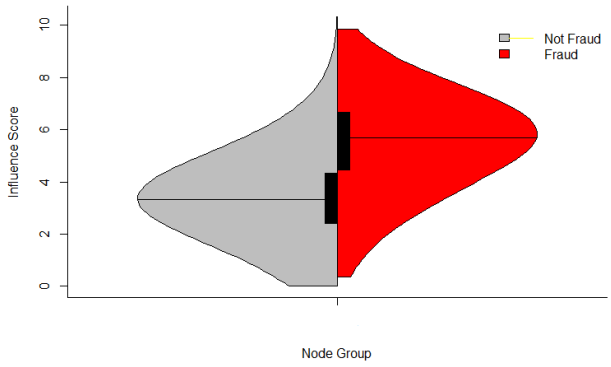



**Figure 4.3:** STARBRIDGE Influence values for Fraud/ Non Fraud Nodes

Figure 4.4, breaks down Influence by the different roles. It is possible to observe the same gap, notably between control ***(d)*** and bridging driver roles ***(dt)***.

The value span for Influence is strongly impacted by bridging centrality. This also supports a clear separation of roles, as shown in figure 4.5 that ***(d)*** and ***(dt)*** nodes lead the higher levels of Influence score in the network.

Applying the fraud label to the same set of nodes, it's possible to observe on the scatterplot of figure 4.5, that the highest majority of fraud was originated from nodes with control role ***(fd)*** followed by nodes with control and bridging roles ***(fdt)***.

Table 4.2 provides the average $\mu$ , $\eta$ .25% and $\eta$ .75% percentile values for the influence score of

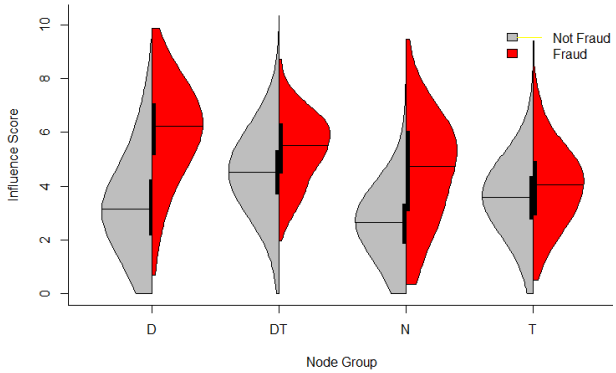



**Figure 4.4:** STARBRIDGE Influence values per role for Fraud/ Non Fraud Nodes

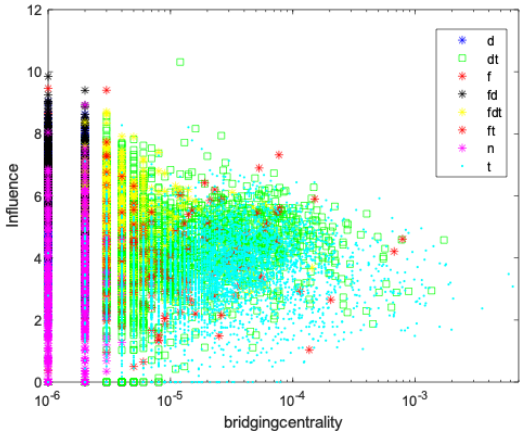


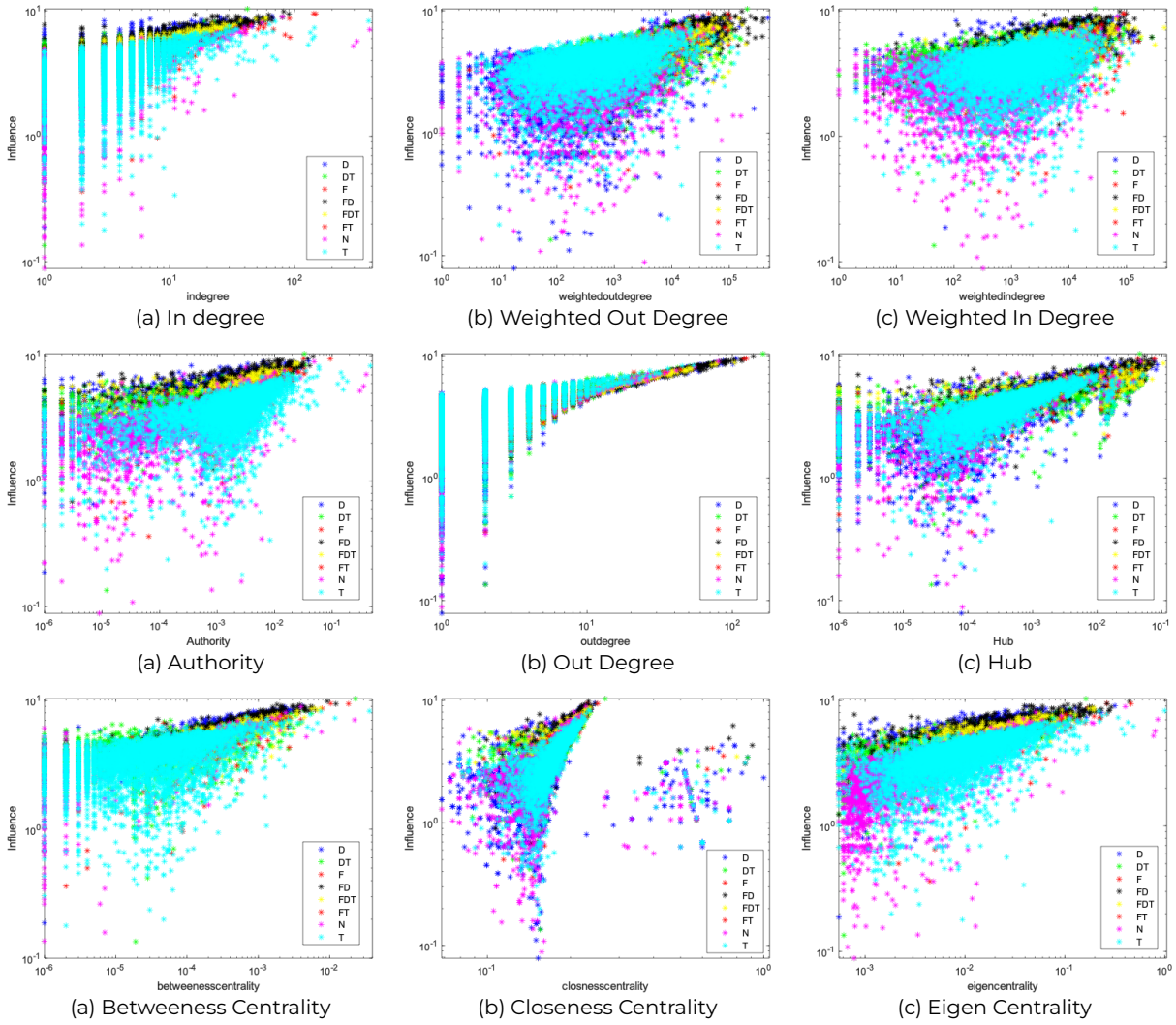

**Figure 4.5:** Influence and Bridging Centrality Correlation (LinLog)

**Table 4.2:** Node Role Influence Score

| **Node Role** | $\eta$ **.25%** | $\mu$ | $\eta$ **.75%** |
|---|---|---|---|
| **n** | 1.86 | 2.65 | 3.32 |
| **d** | 1.44 | 2.31 | 3.15 |
| **t** | 1.88 | 2.56 | 3.22 |
| **td** | 2.62 | 3.94 | 3.28 |
| **Role** | $\eta$ **.25%** | $\mu$ | $\eta$ **.75%** |
| **f** | 2.23 | 4.32 | 5.87 |
| **fd** | 3.50 | 4.45 | 5.37 |
| **ft** | 2.04 | 2.77 | 3.53 |
| **ftd** | 3.05 | 3.92 | 4.68 |

the nodes with (top) and without fraud (bottom). The difference is visible on how nodes related to fraud have a much higher Influence score in all percentiles. This is especially relevant on the category where most fraud was identified ***(fd)***, revealing how control and high influence scores are associated to fraud.

To further explore the proposed method, figure 4.6 provides the correlation of the influence score with most used centrality metrics (LogLog).




(a) In degree (b) Weighted Out Degree (c) Weighted In Degree

(a) Authority (b) Out Degree (c) Hub

(a) Betweeness Centrality (b) Closeness Centrality (c) Eigen Centrality

**Figure 4.6:** Correlation between influence and centrality metrics (LogLog)

It is possible to observe that Influence has a positive correlation with all the metrics, yet, the same is not valid for suspicious fraudulent nodes. The only positive relation of fraudulent behaviour is on the out degree and weighted out degree metrics. Exception for a borderline relation where nodes with medium to high betweenness centrality have an incidence of suspicious fraud nodes. The relationship between these metrics can be observed in figure 4.6. For visualisation, we used modularity to split the network into different communities (Figure 4.7), to visualize the role of the different nodes in the network and how they connect. Red nodes represent the suspicious nodes, ***(f,fd,ft,ftd)*** while green represents driver nodes ***(d)***,

blue the high bridging centrality nodes ***(t)*** and black the boundary controllers nodes ***(td)***. Nodes with no role in the network (n) were excluded and are not visible in the community cut.

Suspicious Fraudulent nodes have a higher out degree, forming small hubs, yet, these hubs are interconnected at a distance of 1 or 2 edges by boundary control nodes, with other suspicious fraudulent nodes, which highly increases the influence score (a distance of 2 was used). This is an observation that is consistent in all the 26 communities identified in this dataset. This is complemented by two additional observation, also prevalent in all communities that corroborate these values. First, nodes with suspicious fraud ***(fd,ftd)***,control most of the driver nodes ***(d)*** and form a path with other fraudulent nodes either directly or using gatekeepers ***(t)*** and boundary control nodes ***(td)***. Edges with highest influence ***(ft and ftd)*** define the path that connects most of the fraudulent nodes in the network, revealing the bridge and control path used by the suspicious nodes.

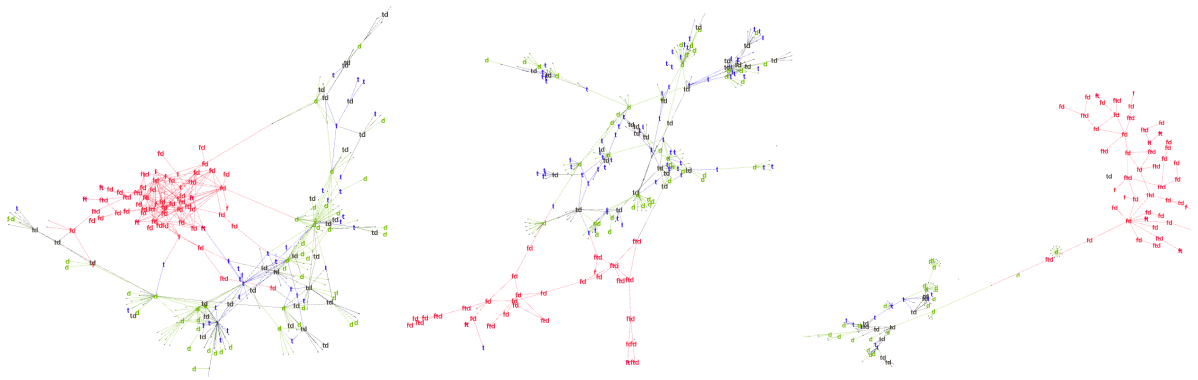


**Figure 4.7:** Nodes in Fraudulent Communities

Out of the 1430 suspicious nodes, 958 belong to a single connected component, distributed across 24 communities. These communities show strong interconnectivity, forming an almost complete clique, and access to them is only possible through specific control nodes that have already been linked to fraudulent activity.

The power of the most influential nodes can be further tested with the robustness of the network. Scale free networks are highly biased structures, in contrast with random based models.

While earlier studies [68] identify betweenness-based removal strategies as the most effective for disrupting the controllability of real-world networks, Table 4.3 presents the results of a simulated network attack that compares four approaches: Betweenness Centrality, Degree, Influence, and a Random baseline. The choice of centrality metrics in this evaluation follows the benchmarking outcomes shown in Figure 4.6, where out-degree and betweenness centrality exhibit the strongest association with suspicious fraud classifications.

While nodes with the highest degree and betweenness are the most critical ones for the network collapse until 4%, high influence nodes become the most critical ones after that

**Table 4.3:** Network Attack Benchmark (Node Removal)

| Removed% | Betweenness% | Degree% | Influence% | Random% |
|---|---|---|---|---|
| 1 | 1.53 | 4.53 | 1.53 | 1.53 |
| 2 | 3.33 | 9.83 | 3.33 | 3.33 |
| 3 | 12.99 | 12.99 | 6.61 | 7.55 |
| 4 | 12.99 | 12.99 | 6.61 | 7.55 |
| 5 | 16.10 | 25.08 | 54.01 | 10.76 |
| 6 | 22.41 | 27.96 | 57.57 | 13.9 |
| 7 | 22.41 | 27.96 | 57.57 | 13.9 |
| 8 | 25.08 | 61.52 | 70.39 | 17.01 |
| 9 | 25.08 | 61.52 | 70.39 | 17.01 |
| 10 | 59.6 | 62.37 | 71.46 | 20.02 |

point.

Using a confusion matrix (as represented in figure 4.8, it is possible to assess the classifier performance of the influence score and the ground truth from the Fraud Management System.

| | | Actual | |
|---|---|---|---|
| Predicted | | True Positive | False Positive |
| | | False Negative | True Negative |

**Figure 4.8:** Confusion Matrix

Yet, before looking to the results obtained it is important to highlight the performance reported by Telecom operators regarding Fraud detection using industry solutions.

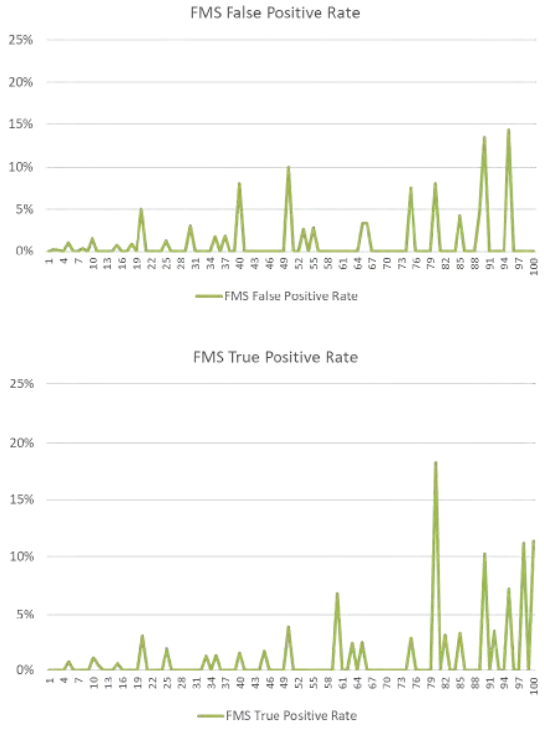



**Figure 4.9:** FMS False Positive Rate (FPR) and True Positive Rate (TPR)

More than 28% of the telecom operators report a False Positive Rate (FPR) of 90% (Figure 4.9). In particular, 14% of the operators reported 97% FPR. Concerning True Positive Rate (TPR) 62%

of the operators reported a TPR of 80%, nevertheless 7% of the operators, reported a TPR of 60%.

| | | |
|---|---|---|
| F | 46 | 78 |
| | 32 | 3148 |

| | |
|---|---|
| Accuracy: | 0,94 |
| Precision: | 0,37 |
| Sensitivity (TPR) | 0,59 |
| Specificity (TNR) | 0,98 |
| Miss Rate (FNR) | 0,41 |
| Fall Out (FPR) | 0,02 |

| | | |
|---|---|---|
| T | 116 | 1195 |
| | 29 | 4823 |

| | |
|---|---|
| Accuracy: | 0,80 |
| Precision: | 0,09 |
| Sensitivity (TPR) | 0,80 |
| Specificity (TNR) | 0,80 |
| Miss Rate (FNR) | 0,20 |
| Fall Out (FPR) | 0,20 |

| | | |
|---|---|---|
| D | 553 | 1088 |
| | 100 | 3990 |

| | |
|---|---|
| Accuracy: | 0,79 |
| Precision: | 0,34 |
| Sensitivity (TPR) | 0,85 |
| Specificity (TNR) | 0,79 |
| Miss Rate (FNR) | 0,15 |
| Fall Out (FPR) | 0,21 |

| | | |
|---|---|---|
| DT | 330 | 1560 |
| | 66 | 2579 |

| | |
|---|---|
| Accuracy: | 0,64 |
| Precision: | 0,17 |
| Sensitivity (TPR) | 0,83 |
| Specificity (TNR) | 0,62 |
| Miss Rate (FNR) | 0,17 |
| Fall Out (FPR) | 0,38 |

**Figure 4.10:** Influence Score vs FMS Classification Confusing Matrix

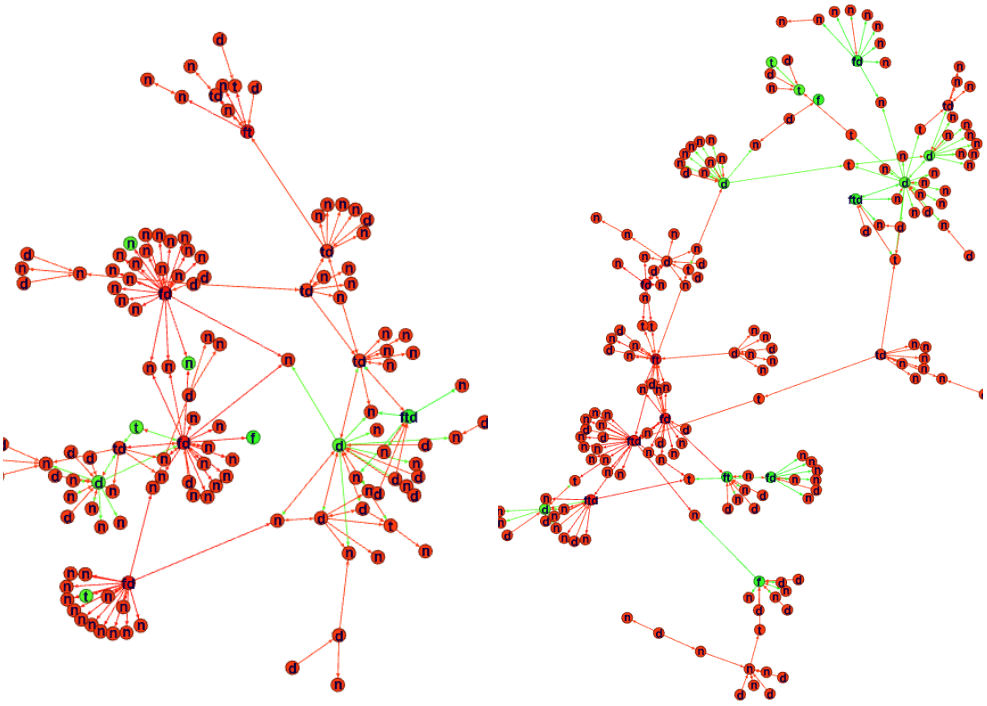


**Figure 4.11:** Modularity cuts classified with TP and TN using STARBRIDGE

While, TPR falls into an acceptable range, even though with some concerning outliers, the same is not true for the FPR. A high number of non fraudulent nodes are considered fraud, implying an untunned or oversensitive system In order to benchmark the Influence Score with the classification of the Fraud Management System, we used the percentile 75 of table 4.2 as the fraud baseline threshold , for the roles ***(n, d,t, dt)***.

On all the different node roles ***(d,t,dt)*** the TPR rate is above the 80%, in line with FMS industry reported value, yet the FPR is considerably below. While no role in the network ***(n)*** implies low influence score, the FPR was 2% implying a True Negative Rate of 98%, providing a high reliability. Despite the fact that TPR is 59% and the Miss Rate (FNR) is 41%, it is relevant as future work to identify if these nodes were in fact fraudulent and the fraud management

missed, especially in the context that TPR below 80% affects 38% of the operators.
The accuracy and precision values shown in figure 4.10 for each role corroborates the symptom that beyond a certain threshold value of influence, the node is associated to suspicious fraud behavior, hence the high accuracy, even thought that score values can oscillate in magnitude as described in table 2, therefore a low precision.
Figure 4.11 illustrates how STARBRIDGE is able to classify correctly each node in the community concerning TP (Green) and TN (Red), based on the percentile 75 of influence for each role.

## 4.5 Static and Dynamic Analysis of Fraud Networks in Telecommunication Industries

This chapter has shown that fraud networks exert a substantial impact on the revenue of telecommunications operators. Effective Fraud Management Systems must be capable of identifying all relevant actors within these structures, including both the ringleaders and their supporting members. Strategic positions within such networks grant fraudsters the ability to exercise control and influence, shaping network behaviour according to their roles, such as bridges ***(t)*** or control points ***(d)***.

While static network analysis provides valuable insights into these structures, it is equally important to acknowledge that fraud networks evolve quickly. This underscores the need for methods capable of analysing time-varying networks. In our static evaluation, we ranked nodes based on their neighbourhood characteristics and structural roles, revealing not only associations among nodes but also distinctive influence patterns. This helped identify critical fraudulent actors and supported the design of removal strategies that maximise disruption.

Furthermore, combining multiple structural attributes, such as role and influence score, proved effective in detecting nodes with high fraud potential. Validation against real classifications from a fraud management system confirmed the relevance of this metric, with nodes in the top 25 percentile of influence showing a markedly higher likelihood of fraudulent behaviour.

Complementing this analysis, a simulated network attack demonstrated that the “Influence” metric led to faster network degradation than betweenness centrality, reinforcing the importance of developing approaches suited for dynamic network environments. As fraud networks continuously change, relying solely on static analysis may be insufficient. Integrating both static and dynamic analytical methods will strengthen the ability to detect, understand, and mitigate fraud in complex and evolving network ecosystems. The subsequent chapter extends this discussion by focusing on analytical methods suited for dynamic and time-varying fraud networks.

### 4.5.1 Credit authorship contribution statement

**Paper:** Star-Bridge: a topological multidimensional subgraph analysis to detect fraudulent nodes and rings in telecom networks
**Doi:**https://doi.org/10.1109/BigData55660.2022.10020714 2022 IEEE International Conference on Big Data (Big Data)

**Pedro Fidalgo**: Investigation, Software, Writing – original draft & editing.
**Rui J. Lopes**: Investigation, Methodology, Supervision, Writing – original draft & editing.
**Christos Faloutsos**: Investigation, Methodology, Supervision.

# Chapter 5

# Dynamic Analysis of Fraud Networks in Time-Evolving Graphs

## 5.1 Introduction

In the context of the previous chapter on fraud networks and their impact on telecommunication industry, we now transition to a crucial aspect of our study ***Dynamic Analysis of Fraud in Time-Evolving Graphs***. This chapter aims to address the evolving challenges faced by analysts in detecting and understanding anomalies within dynamic networks.

As we have established the effectiveness of static analysis in identifying key fraudulent nodes and patterns within a static snapshot of a network, it's imperative to extend our approach to encompass time-evolving graphs. These dynamic graphs, exemplified by networks of phone calls, are characterized by their ever-changing nature, with new nodes and links continuously forming over time. This presents unique challenges in fraud detection, where traditional static methods may fall short.

Focusing on an unsupervised framework, we delve into scenarios where no predefined labels exist. Our challenge lies in analyzing millions of phone call records, each annotated with source and destination numbers, timestamps, and call duration. The objective is to develop methodologies that not only locate anomalous activities and identify potential fraudsters but also adapt to the network's temporal changes.

This chapter will explore innovative techniques to assist analysts in detecting, visualizing, and understanding various anomalies within large-scale, dynamic 'who-calls-whom' graphs. We aim to provide tools and strategies that not only identify current fraudulent activities but also predict and adapt to future changes in the network's structure and behavior.

In summary, this chapter sets the stage for a comprehensive exploration into dynamic analysis methods suitable for time-evolving graphs, building upon the foundation laid in static

analysis and advancing towards a more holistic understanding of fraud detection in rapidly changing network environments.
To actualize the dynamic approach in analyzing time-evolving fraud networks, we have developed a novel method named 'TGRAPHSPOT'. TGRAPHSPOT is tailored to navigate and analyze the complexities of dynamic graphs effectively, providing a robust framework for fraud detection in evolving networks. This method is structured into three distinct interconnected steps:

- Step 1: **Feature Selection**
- Step 2: **Summary**
- Step 3: **Deep Dive**

***Feature Selection***: Central to our dynamic approach is the careful selection of features from each node in the network. TGRAPHSPOT employs advanced algorithms to identify and extract the most relevant and dynamic features, allowing for an accurate representation of the network's evolving nature.

***Summary View***: The second step involves creating a high-level, interactive summary of the data. This feature empowers analysts to gain a quick yet comprehensive overview of the entire network, highlighting key areas and trends that may warrant further investigation. The summary view is designed to adapt and update in real-time, reflecting the ever-changing landscape of the network.

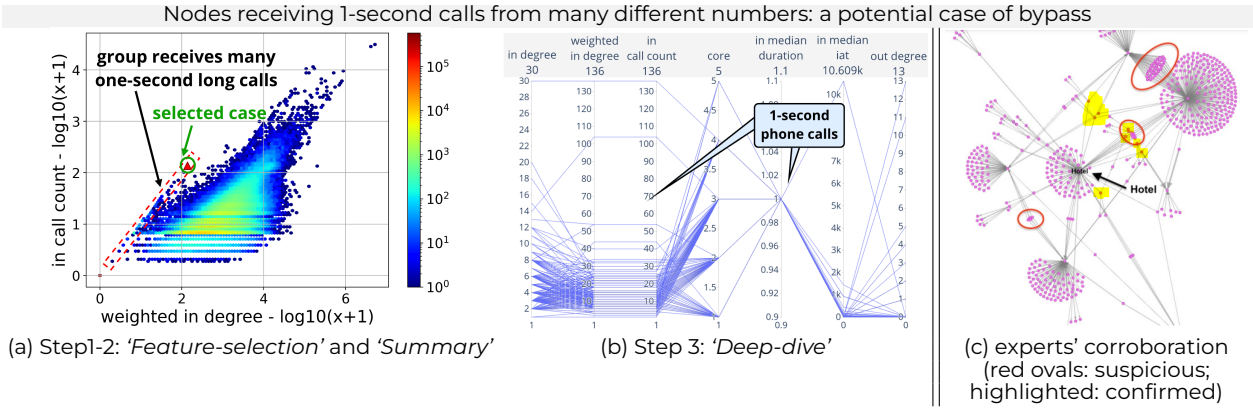
***Deep Dive Analysis***: The final step of TGRAPHSPOT allows for a focused investigation of suspicious nodes. By drilling down into specific areas of interest identified in the summary view, analysts can explore intricate details and relationships within the network. This deep dive is crucial for understanding the mechanisms of potential fraud and for devising targeted strategies to combat it.

TGRAPHSPOT stands as an advancement in dynamic network analysis, offering a powerful tool for analysts struggling with the complexities of time-evolving fraud networks. Through its innovative steps, TGRAPHSPOT not only detects current anomalies but also provides insights into the evolving patterns of fraud, paving the way for more proactive and effective fraud management strategies.

Figure 5.1 demonstrates TGRAPHSPOT effectively in use, highlighting a particular group of nodes known as '1-second in-calls'. In Figure 5.1(a), a heatmap is presented, displaying the quantity of incoming calls versus their total duration on log-log scales. Here, a distinct diagonal cluster of nodes becomes noticeable, a key finding in the *''Summary'* phase. Figure 5.1(b), we see the *'Deep-dive'* phase in action: it features a parallel axes plot depicting the ego-network of a chosen node (marked as a 'red triangle' in Figure 5.1(a). A striking observation here is the abnormal pattern in this egonetwork, where the majority of calls are incoming and last only about 1 second, a pronounced deviation from typical call patterns. The outcome of the analysis is shown in Figure 5.1(c) shows the result of the investigation: using a spring-model visualization of the same ego-network, further analysis determined that the central node is a hotel's phone number, predominantly receiving short international calls. The brief duration of these calls is attributed to the involvement of dubious telecom operators offering illegally low rates and poor quality, resulting in calls being disconnected almost immediately upon connection. While the hotel itself was not involved in fraud, the pattern of calls to this number (and others along the 45-degree line in Figure 5.1(a)) aligns with the characteristics of a well-known fraud scheme termed 'International Bypass'. In Figure 5.1(c), the suspicious callers are enclosed within red ovals, and the verified fraudsters are marked with a yellow highlight. Notably, these particular callers were previously undetected, underscoring the potency and utility of our TGRAPHSPOT method in uncovering and elucidating complex fraudulent activities.

Our newly developed TGRAPHSPOT presents several key benefits:

- **Effectiveness** As illustrated in Figure 5.1 and further discussed in section 5.3, TGRAPHSPOT effectively aids professionals in identifying subscribers or nodes exhibiting unusual activities that were previously unnoticed.
- **Interactivity:** TGRAPHSPOT not only provides a rapid overview of the dataset, but also enables experts to engage interactively, focusing on specific data subsets as part of the 'Deep-dive' phase.
- **Explainability:** We have selected all visual representations with an emphasis on clarity and interpretability, as demonstrated in Figure 5.1.
- **Scalability:** In the 'Feature-selection' phase, we have intentionally chosen features that




(a) Step1-2: *'Feature-selection'* and *'Summary'*

(b) Step 3: *'Deep-dive'*

(c) experts' corroboration (red ovals: suspicious; highlighted: confirmed)

**Figure 5.1:** TGRAPHSPOT at work:
(a) Several nodes appear aligned along the 45-degree reference line (outlined with a red dashed box), clearly separated from the main cluster. Both the axes and the colour scale use logarithmic scaling. (b) *'Deep-dive'* on the red triangle: a parallel-axis representation of the EgoNet associated with the 'red triangle', indicating that these nodes receive phone calls of one second in duration. (c) Upon inspection of the nodes enclosed by the red ovals, domain experts verified that the *callers* shown in yellow highlight exhibit characteristics consistent with 'International Bypass' fraud — see text for details.

ensure speed and efficiency, scaling linearly with the size of the input data.

**Reproducibility:** We have made our code publicly available. Due to customer privacy concerns, accessing the datasets requires a Non-Disclosure Agreement (NDA).

Figure 5.2(A) displays TGRAPHSPOT along with its various modules, organized as follows:

- **Module (i):** focuses on the extraction of features.
- **Module (ii):** encompasses both static and interactive visualizations, including:
  - (a) heatmaps
  - (b) scatter matrices of the chosen features.
- **Module (iii):** enhances deep dive functionalities, offering:
  - (a) detailed analysis for individual nodes, such as cumulative in/out degree, call counts, and cumulative in/out call durations per hour
  - (b) for a node group, tools like adjacency matrices, parallel coordinates, and graph spring models.
- **Module (iv):** titled Attention Routing, is designed to spotlight outliers and micro-clusters in a hierarchy of significance.

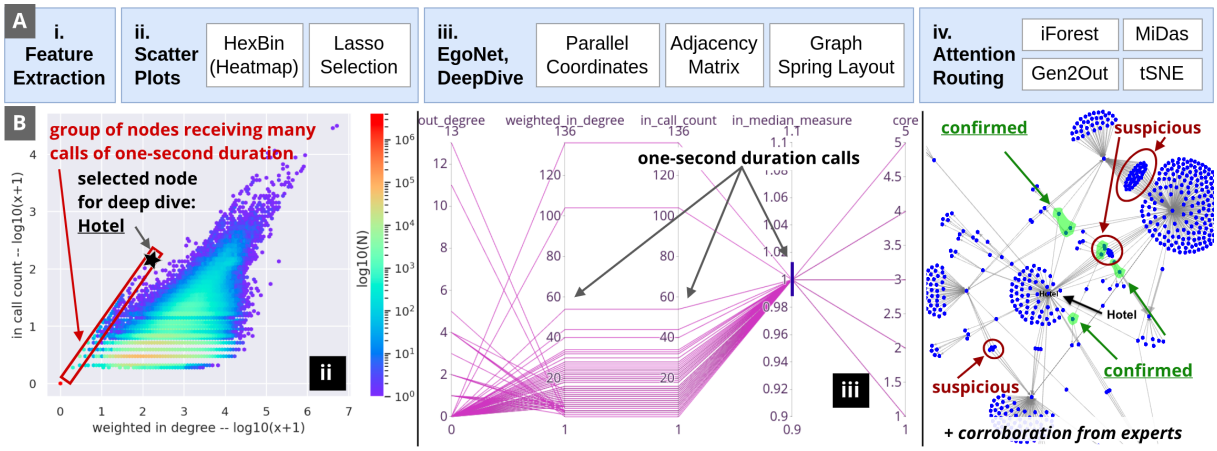



**Figure 5.2:** TGRAPHSPOT in action, spotting fraudsters:
The pipeline in (A) illustrates the sequence of steps executed by TGRAPHSPOT, while (B) displays the corresponding visualisations. After (i) feature extraction, TGRAPHSPOT offers tools for visualising (ii) combinations of features through scatter plots. The Lasso Selection enables users to choose a subset of nodes for deeper inspection. TGRAPHSPOT then constructs (iii) an EgoNet for the selected nodes, presenting the adjacency matrix, parallel coordinates, and the EgoNet rendered with the Spring Layout. Finally, the (iv) attention routing module provides mechanisms for identifying fraud within the dataset.

## 5.2 The Proposed Method: TGRAPHSPOT

As mentioned, our TGRAPHSPOT comprises three steps: *'Feature-selection'*, *'Summary'*, and *'Deep-dive'*. Algorithm 1 provides the pseudocode together with the design decisions for each of the steps.

**Algorithm 1:** TGRAPHSPOT Outline

**Data:** Phone calls with source, destination, duration
**Result:** Static and interactive plots
Build a time-evolving graph $G$ ;
`/* Step 1: Feature Extraction */`
Extract static and temporal features ;
`/* Step 2: Summary and Visualization */`
**User action:** Select features ;
– Print heatmap plots of combined features ;
– Print scatter feature matrix ;
– Print interactive 2-D pair plots ;
`/* Step 3: Deep Dive */`
**User action:** Select multiple nodes of interest ;
– Print adjacency with cross-associations ;
– Print parallel coordinates ;
**User action:** Select a single node for a deep dive ;
– Print cumulative in/out degree and calls per hour ;
– Print cumulative in/out call duration per hour ;

### 5.2.1 Step 1: Proposed Features

We decided to focus on node-based features, thereby transforming the problem into detecting anomalies in an $n$-dimensional point cloud.

**Static Features** For a static graph, we use the degree (number of distinct connections), call-count (total number of calls), and duration (total call minutes), computed for both in- and

out-going calls. We also calculate the coreness of each node, reflecting their connectivity level.

**Time-Evolving Features** We propose the median inter-arrival time for both incoming and outgoing calls, as well as the median call duration, favouring medians over averages to ensure greater robustness to outliers. We also derived the 25% and 75% quantiles for these metrics, although these percentile values did not lead to significant discoveries.

### 5.2.2 Step 2: *'Summary'*- Proposed Visualizations

To summarize each node as a point in a medium-dimensional space ($d \approx 30$), we considered various visualization techniques.

**Proposed Plots** TGRAPHSPOT offers 1-d histograms, 2-d scatter-plots, Adjacency Matrix and heatmaps , 2-d pair-plots , Original and Cross association matrix and parallel coordinates (Figure 5.3,5.4,5.5,5.6),interactive plots, allowing analysts to focus on points of interest.

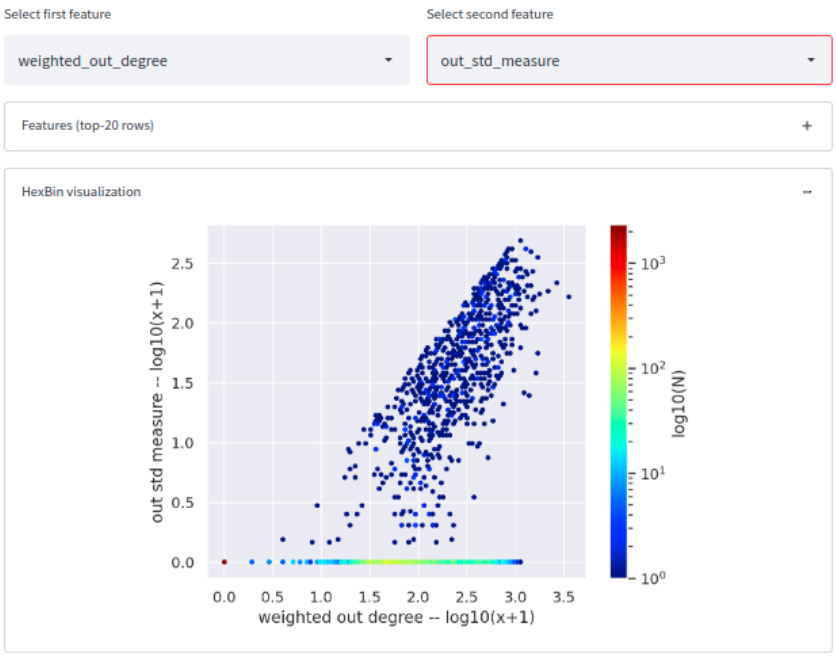



**Figure 5.3:** Screenshot of TGRAPHSPOT: Heatmap plot of selected features.

**Axis Scaling** We use $\log(x+1)$ scaling for axis and color-maps in heatmaps due to the expected power-law distributions and the presence of possible zeros.


**Figure 5.4:** TGRAPHSPOT - Deep-dive: matrix of scatter plots and *lasso* selection by the user.

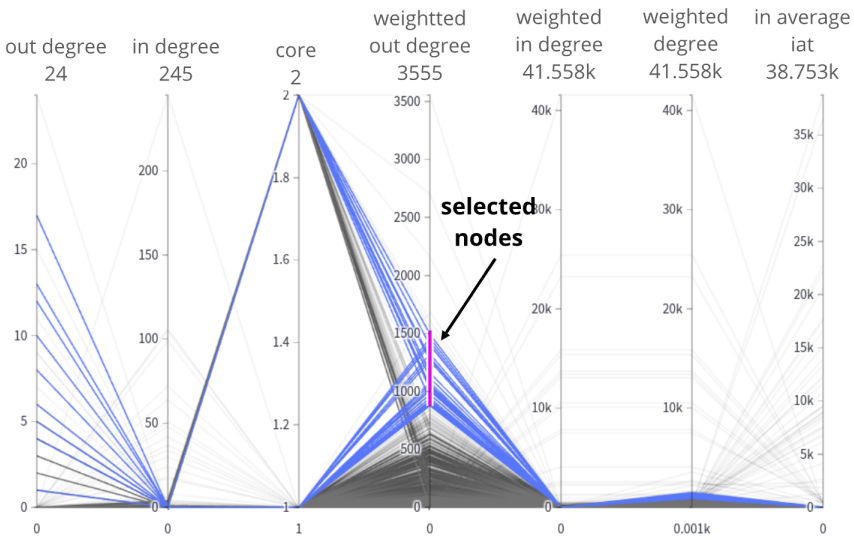



**Figure 5.5:** Parallel coordinates of features from nodes in the generated EgoNet.

### 5.2.3 Step 3: *'Deep-dive'*- Proposed Interactions

For in-depth analysis, TGRAPHSPOT can display:

- (a) the adjacency matrix of the *n*-step EgoNet of selected nodes (with row and column reordering);
- (b) the parallel-coordinates plot for the EgoNet;
- (c) the time evolution (number of calls per hour over time - see

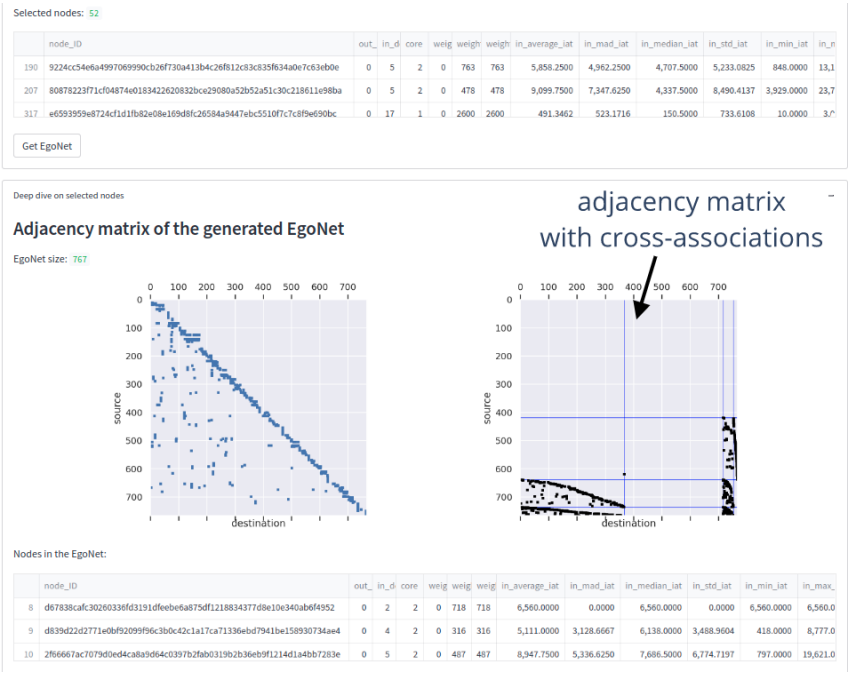



**Figure 5.6:** Adjacency matrix (original and cross-associations)

Furthermore, TGRAPHSPOT incorporates a negative-list feature specifically designed to filter out nodes such as customer support lines, toll-free numbers, national emergency and similar services. These nodes, exhibit high centrality and significantly reduce the network's diameter due to their frequent call reception, predominantly they generate noise that obscures meaningful data analysis. While their routine and widespread usage contributes little to the detection of fraudulent activities, their exclusion needs to be evaluated and analysed. Later we will notice how these numbers can be used to camouflage and hide abnormal behavior by the fraudsters.

## 5.3 Experiments

### 5.3.1 Effectiveness

Figure 5.7 reveals a suspicious group of nodes, termed '10-second in-calls' that TGRAPHSPOT helped discover. In the ('Summary') step, a heatmap (Figure 5.7a) contrasting median call duration with in-call count displays an anomaly: a few points forming a horizontal line (red dashed box), distinct from the majority that shapes a triangle. All axes, including the colormap, are logarithmic.

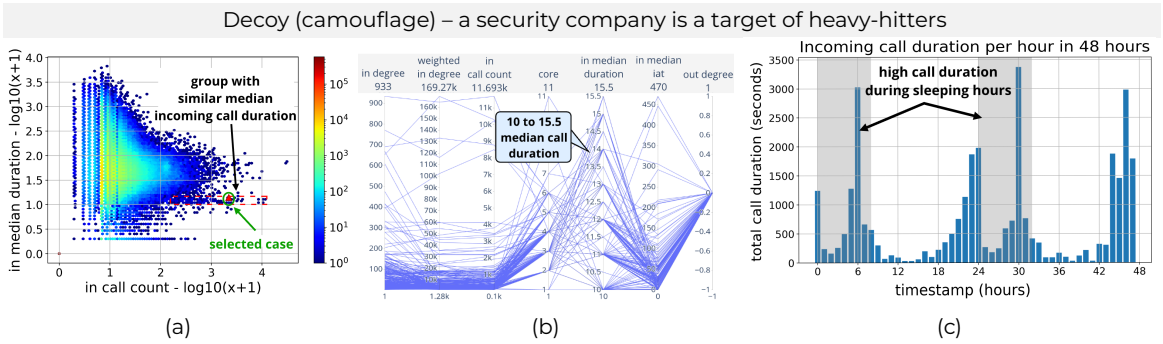



**Figure 5.7:**
TGRAPHSPOT spots unusual behaviours that are later confirmed as fraud victims: (a) several nodes show a very similar median duration for incoming calls; (b) a cluster of nodes predominantly receives calls lasting between 10 and 15 seconds; (c) the node highlighted in (a) receives a high volume of incoming calls during typical sleeping hours, with clear spikes at 6 am on consecutive days.

Further investigation in the 'Deep-dive' step (Figure 5.7b) through a parallel axis plot of these nodes shows a narrow median call duration of 10-15 seconds and no outgoing calls. Additionally, the time-plot (Figure 5.7c) indicates most calls were received during typical sleeping hours, with a spike at 6 am. These 'red flags' led to a closer examination of the anonymized call records, revealing:

- Call durations predominantly in three specific lengths: 10, 13, and 30 seconds.
- Uniform ringing times, suggesting an automated answering mechanism.
- The caller nodes are mostly outbound 'stars' with no inbound calls and no participation in forming triangles.

Using the de-anonymized dataset, analysis suggested these numbers belong to legitimate businesses utilizing Interactive Voice Response (IVR) systems, which through dual tone multi-frequency' (DTMF) codes allow navigation on the IVR menus forcing a specific path of options until time out and hangup.

The pattern of calls, especially at unusual hours and of brief duration's, indicated these destination numbers (victims) might be unintentionally involved in schemes to disguise illegitimate activities, such as camouflage suspicious traffic with normal domestic calls, since most fraud systems use rules based in ratios to identify abnormal behavior. Additionally, the high volume of calls to these numbers was identified as symptoms of Telephonic Denial of Service (TDoS) attacks, (supported by release codes that signal capacity-related errors).

This case underscores the importance of careful consideration in populating the 'negative-list.' While it's crucial to filter out nodes that contribute noise and distort network analysis, attention must be paid to ensure that this filtering doesn't overlook sophisticated fraud schemes.

### 5.3.2 Scalability

Figure 5.8 shows how TGRAPHSPOT scales linearly from tiny datasets (100 phone calls, 122 nodes) to huge ones (29M phone calls, 7M nodes), taking 2.5 hours for 29 million phone calls, on a i7-8565U CPU@1.80GHzx8,16GB machine.This linear scaling characteristic is particularly noteworthy because graph-based algorithms often suffer from exponential or polynomial time complexity. The ability to maintain linear performance even as the dataset grows to millions of nodes suggests that TGRAPHSPOT design effectively manages computational resources and avoids common scalability bottlenecks in graph processing. This makes it practical for deployment in real-world telecommunications environments where processing large volumes of call data is a daily requirement.

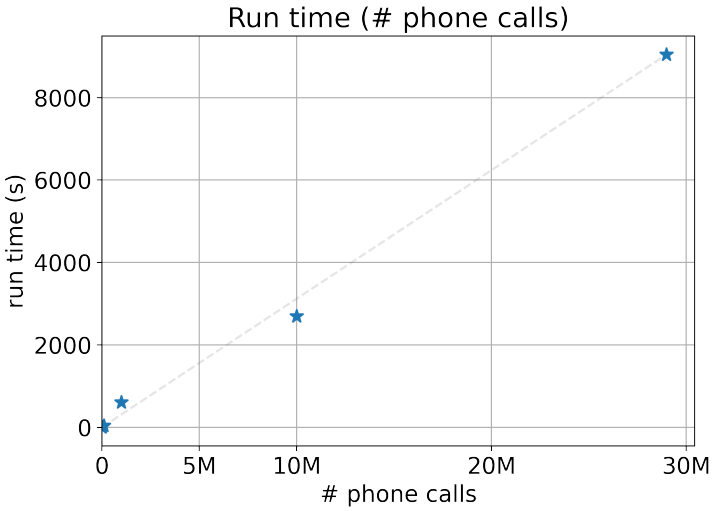



**Figure 5.8:** TGRAPHSPOT scales linearly

### 5.3.3 Credit authorship contribution statement

**Paper:** TgraphSpot: Fast and Effective Anomaly Detection for Time-Evolving Graphs
**Doi:**https://doi.org/10.1109/BigData55660.2022.10020898
2022 IEEE International Conference on Big Data (Big Data)

**Paper:** TgrApp: Anomaly Detection and Visualization of Large-Scale Call Graphs
**Doi:**https://doi.org/10.1609/aaai.v37i13.27062
Proceedings of the 37th AAAI Conference on Artificial Intelligence - Vol. 37 No. 13: AAAI-23 Special Programs, IAAI-23, EAAI-23

**Pedro Fidalgo**: Investigation, Methodology ,Software, Writing – original draft & editing.
**Mirela T. Cazzolato**: Investigation, Methodology ,Software, Writing – original draft & editing.
**Saranya Vijayakumar**: Investigation, Writing – original draft & editing.

**Xinyi Zheng**: Investigation, Software

**Namyong Park**: Investigation, Software

**Meng-Chieh Lee**: Investigation, Software

**Bruno Lages**: Investigation , Software

**Agma J. M. Traina**: Investigation, Methodology, Supervision.

**Christos Faloutsos**: Investigation, Methodology, Supervision, Writing – original draft & editing.

# Chapter 6

# Mining Billion-Scale Call Graphs for Fraud Detection and Visualization

The use of TGRAPHSPOT has unveiled a range of behaviors that were previously unknown and challenging to justify. A key issue encountered was the explainability of these anomalies, particularly when they manifested as organized behaviors among groups of nodes within the network. The importance of visualization in explaining these anomalies cannot be overstressed, as it plays a critical role in identify the complex patterns and aiding in the interpretation of such organized fraudulent activities.

This need for enhanced explainability and effective visualization becomes even more pressing in light of a report by CFCA [26], which highlights the overwhelming False Positive Rate (FPR) of 90% experienced by 28% of telecom operators. This alarmingly high FPR has led to analysts dedicating a significant amount of time to investigating false leads, causing delays in detecting and mitigating actual fraud cases and leading to considerable financial consequences. In a domain plagued with such high FPRs, the operational burden of unproductive investigations is a major concern.

Adding to the complexity is the detection of unknown types of fraud. These emerging patterns, which evade pre-existing rules and detection systems, are elusive for traditional, rule-based fraud management systems. Even machine learning models, trained on known fraud patterns, often fall short in identifying these new types of fraud due to the lack of appropriate training data. Supervised learning, by its nature, relies heavily on the quality and accuracy of labeled data, however, this becomes particularly challenging in the dynamic and complex world of telecommunications fraud, where labels are often imperfect or incomplete. This issue raises a critical question: how can we effectively detect and understand fraud when our guiding signals may be flawed or misleading? Understanding that fraudsters often employ sophisticated techniques to avoid detection, our enhanced focus in supervised learning is now

directed towards accurately identifying nodes similar to those already recognized, despite the potential camouflage and low and slow techniques used by fraudsters. This refinement in our approach is crucial for maintaining the effectiveness of our fraud detection strategies, especially in situations where deceptive practices are employed to evade traditional detection methods.
To meet this challenge we introduce CallMine, a natural evolution of TGRAPHSPOT, specifically designed to thrive in the dynamic landscape of time-evolving graphs, even when faced with imperfect labels. CallMine robustness in supervised settings, where data labels may be inaccurate or incomplete, sets it apart from traditional methods that falter under such conditions.

## 6.1 Introduction

Phone calls are a ubiquitous method of communication. However, they are often used for fraudulent purposes and monetary gain.

*With datasets comprising millions to billions of who-calls-whom records,what strategies can be employed to identify abusive or fraudulent communications?*

*What methodologies can be developed to assist analysts in elucidating the anomalies and effectively visualizing these patterns within dynamic, time-evolving graph structures?*

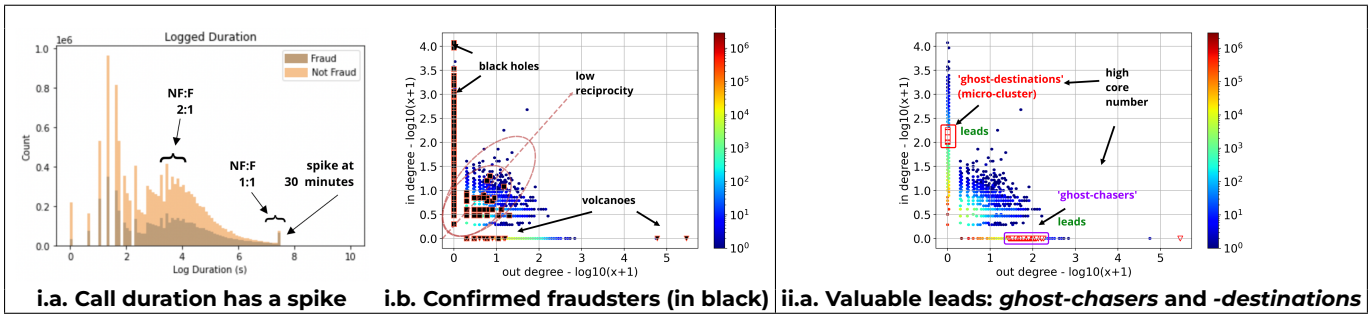
Our goal can be summarized:

**Problem 1** (Anomaly detection and visualization)**.**

- **Given**
  - *> Who calls whom, when, and for how long*
  - *> Fraud/non-fraud labels for some of the nodes (optional)*
- **Find**
  - *> Fraudsters similar to the ones already labeled*
  - *> New types of fraudsters*
  - *> Explanations of fraud/non-fraud labels*
  - *> Visualization and interaction*

### 6.1.1 CALLMINE Discoveries

Figure 6.1 illustrates some of CALLMINE's discoveries, as described above.




**i.a. Call duration has a spike** **i.b. Confirmed fraudsters (in black)** **ii.a. Valuable leads: *ghost-chasers* and *-destinations***

**Figure 6.1:**
CALLMINE works for the supervised (i.a-b) and unsupervised settings (ii.a). i.a: 1-d histogram of duration - fraudsters tend to do long phone calls. i.b: 2-d heat-map of in- vs out-degree - confirmed fraudsters either have zero in-degree ('volcanoes', black triangles) or zero out-degree ('black holes', in black squares). ii.a unsupervised case: CALLMINE discovers two suspicious groups forming a near-bipartite core (ii.a): *ghost-chasers* (purple box) and *ghost-destinations* (red box). See text for more details.

*Sudden Cut-off.* Figure 6.1 (i.a) shows the probability density function (PDF) of the duration of phone calls, revealing an sudden cut-off at precisely at 30 minutes. We describe this phenomenon as *Sudden Cut-off* and provide further details in Observation 1

*No Reciprocity.* Figure 6.1 (i.b) illustrates a scatter-plot of customers, plotting in degree against out degree on 'triple-log' scales (even the color map is logarithmic). A significant number of customers fall into two distinct categories: 'volcanoes' (high out degree; very low in degree) or 'black holes' (high in degree; low out degree). Black points indicate confirmed (labeled) fraudsters (squares for black holes and triangles for volcanoes). Notably, there's an unusually small number of points along the diagonal, indicating a lack of reciprocity. While some level of reciprocity is typical in phone call networks, it is missing in this case.

*Useful Leads.* Figure 6.1 (ii.a.) demonstrates how CALLMINE yielded insightful leads, highlighted in a purple box and referred to as Ghost Chasers in the figure. This plot bears resemblance to Figure 6.1 (i.b), but with two key differences: (1) the labeled nodes are omitted, and (2) it instead highlights two groups of nodes identified by CALLMINE as suspicious: the Ghost Destinations, which are non-operational phone numbers (thus labeled 'ghosts'); and the Ghost Chasers. Intriguingly, CALLMINE discovered that all the Ghost Chaserswere making calls to most of the Ghost Destinations. Even more notable is that none of the Ghost Chasersare labeled as 'fraud', yet their behavior is distinctly abnormal, mirroring that of confirmed fraudsters (volcanoes with lower degree, represented as black triangles in Figure 6.1(i.b)).

Section 6.1.11 delves deeper into how CallMine helped in identifying these two groups.

### 6.1.2 Properties

CallMine exhibits the following properties:

– **Scalable.** CallMine is built to operate with linear complexity in relation to the size of the dataset during feature extraction. To maintain this efficiency, computationally costly tasks such as triangle enumeration and shortest–path calculation are deliberately omitted.

– **Effective.** CallMine integrates practical knowledge about calling behaviour, recurring graphical patterns, and the characteristics of call–graphs. Its design emphasises usability, enabling analysts without technical backgrounds to interpret the results confidently.

- *Interactive* – supports detailed inspection of suspicious nodes and accommodates large datasets (7M nodes; 35M edges).
- *Flexible* – is suitable for both labelled *and* unlabelled datasets.
- *automatic* – operates without requiring manual parameter configuration.
- *explainable* – generates visualisations that aid interpretation.

– **Novel Discoveries.** The use of CallMine revealed several noteworthy patterns in the data, including the Sudden Cut-offshown in Figure 6.1(i.a), as well as the Ghost Destinationsand Ghost Chasersin Figure 6.1(ii.a), previously discussed under Useful Leads. Further findings are summarised in Observations 1–5.

To the best of our knowledge, CallMine is the first approach for fraud detection in call–graphs that brings together all of the properties outlined above. CallMine relies solely on source, destination, duration, and timestamp information, without requiring supplementary metadata or subscriber attributes. This work also documents categories of fraudulent activity that are not commonly studied in the academic literature.

In addition, we describe the feature design and explain how it is combined with domain knowledge to enable precise identification of fraudulent behaviour.

### 6.1.3 CallMine in the real world

CallMine identified suspicious behaviour within a matter of hours. These cases were only acknowledged as confirmed fraud by human analysts around ten months later, when affected

customers eventually reported the issue to their operator. Figure 6.2 illustrates that this represents an improvement of roughly *7,000 times* in detection speed.

Had CALLMINE been deployed operationally, it would have prevented activity from approximately 2,000 nodes, which together generated nearly 100K calls per day over a period of ten consecutive months, which amounts to roughly 10M fraudulent calls in total.

CALLMINE supports both supervised and unsupervised operations. The unsupervised mode is particularly valuable, as it enables CALLMINE to identify entirely new and previously *unknown* forms of fraud. Through its interactive capabilities, it substantially accelerates investigative work, achieving speed-ups of up to 7,000 times, as noted earlier; and through its visual outputs, it offers clear explanations for the decisions it highlights

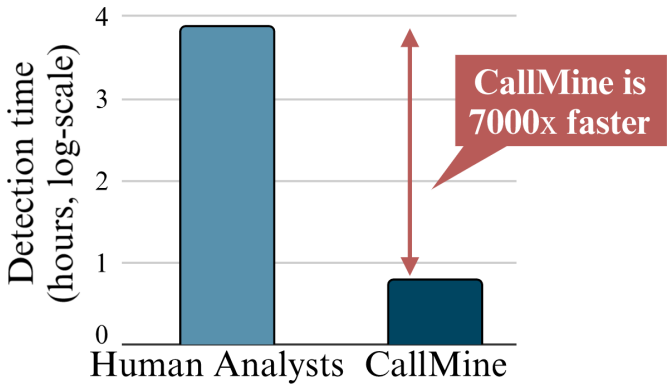



**Figure 6.2:**
**CALLMINE outperforms human analysts** and spots abnormal behavior among millions of subscribers.

### 6.1.4 Suitability for Production

– **Deployment:** CALLMINE is presently undergoing a three–month pilot phase that began in May 2023. It is being evaluated within a research laboratory environment and is scheduled to operate on 5G edge–node data, processing tens of millions of call records per day for a mobile network operator.

– **Applications:** our framework models the daily activity of millions of calls originating from both legitimate subscribers and individuals involved in fraudulent behaviour.

– **Analytics and machine learning:** we derive informative features from call records and identify fraudsters through a combination of visual inspection and automated detection. We additionally highlight the most revealing plots to support analysts in interpreting unusual patterns.

– **Data presentation:** we present the discovered patterns through intuitive visualisations aimed at large–scale anomaly detection in call–graph data. The visual tools also allow users to explore the patterns interactively and perform detailed examinations of specific nodes and their ego–networks.

The most important phase in the process is feature extraction: *which aspects of customer behaviour can signal abnormal patterns or fraudulent activity?* To address this, we begin by outlining what domain specialists typically understand. We first present key facts about phone–call datasets and the terminology involved; afterwards, we summarise known types of fraudulent behaviour and highlight the features that allow us to detect them.

***Volume and 'power-laws'.*** Phone call datasets comprise millions of nodes (customers/subscribers) and hundreds of millions of newly created edges each day; degree distributions are heavy-tailed (power-law), with the majority of customers placing very few phone calls, while a very small fraction of customers generate a large number of calls on a daily basis. Extreme behaviour (such as a high volume of international calls, very large in- or out-degrees, etc.), is frequently, though not always, an indicator of fraud: for instance, a high number of incoming and outgoing calls may originate from a large institution with a 'Private Bank Exchange' (PBX). As a result, it is necessary to consider combinations of multiple, carefully constructed features.

***Adversarial nature and 'camouflage'.*** Fraudsters try to camouflage themselves with multiple techniques, such as Human Behavior Simulation (HBS) or statistical and profiling methods. Thus, more than 42% of operators report a FPR higher than 90% [25].

***Multiple types of fraudulent behavior.*** There is a large, and growing, number of *types* of fraud: [26] gives a list of the most well known of them; we summarize them in Table 6.1, where we also list the features (in-/out-degree, etc.) that could help us detect them.

***'Fraud Types' and 'Fraud Methods'.*** Following the literature, we introduce the two concepts.

*Fraud Type* is the way that a fraudulent actor monetizes. For example, by sending calls to a premium number that he/she owns, the fraudster will charge the victims a high price.

*Fraud Method* is an enabling mechanism: For instance, through the execution of a 'Wangiri' (One Ring) attack, using a premium-rate number under his control, the fraudster places calls to a large number of different subscribers and terminates the call before it can be answered. Subscribers who return the call are then charged a premium call fee.

### 6.1.5 Fraud Types – Modus Operandi (M.O.)

Below, we list some of the most prevalent fraud types, describing the features needed to detect each type.

***International Revenue Share Fraud (IRSF)*** Fraudsters often gain access to an operator's network and direct many calls into high-cost 'revenue share' service numbers. Fraudsters

**Table 6.1:** Indicator signs for some of the fraudulent behaviors.

| | out degree | in degree | *in-weighted-deg.* | *out-weighted-deg.* | in-call count | out-call count | **IAT** | Core number informal |
|---|---|---|---|---|---|---|---|---|
| **Fraud Type** | | | | | | | | |
| Revenue Share (IRSF) | + | – | – | + | | + | | |
| Arbitrage | + | – | – | + | | + | – | + |
| Voice Interconnect Bypass | + | | – | | – | + | – | + |
| **Fraud Method** | | | | | | | | |
| CLI Spoofing | + | | | + | | | | – |
| Wangiri | + | | | + | | + | | |
| Robocalling | + | – | | + | | | | + |

achieve that through multiple fraud methods, like Wangiri (see below), PBX Hacking, etc.
Signature:High out-degree; near-zero in-degree.

***Arbitrage (MTR)*** A shady telecommunications company routes international long-distance calls through a third country to achieve lower settlement rates.
Signature:Huge out-degree; small in-degree; small inter-arrival times (IAT) (to handle the volume).

***Voice Interconnect Bypass (VOIP/SIMbox)*** Specifically for the SIMbox scenario, fraudsters partner with international entities that route international calls through local subscriber identity module (SIM) cards installed in SIMboxes, avoiding international termination fees and paying a much cheaper local termination cost. The difference between the two rates is the fraudsters' profit. Whenever taxes are applied by local governments on terminating international calls, they are also victims of this fraud type, since taxes will not be charged.
Signature:Similar to 'arbitrage' (high out-degree; small IAT). Low in-degree, but often above zero (attempting camouflage).

### 6.1.6 Fraud Method – Enabling Technique

***Caller ID Spoofing*** Fraudsters will often change their numbers to something with a similar area code so that the receiver is more likely to pick up the phone call.

Signature:Similar to Arbitrage and Voice Bypass.

***Wangiri*** This is call-back scam. Fraudsters call victims and immediately hang up; some of the victims call back, their call is re-routed to a premium number that the fraudsters own; this will incur a premium fee for the victims and their Telecom Operator.

Signature:High out-degree; zero call duration; regular **IAT**.

***Robocalling*** Robocalling refers to automated phone calls that deliver pre-recorded audio messages without human interaction. These calls are often initiated by software systems that can place thousands of calls per minute. While some robocalls are legitimate such as appointment reminders, public service announcements, or political surveys, **illicit robocalling** is commonly associated with fraud. Typical scams include impersonations of tax authorities, bank security teams, or government agencies, attempting to extract personal information, payment, or access credentials from the recipient.
Signature: High out-degree; no in-degree; too regular inter-arrival times. Short/zero duration of calls.

Here, we present CallMine and detail our design decisions. The main challenges are (a) the volume of data and (b) the adversarial nature of fraudulent actors ('camouflage'). CallMine addresses these issues by (a) designing scalable algorithms and visualizations with care, and (b) summarizing data without rigid thresholds to allow analysts to identify evolving/emerging types of fraud.

### 6.1.7 Features

We use node-level features: if a given node is a fraudster, we want to capture its behavior, and spot patterns and deviations from the behavior of a non-fraudulent subscriber. Our proposed features are in two groups: the *static* case, without timestamps and aggregating all the phone calls of a subscriber throughout the whole duration of observation, and the *dynamic* case, with the temporal information.

#### Static Case

There are countless features we can extract for each node ,PageRank, radius, several betweenness measures, clustering coefficient, linear embeddings (PCA/SVD/Laplacian),non-linear embeddings (GNNs), to name a few. Which ones, and how many should we use?

We propose a *small*, *carefully chosen* set of features, that are fast to compute. Our guiding principles are the following: (a) *scalability*: the features should be fast to compute (thus, radious is out); (b) avoiding *curse of dimesionality*: too many features may confuse the classification and clustering algorithms; (c) *explainability*: this is a must, according to domain experts - thus, black-box methods are out.

One would be tempted to extract all possible features, and let a classifier decide which ones are useful, for the following reasons:

- *Scalability*: some features, like the radius of a node, are expensive to compute, and thus prohibitive for the million-scale graphs.
- *Curse of dimensionality*: in high dimensionalities, nearest-neighbour methods become slow and thus delay several clustering methods; classifiers may discover spurious correlations; visualization methods suffer.
- *Explainability*: Black-box methods, such as common deep learning algorithms are often unexplainable, which does not suit the needs of a fraud detection algorithm.

Notice that all the features we use are:

- fast to compute (linear on the number of phone calls) we specifically stayed away from the number of triangles, radii, shortest paths.
- explainable, all of them linked to some Fraud Type, Fraud Method, or to an observation from earlier work (like reciprocity [10]) Thus, we stayed away from GCNs and deep-learning embeddings.

We aimed for a *small* set of node-level features, that are *fast* to compute, and are known to be *related* to fraudulent activity (either from Section 6.1, Table 6.1) or from earlier works on the lockstep behavior of fraudsters, log-logistic behavior of typical users, etc.

Thus, we propose the following features:

- in degree, out degree, **in-weighted-degree**, **out-weighted-degree**, in-call count, out-call count: These features help spot high/low activity and lack of reciprocity.
- **core-number**: Used to detect lockstep behavior, where groups of people have the same contacts.

The *core number* of a node is $k$, if the node belongs to the $k$-core, but not the $k+1$-core of the graph. A high core value for a node means that the node is well connected (e.g., part of a near-clique or a near-bipartite core).

#### Dynamic/time-evolving case

Inter-arrival times ('**IAT**') of events often reveal fraudsters: for example, telemarketers will call a new number every few minutes, with small variance.

**No averages or standard deviation** Both measures suffer from subtle issues: the average is effectively the 1/(out-call count), carrying no extra information; the standard deviation is huge

and thus also uninformative, since we usually have heavy-tailed distributions (like power-laws, Pareto, or log-logistic). Therefore, we exclude both features from our analysis intentionally.

Instead, we propose robust features: median rather than the mean, and MAD (Median Absolute Deviation) and inter-quantile range (IQR) instead of standard deviation. MAD is defined as median $(|x_i - \tilde{x}|)$, and IQR is defined as $IQR = Q3 - Q1$, where $Q3$ is the 3rd quarter ($75^{th}$) percentile and $Q1$ is the first quarter ($25^{th}$) percentile.

The list of dynamic features for every node is the following:

- for Inter-Arrival Time(**IAT**): **median-IAT**, Mean Absolute Deviation of IAT,
- for duration of the call: for incoming phone calls median call duration, in-MAD duration, and similarly for out-going phone calls: out-median duration, out-MAD duration.

### 6.1.8 Algorithm

Algorithm 2 shows the pseudocode of CallMine, with: feature extraction (line 1-3); attention routing (line 4, 10-14); and interaction (line 7-9). In short, the system does:

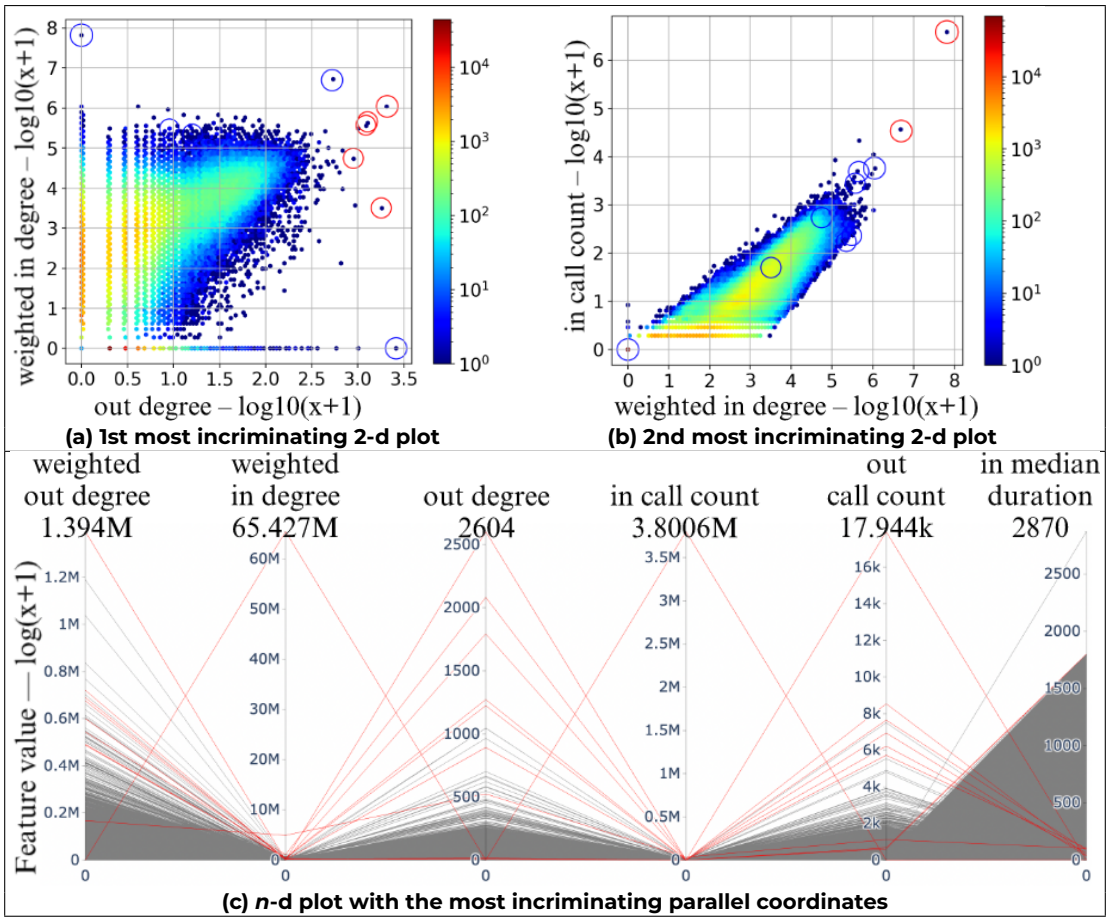
- feature extraction (line 1-3)
- attention routing CallMine-Focus(), (line 4-6, 10-14)
- interaction (line 7-9)

See Figure 6.3 with our findings.

**Algorithm 2:** CallMine: outline

**Data:** log of phone calls, and labels for 'fraudsters' (optional)
**Result:** fraudsters and outliers in $G$, and top-plots

1 Build a time-evolving graph $G$;
2 Extract static features: *core number, in/out-degree, in/out-weighted degree, and in/out-call count*;
3 Extract temporal features: *in/out-median-IAT, in/out-IQR-IAT, in/out-median-duration, and in/out-IQR-duration*;
4 Get anomalies and top-plots: `CallMine-Focus`*(n, d, b)*;
5 **if** *labels* **then** codify node colors;
6 Generate visualizations (see Sec 6.1.9): 1-d histograms, 2-d contour plots, interactive 2-d pair-plots, and n-d parallel coordinates;
7 **if** *user selected points with lasso* **then**
8 | Generate ego-net and plot corresponding features;
9 **Function** `CallMine-Focus`(*n, d, b*)**:**
| /* $n$ is the number of anomalies; $b$ is the budget (number of plots to show); $d$ is the dimensionality of plots ($d = 2$ for scatter-plots, $d > 2$ for parallel coordinates) */
10 | Detect $n$ anomalies/micro-clusters;
11 | Get the anomaly score (Isolation Forest) for every $d$–dimensional feature combination;
12 | Rank feature combinations according to scores;
13 | **return** top-$b$ $d$-dimensional feature combinations




(a) 1st most incriminating 2-d plot

(b) 2nd most incriminating 2-d plot

(c) *n*-d plot with the most incriminating parallel coordinates

**Figure 6.3:**
CALLMINE-Focus shows most-incriminating plots and anomalies. See in (a-b) circles in red indicating nodes most incriminated by the plots, and circles in blue indicating other outliers detected by CALLMINE-Focus. In (c), red lines highlight detected outliers in the 'parallel coordinates'.

### 6.1.9 Visualizations

The second main design goal is to make our system effective by optimizing for explainability and interactivity. We propose to use visualization.

There are several subtle issues:

- **Choices**: We must choose which scale to use (linear, logarithmic, etc.) and types of visualizations – between plotting methods, *e.g.*, histograms, scatter plots, violin, and pie plots.
- **Curse of dimensionality**: We must visualize a medium-dimensionality space in an effective manner.
- **Scalability**: We must plot, and interact with, millions of data points.

For explainability, we adopt a set of intuitive and interpretable visualizations:

- **(i) 1-D Histograms**: These plots help visualize the distribution of individual features and

reveal patterns such as skewness or outliers.

- **(ii) 2-D Scatter Plots and Scatter Matrices** (as shown in Figure 6.4): These allow analysts to explore pairwise relationships between features and detect clusters, correlations, or anomalies.
- **(iii) Parallel Coordinates**: This technique enables the visualization of multi-dimensional patterns and helps identify feature combinations associated with suspicious behavior.
- **(iv) Logarithmic Scaling**: For axis scaling, we routinely apply logarithmic transformations (typically $\log(x+1)$ to handle zeros), as many features follow power-law or heavy-tailed distributions. This improves visibility and interpretability in the presence of extreme values.

We use only a few carefully designed features, to address the curse of dimensionality. For scalability, plotting millions of user data points can make interactions slow or even infeasible. We propose two solutions to address this:

- **(i) Heatmaps** (as in Figure 6.4(i.b–c)), which eliminate duplicate points and reduce overplotting issues;
- **(ii) Filtering of low-activity users or nodes**, which allows us to discard a large number of points unlikely to correspond to fraudsters.

For interactivity, we enable three main interactive visualisation features to support analysts in exploring the data:

- **(i) Label Hovering**: When the user hovers over a node in the plot, a label card is displayed showing the node's ID and its corresponding feature values. This allows analysts to quickly inspect specific points of interest.In addition to this basic functionality, the label card may dynamically adjust its position to avoid occlusion and remain visible regardless of the plot's density.
- **(ii) Labelled Node Highlighting**: If labels (e.g., known fraudsters) are available, the system automatically highlights these nodes in the plots. Analysts can configure visual attributes such as colour or opacity to distinguish them more clearly from unlabelled data.This visual distinction aids in pattern recognition, allowing analysts to quickly identify clusters or connections involving labelled entities.
- **(iii) Brushing and Linking**: Users can select a region of interest by dragging the mouse across a plot. This selection is immediately reflected across all other plots, enabling analysts to explore conditional distributions and correlations between features in real time.This coordinated interaction facilitates multidimensional analysis by synchronising views across multiple visualisations such as scatter plots, histograms, or parallel coor-

dinate plots, based on the selected subset of data. It helps uncover hidden patterns, outliers, or class separability that might not be apparent in a single view.

To further support explainability and assist analysts in investigating suspicious-looking nodes or groups of nodes, we provide subgraph-level visualisations tailored to the scale of the structure under inspection.

For small neighbourhoods (e.g. the e.g.o network of a single node or a tightly connected group with fewer than 100 nodes), we use a *spring-model layout*, which places nodes using force-directed positioning. This intuitive visual representation reveals local connectivity patterns, such as hubs, bridges, and tightly-knit communities, making it easier to spot anomalies like star-like topologies or disconnected components.This makes the layout particularly effective for exploratory tasks involving local structural roles, such as identifying central actors, peripheral outliers, or unusual connection motifs.

For larger subgraphs, spring layouts quickly become unreadable due to node overlap and edge clutter. In such cases, we switch to a *spy-plot* representation of the adjacency matrix. To enhance interpretability, we apply spectral or heuristic reordering of rows and columns, bringing structure (e.g., block patterns or bipartite cores) into view. This matrix-based view is particularly effective at exposing regularities, cliques, or near-bipartite structures, as illustrated in Figure 6.4 (ii.b), where the two dense groups of ghost chasers and ghost destinations form a clear pattern despite their attempt to blend into the network.This approach scales well and offers a compact alternative to traditional graph layouts. By revealing structural symmetries or isolated blocks, the *spy-plot* becomes a valuable tool in detecting covert coordination or segmentation in large-scale networks.

Together, these visual tools provide analysts with both a local and global lens to investigate anomalous behaviour in the network structure, tailored to the size and complexity of the subgraph in question.

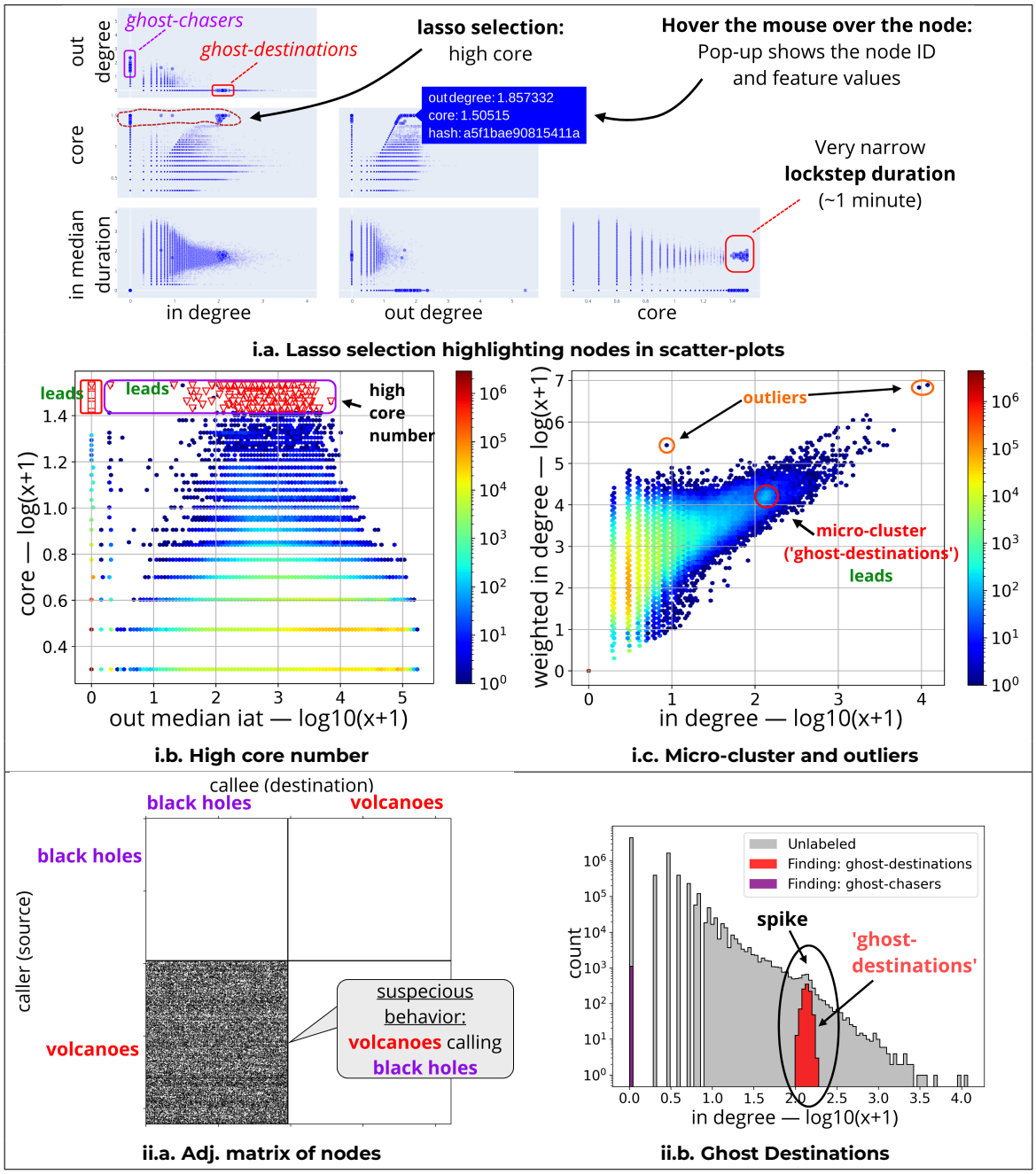



**Figure 6.4:**

CallMine spots suspicious behaviour with useful tools. i.a: 'Lasso' selection highlights nodes in all scatter-plots. Selected nodes are suspicious: i.b. they have high core numbers and ii.c. form a micro-cluster. These nodes are the same as the Ghost Chasersand Ghost Destinationsof the Introduction. The ego-net (ii.a) of selected nodes confirms that. The two groups form a near-bipartite core with Ghost Chasersand Ghost Destinations. ii.b. Despite their efforts to blend in, CallMine finds them.

### 6.1.10 Complexity Analysis

The time complexity of CallMine is $O(|E|)$, that is, linear on the number of edges $E$.

*Proof.* All our proposed features require a constant number of passes over the edges (phone calls): the in-/out- degrees require each a single execution and a group-by; similarly for the inter-arrival time features. For the core number, the straightforward algorithm requires a constant number of passes over the set of phone calls. ■

Figure 6.5 from Section 6.1.12 shows a linear plot, as expected.

### 6.1.11 Experiments

Here we aim to answer the following questions:

***Q1.*** How **scalable** is CALLMINE?

***Q2.*** How **effective** is CALLMINE on real data?

The anonymized phone-call graphs we used in our experiments are described in Table 6.2. They are quadruplets of the form (caller, callee, timestamp, duration). Each row of the datasets is a call.

**Table 6.2:** Specifications of our datasets.

| Dataset | #calls | #nodes | #edges | #known fraudsters |
|---|---|---|---|---|
| *ds-large* | 17.6M | 515.6k | 3.4M | 21.7k |
| *ds-huge* | 34.3M | 7.5M | 10.6M | 8.5k |

### 6.1.12 Q1 - Scalable

Figure 6.5 shows the execution time for *ds-large* and *ds-huge* and some of their subsets. It takes about *1 hour* for about *35 million* phone calls, on a stock laptop (M1 MacBook Air, 16GB RAM).

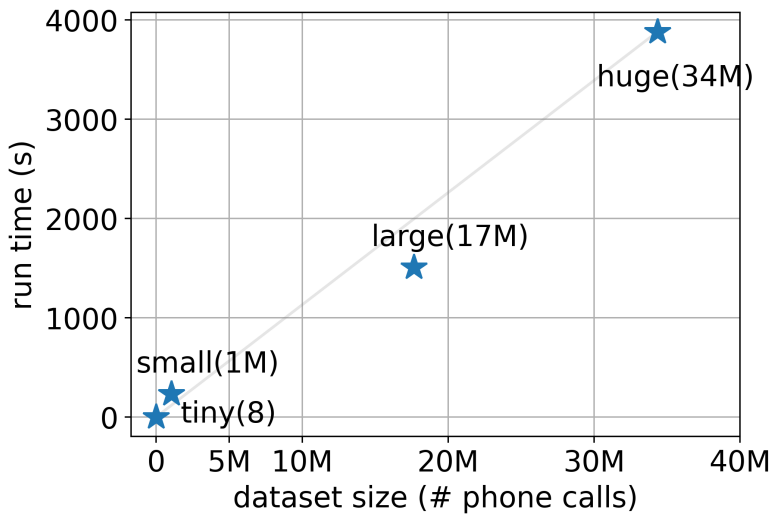



**Figure 6.5:**
CALLMINE scales near-linearly on the dataset size. : run time (seconds) versus dataset size (# phone calls). using an M1 chip MacBook Air with 16GB RAM.

Figure 6.6 show the scalable visualization of the average call duration versus call count per entity, generated from the 34M-call "huge" dataset. The result shows that the system supports efficient summarization and plotting even at large scale.

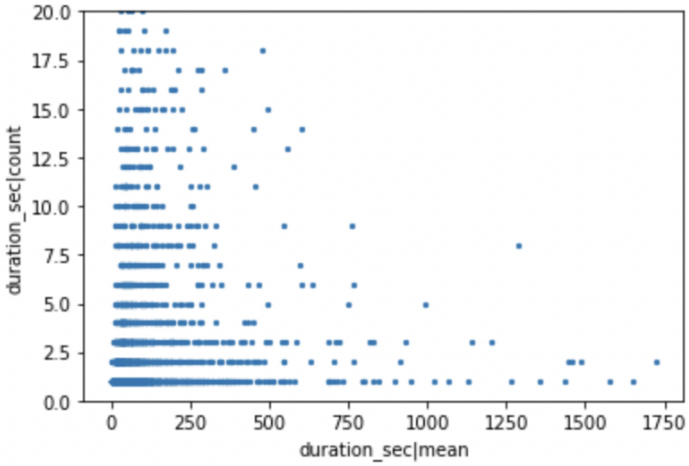



**Figure 6.6:** Average call time versus count

### 6.1.13 Q2 - Effective

CALLMINE processed real data and noticed the following traits.

**Observation 1** (**Sudden Cut-off**). *There is an unusual cut-off at thirty minutes for a large number of phone calls.*

Figure 6.1(i.a) shows the probability density function (PDF) of phone call duration (lin–log scales), with labels indicating fraud (light brown) and non-fraud (orange). Calls originate from multiple source numbers and are directed to multiple destination numbers.

Red flags: highly similar call durations, clustered exactly at thirty minutes; a high fraction (approximately 50%) of confirmed fraudsters. Further analysis revealed that roughly half of these calls had already been flagged as fraudulent, and that calls with longer durations also exist, ruling out any network-imposed configuration limit. This suggests a fraud strategy in which high-usage profiles are deliberately created to mask minor deviations, thereby evading profiling-based detection rules.

**Observation 2** (***'bi-lockstep'***). *(bipartite-lockstep) Some customers form bipartite cores, composed of several 'volcanoes' (high out-degree, near-zero in-degree), connected to the same set of 'black holes' (the reverse).*

Figure 6.1(ii.a) presents a scatter plot of customers, showing in-degree versus out-degree (using 'triple-log' scales, including the colour map). A large number of customers appear as either 'volcanoes' or 'black holes'; black marks confirmed fraudsters (squares for black holes

and triangles for volcanoes). There is an unusually low density of points along the diagonal, which would otherwise suggest reciprocity, commonly expected in such networks.

CallMine enabled the identification of suspicious nodes in Figure 6.4(i.a.): We interactively selected these nodes and examined their feature behaviour.

Fraudsters exhibited patterns consistent with voice bypass as identified by a Fraud Management System; however, CallMine was also able to reveal fraudsters that successfully evaded detection, namely the purple group shown in Figure 6.4(i.b).

This group corresponds to the same purple group shown in Figure 6.4(i.b), and they were able to camouflage their behaviour by intentionally increasing the volume of calls to free numbers, service numbers, and non-existent numbers ('ghosts'), which are 'black holes' with zero out-degree.

While a high out-degree towards regular local numbers would typically trigger alarms and prompt investigation, fraudsters instead artificially generated a large out-degree composed of no-cost calls to conceal the actual voice bypass numbers, while varying the **IAT** between calls (Figure 6.4(ii.b)). Fraud analysts may interpret this high dispersion as non-fraudulent behaviour and therefore dismiss the case or exclude the number from further analysis.

By matching known black hole micro-clusters used by fraudsters to camouflage their behaviour (Figure 6.4(i.c, ii.a)), CallMine identified additional potential fraudsters in the network, including nodes with a high core number (Figure 6.4(i.b)) and a dense adjacency matrix (Figure 6.4(ii.a)).

Red flags: high density; high core number; zero out-degree; and non-functioning destination.

**Observation 3** (**Ghost Destinations**). *A micro-cluster of approximately 900 nodes (Figure 6.4(i.c)), composed of multiple unallocated numbers, according to the regulator numbering plan, which only receive inbound calls from several different numbers, with durations close to 1 second each.*

Red flags: The characteristics of this group are consistent with other black hole micro-clusters previously used by confirmed fraudsters to camouflage high call volumes and out-degree.

**Observation 4** (**Ghost Chasers**). *The sources from Observations 2-3 are very similar to confirmed fraudsters (see Figure 6.1(i.b)), and they all mostly call the Ghost Destinationsof Observation 3.*

**Observation 5** (***'heavy-hitters'***). *We applied the 'lasso' functionality of CallMine to Fig-*

*ure 6.4(i.a), which enabled the identification of high-activity nodes. CallMine revealed suspicious outliers (orange, 6.4 i.c), as well as two particularly suspicious groups forming a near-bipartite core (ii.a).*

Red flags: All of them exhibited high density (**core-number**), and the selected set included both confirmed fraudsters (in red) as well as others (either 'honest', or not-yet-detected fraudsters). Further inspection by a domain expert revealed that although many of them interact extensively with confirmed fraudsters,they themselves were not labelled as such. These are the types of nodes that warrant further investigation by a telecom analyst.

Figure 6.1(i.b) shows a scatter-plot of customers, plotting the in-degree versus the out-degree (in 'triple-log' scales: even the colormap is in log). Notice that a huge number of customers are either 'volcanoes' (high out-degree; near-zero indegree) or 'black holes' (the reverse); black indicates confirmed fraudsters (square for black holes and triangles for volcanoes). Also notice the abnormally low count of points along the diagonal (which would imply reciprocity, which is expected in such networks).

Figure 6.1(i.b.) shows fraudsters that were identified as having a pattern similar to voice bypass by a Fraud Management System, yet CallMine was able to detect an interesting group that successfully managed to avoid detection, the purple group in Figure 6.1(ii.a). This group managed to camouflage their behaviour by deliberately increasing their volume of calls to free numbers, service numbers, and non-existent numbers. While high out-degree to regular local numbers would trigger alarms and consequent investigation, fraudsters artificially created a huge out-degree made from no-cost calls to cover the real voice bypass numbers, while playing with different inter-arrival times between the calls 6.1(ii.a). Fraud analysts dealing with these cases can be tricked in associating the high dispersion to non-fraudulent traffic and would dismiss the case, or discard the number from further analysis in the future, ensuring that number will not re-alarm. By matching the known black hole micro clusters used by fraudsters to camouflage their behaviour (Figure 6.1 (ii.a)) CallMine identified other potential fraudsters in the network, both with the same core number and similar adjacency matrix (Figure 6.1 (ii.b)).

### 6.1.14 Credit authorship contribution statement

**Paper:** CallMine: Fraud Detection and Visualization of Million-Scale Call Graphs
**Doi:**https://doi.org/10.1145/3583780.3614662
CIKM 23: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management
**Pedro Fidalgo**: Investigation, Methodology ,Software, Writing – original draft & editing.
**Mirela T. Cazzolato**: Investigation, Methodology ,Software, Writing – original draft & editing.

**Saranya Vijayakumar**: Investigation, Writing – original draft & editing.

**Xinyi Zheng**: Investigation, Software

**Namyong Park**: Investigation, Software

**Meng-Chieh Lee**: Investigation, Software

**Catalina Vajiac**: Investigation , Software

**Agma J. M. Traina**: Investigation, Methodology, Supervision.

**Christos Faloutsos**: Investigation, Methodology, Supervision, Writing – original draft & editing.

# Chapter 7

# Federated Approach to Detect Fraud on the 5G Edge

## 7.1 Context and Motivation

Building on the challenges and insights outlined in Chapters 4,5,6, ranging from scalable graph-based detection, to interactive exploration of dynamic fraud patterns, and the role of visualisation in analyst workflows, we now turn to the paradigm shift introduced by 5G mobile networks. The advent of 5G is not just accelerating communication speeds and reducing latency; it is fundamentally reshaping the architecture of computation through the enablement of distributed intelligence at the network edge.

At the heart of this transformation lies Multi-access Edge Computing (MEC), a cornerstone of the 5G ecosystem. MEC brings compute and storage capabilities closer to end-user devices, thereby minimising dependence on centralised cloud infrastructure. This proximity yields several advantages: improved privacy, reduced response times, and greater energy efficiency, benefits that are particularly salient in high-throughput, latency-sensitive contexts like telecommunications fraud detection.

Among the various fraud scenarios examined in Chapters 5 and 6, International Revenue Share Fraud (IRSF) emerged as a particularly compelling case in MEC enabled 5G environments. Unlike more static or easily localized fraud types, IRSF is characterised by its inherently dynamic behaviour and its distributed execution across multiple network domains. Attackers exploit international interconnect agreements to redirect traffic toward high-cost destinations, often in collaboration with third-party entities, creating a constantly shifting pattern that is difficult to pinpoint. The dynamic and distributed nature of this fraud type makes it especially difficult to detect using traditional methods, which typically rely on centralized data processing and model training.

Mainly, the limitations of centralisation in 5G-enabled environments, ranging from the technical burden of aggregating data across distributed sources to the increasing weight of privacy regulations and data sovereignty concerns—underscore the need for a new approach. This approach must reconcile the demand for accurate and timely fraud detection with the imperative to keep sensitive data local.
In this chapter, we investigate the use of Federated Learning (FL) as a privacy-preserving, scalable solution for fraud detection in MEC environments. FL enables collaborative training of machine learning models across distributed nodes without requiring raw data to be shared centrally. Within the scope of the Opti-Edge research initiative, we explore how lightweight FL architectures can effectively detect IRSF at the edge, leveraging local insights while preserving confidentiality.
The core motivation driving this exploration is the reconciliation of two key objectives: maximizing detection accuracy and maintaining decentralized control over sensitive data. This direction aligns with broader industry trends seeking AI-driven, edge-native solutions capable of addressing the unique demands of next-generation telecom infrastructures.

## 7.2 Federated Learning Architecture

### 7.2.1 Design Goals and Constraints

As outlined in Chapter 5, detecting International Revenue Share Fraud (IRSF) in mobile networks requires timely, localized analysis of signaling and usage patterns. Traditional ML models that rely on centralized data collection are often impractical in 5G environments due to privacy constraints, regulatory fragmentation, and the geographic dispersion of network nodes. Moreover, as discussed in Chapter 5, the temporal and spatial dynamics of fraud demand frequent model updates and local adaptation.

To address these challenges, we adopt a Federated Learning (FL) approach tailored to Multi-access Edge Computing (MEC) environments. The main design goals are:

- Enable collaborative training of fraud detection models without sharing raw data.
- Support lightweight inference and training at the edge.
- Allow dynamic adaptation to local patterns while preserving global consistency.

Figure 7.1 provides an overview of the proposed architecture, highlighting the interplay between edge nodes and a central aggregator.

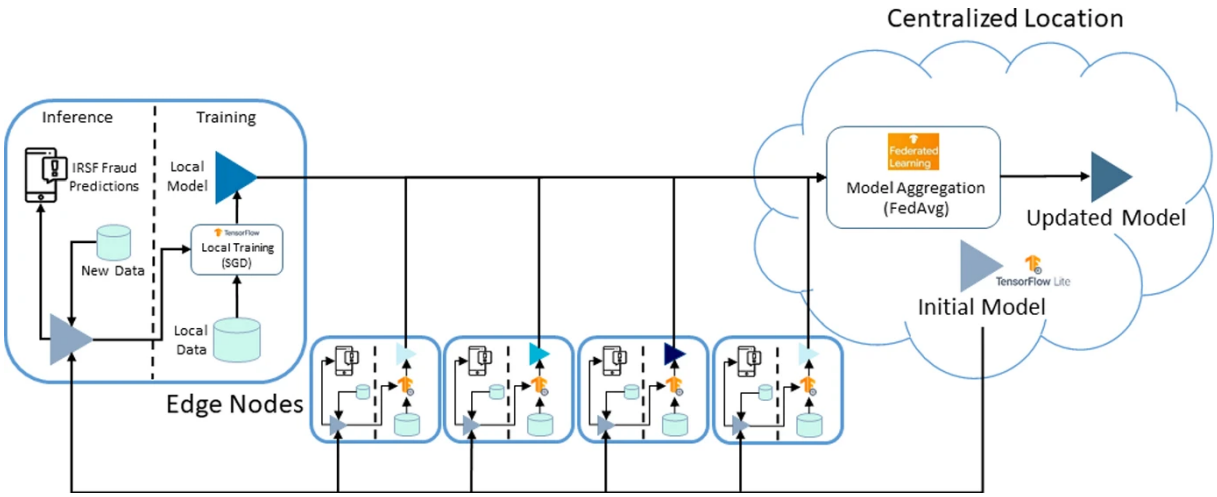



**Figure 7.1:** The proposed ML architecture for Federated Learning at the edge.

### 7.2.2 Edge Node Functionality

Each edge node is responsible for two main tasks: real-time fraud prediction and local model training. For inference, a shared model is deployed in TensorFlow Lite format to ensure compatibility with constrained devices. The model receives input features derived from call detail records (CDRs), such as duration, origin/destination numbers, and country codes, and outputs a binary fraud score.

Periodically, when sufficient labeled data becomes available, the edge node updates its local copy of the model. This local training process is executed using only the node's internal dataset, preserving data privacy. Nodes without enough fresh data may skip training rounds, ensuring resource efficiency.

The internal workflow of an edge node is illustrated in Figure 7.2, showing the distinct paths for inference and training, and how updated model weights are produced locally.

### 7.2.3 Model Aggregation and Coordination

Once local training is completed, each edge node transmits its updated model to a central aggregator. This component does not access the raw data but performs model fusion using the Federated Averaging (FedAvg) algorithm. Each local model's contribution is weighted based on the volume of training data used.

The complete lifecycle of a federated training round including distribution of the base model, local updates, and global aggregation is depicted in Figure 7.1. After aggregation, the global model is redistributed to all edge nodes, replacing previous versions and resuming inference with improved weights.

This cyclical process ensures that the global model continually evolves with localized insights, while maintaining cohesion across the distributed system.

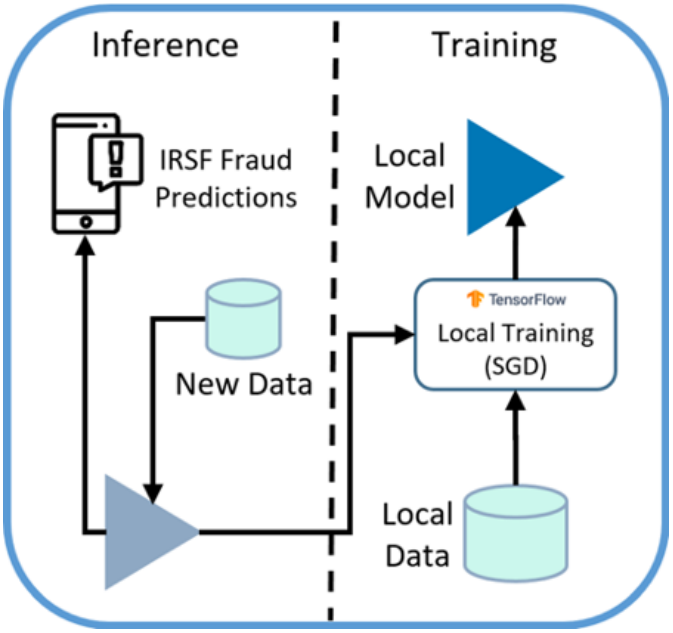



**Figure 7.2:** Representation of an edge node.

### 7.2.4 Adaptation and Resilience Mechanisms

To improve robustness, the architecture supports selective participation during model aggregation. Only local models that meet predefined quality thresholds (e.g., minimum AUC or data volume) are included. This prevents degraded local data from contaminating the global model.

In addition, the system accommodates asynchronous updates. Edge nodes may train or receive updates at different intervals, accommodating heterogeneous hardware capabilities, network latencies, and operational loads. This design choice enhances real-world deployability, especially in mobile operator networks with uneven infrastructure maturity.

Collectively, these mechanisms provide a scalable and privacy-preserving framework for deploying AI-powered IRSF detection at the edge of 5G networks.

## 7.3 Data and Methods

### 7.3.1 Data

The dataset used in this work consists of anonymized call detail records (CDRs) collected over a one-month period from a real telecommunications environment. Each record is labeled as fraudulent or non-fraudulent based on retrospective analysis and expert heuristics, with the fraud type corresponding to International Revenue Share Fraud (IRSF), a scheme explored in Chapter 5.

The dataset includes 78.174 events and 14 features retained after cleaning. These features

include subscriber identifiers (IMSI, IMEI), phone numbers (A and B numbers), call metadata (start time, duration, type), and indicators such as roaming status and country code. All categorical attributes were encoded using label encoding, transforming textual values into integers for compatibility with neural models.

### 7.3.2 Partitioning and Training Strategy

To reflect time-dependent fraud dynamics as discussed in Chapter 5, the dataset was split chronologically. The first half was used to train a centralized baseline model, and the second half was divided among three virtual edge nodes for federated training.

Partitioning across nodes was done using country dialing codes (e.g., +351, +352), simulating the geographic and routing-based data segmentation seen in telecom networks. This setup supports realistic evaluation of the federated architecture in distributed conditions.

### 7.3.3 ML Framework: TensorFlow, Lite, and Federated

The model development and training pipeline used the TensorFlow ecosystem [1].TensorFlow supports both high-level APIs for model creation and low-level APIs for advanced control across CPUs, GPUs, TPUs, mobile devices, and edge nodes.

Two specific components were used:

- **TensorFlow Lite** a lightweight runtime optimized for edge inference [2].
- **TensorFlow Federated** a library designed for training models across decentralized clients without sharing raw data [3].

These tools were chosen for their alignment with the edge-centric requirements discussed in Chapter 6.

### 7.3.4 Neural Network Architecture and Training Parameters

The classification model was a Multilayer Perceptron (MLP) with four hidden layers using Refctified Linear Unit (ReLU) activations, and a logistic output for binary classification. The model was trained using the Adaptative Moment Estimation (Adam optimizer) and binary cross-entropy loss. The Area Under the ROC Curve (AUC) was used as the evaluation metric due to its robustness in imbalanced fraud detection contexts.

Early stopping was enabled using validation data (25% of training data), and the model was executed over a maximum of 100 epochs. The final model was converted into TensorFlow Lite format for deployment in simulated edge nodes.

## 7.4 Evaluation and Results

To assess the effectiveness of the proposed federated learning approach for IRSF detection, we conducted a comparative evaluation against a centralized baseline model. This section presents the AUC performance results across different stages of training, highlighting both the consistency and the adaptability of the federated model when deployed at the edge.

The initial model was trained using the first half of the dataset in a centralized manner and served as a performance reference. Its AUC score was 0.980 on the corresponding test set, confirming that the base neural architecture was capable of learning relevant fraud patterns.

For the federated learning setup, two rounds of training were simulated using the second half of the dataset. Each of the three edge nodes received a distinct partition of the data based on dialing codes. In each round, nodes that had sufficient data performed local training, and their updated models were sent to the central aggregator.

Table 7.1 summarizes the results. During the first federated round, each edge node produced a local model with an AUC between 0.973 and 0.977, while the aggregated global model achieved 0.981, slightly outperforming each individual node. In the second round, the global AUC remained at 0.980, showing consistency with the initial centralized model.

For reference, a Random Forest model trained using a centralized approach previously used as a production benchmark achieved an AUC of 0.932. This underscores the advantage of the proposed neural model architecture and the effectiveness of the federated training process.

**Table 7.1:** AUC performance of local and aggregated models across FL rounds.

| Training Phase | Edge 1 | Edge 2 | Edge 3 | Global (Main) |
|---|---|---|---|---|
| Initial Model (Centralized) | – | – | – | 0.980 |
| 1st FL Round | 0.974 | 0.977 | 0.973 | 0.981 |
| 2nd FL Round | 0.976 | 0.964 | 0.975 | 0.980 |
| Centralized RF Baseline | – | – | – | 0.932 |

To examine model drift and support the need for ongoing retraining, we evaluated each global model on later test partitions. Table 7.2 shows that AUC degradation over time was minimal, with drops under 1 percentage point across training rounds. This suggests that, despite concept drift, the model remains stable over the short-to-medium term.

**Table 7.2:** AUC degradation of past global models on newer test sets.

| Evaluation Round | Current Model | Previous Round | Initial Model |
|---|---|---|---|
| 1st FL Round | 0.981 | 0.979 | 0.980 |
| 2nd FL Round | 0.980 | 0.978 | 0.977 |

Overall, the results confirm that the federated learning strategy:

- Maintains performance parity with centralized training;
- Offers better generalization through aggregation;
- Adapts well to new data while preserving edge-level privacy;
- Outperforms the previously used centralized Random Forest baseline.

## 7.5 Results

Table 7.3 reports the results obtained from the conducted experiments. For each Federated Learning step, the table presents the test set AUC values, both for the local models (trained on the edge nodes) and for the aggregated main model (Main) produced from the combination of the local models. As the initial model was trained using centralized data, the corresponding local model results are not reported for that round. For each learning step, the reported AUC corresponds to the performance on the test set for that specific round, for instance, the AUC values for the first round of local training were all computed using the same test set.

Additionally, we include the results of a Random Forest model developed by the company using a fully centralized approach. This machine learning model is used as a baseline for comparison, as it is currently employed by the software company for IRSF detection.

**Table 7.3:** Results obtained by the local models and the aggregated models during the Federated Learning iterations.

| Round | Edge 1 | Edge 2 | Edge 3 | Main |
|---|---|---|---|---|
| Initial Model Round | - | - | - | 0.980 |
| Local Training (1st Round) | 0.974 | 0.977 | 0.973 | 0.981 |
| Local Training (2nd Round) | 0.976 | 0.964 | 0.975 | 0.980 |
| Baseline | - | - | - | 0.932 |

The results indicate that, during the two rounds of local training, the AUC values obtained on the test set can be regarded as being of excellent quality (consistently above 0.97), both for the local models and for the main models (which aggregate all local models of the corresponding round). Furthermore, across the two rounds of local training, the main model was able to preserve the AUC value achieved by the initial model, which had been trained using centralized data only. This represents a positive indication that the Federated Learning process allows the ML model to retain its predictive performance when identifying IRSF fraud, especially considering that, in the telecommunications domain, ML models typically require frequent updates. This requirement is largely driven by the fact that telecommunication companies continuously generate very large volumes of data (e.g., phone calls, SMSs), while new fraud strategies are constantly introduced by fraudulent actors (e.g., new fraudulent phone numbers).

Another relevant observation is that, in both the first and second rounds of local training, the performance of each of the three local models (one per edge node) was slightly inferior to

that of the main model. This outcome can be explained by the fact that the test sets for each round include phone calls originating from all three edge nodes (which partition the data by phone codes). As a result, it is not expected that an individual local model performs at the same level as the aggregated main model. Nevertheless, the maximum predictive gap between a local model and its corresponding main model was below 1 percentage point (pp).

When comparing these results with the baseline model, it can be observed that both the initial model and the two aggregated models achieve higher AUC values than the baseline, with an average improvement of 5 pp. In a similar manner, all local models also outperform the baseline Random Forest, showing an average improvement of 4 pp. These comparisons should, however, be interpreted with caution, as the baseline model may have been trained using a different time period. In addition, unlike our approach, the baseline relied on distinct preprocessing procedures and was developed using a centralized dataset. Even so, this comparison provides further evidence of the potential of adopting Federated Learning at the edge in the telecommunications domain.

In addition, we examined the predictive performance of previous main models across different rounds of local training. The primary objective was to assess performance degradation over time and to evaluate the need for model updates, which are achieved through the processes of local training and aggregation.

The corresponding results are presented in Table 7.3, confirming that the performance of the main models gradually decreases over time when applied to more recent data. It should be noted that these differences are relatively small (with a maximum variation of less than 1 pp), which can be attributed to the limited duration of the dataset (one month of data). As reported in other studies applying ML in the telecommunications domain, model performance may remain stable for several months before experiencing a more pronounced decline. Nonetheless, these findings reinforce the necessity for continuous model updates, which can be effectively supported through Federated Learning in decentralized data settings.

In this domain, it is common to have multiple edge nodes that are unable to share data with each other or with the cloud (e.g., due to confidentiality constraints). Accordingly, the proposed framework employs Federated Learning to aggregate models trained on distributed datasets. To evaluate the framework, we simulated two rounds of federated training and aggregation using a dataset provided by the company related to IRSF fraud. The obtained results were then compared against a baseline model developed prior to this work. The results demonstrate that the decentralized approach based on Federated Learning achieves an excellent level of class discrimination (consistently above 0.97) and is able to maintain its performance over two training rounds. Moreover, the federated ML model outperformed the centralized Random Forest model currently used in production.

## 7.6 Discussion

The experimental results confirm that federated learning is a viable and effective approach for training fraud detection models in distributed telecom environments. By decentralizing the learning process, the proposed architecture addresses key challenges of centralized systems, including data privacy, regulatory constraints, and communication overhead all of which are particularly relevant in real-world telecom infrastructure.

The federated learning models achieved predictive performance on par with, and in some cases superior to, the centralized baseline. As shown in Table 7.3, the global model obtained through federated aggregation consistently achieved AUC values of 0.980 or higher across two training rounds. Each of the three edge nodes also produced strong local models, with AUC values ranging from 0.964 to 0.977. While the performance of individual local models varied slightly, the aggregated model consistently achieved the highest score in each round. This highlights the value of model aggregation in capturing broader fraud patterns that no single node observes in isolation.

This behaviour confirms the strength of the aggregation strategy: although each edge node had access to only a limited and partitioned subset of the data, the federated approach allowed the global model to generalize better by synthesizing knowledge across all participants. Importantly, this was achieved without sharing raw data, thus respecting data localization and privacy constraints.

In contrast, the centralized Random Forest model previously used in production attained a lower AUC of 0.932. While not a perfect comparison due to differences in model type and validation strategy, this result reinforces the advantage of using a neural network trained via federated learning in this domain.

Temporal stability was also observed. Table 7.2 shows that the global models retained strong performance when evaluated on newer data. AUC values dropped by less than 1 percentage point between rounds, suggesting resilience to short-term changes in fraud behaviour. This supports the architecture's suitability for deployment in dynamic environments where frequent model updates are needed but complete retraining may not be feasible.

These findings align with and complement the technical insights developed in earlier chapters. Chapter 4 focused on static topological analysis of fraudulent call networks, Chapter 5 emphasized the importance of time-evolving behaviour in fraud structures, and Chapter 6 addressed the scale and interpretability challenges of operating over billion-node graphs. The federated learning framework builds on these foundations by introducing a decentralized training mechanism that is both scalable and adaptive, capable of operating close to the data source while preserving detection performance.

From a deployment perspective, the system supports lightweight models using TensorFlow Lite, asynchronous updates, and selective participation of edge nodes. These features make it particularly well-suited for environments with heterogeneous infrastructure or limited compute resources. In practical terms, this enables telecom operators to deploy and maintain fraud detection models at the edge of 5G networks without requiring extensive centralized data movement or manual retraining cycles.

In summary, this work demonstrates that federated learning enables scalable, privacy-preserving, and high-performance fraud detection on the network edge. It represents a key step toward operationalizing distributed intelligence in telecom security systems, and opens the door to future work on personalized models, cross-operator collaboration, and integration with graph-based behavioral analytics.

### 7.6.1 Comparison with Centralized Models

The centralized and federated approaches employ different model families due to the fundamentally different operational constraints of each environment. The centralized setting enables the use of more computationally intensive models trained on fully aggregated data, benefiting from complete visibility over the dataset and fewer limitations in terms of processing power and memory. In this work, the centralized baseline is implemented using a Random Forest model, a strong and widely adopted approach for tabular data classification.

In contrast, the federated setting operates under stricter constraints, including data decentralisation, limited computational resources at the edge, communication overhead, and privacy requirements. As such, it relies on lightweight and communication-efficient models that can be trained locally and aggregated across distributed nodes. The federated approach proposed in this thesis is based on a neural network architecture trained using a Federated Learning paradigm.

This distinction places the federated approach in an inherently disadvantaged position when compared to the centralized baseline. Not only is the model capacity typically lower, but each local model is trained on partial and potentially non-IID data, without access to the full global distribution. Despite these constraints, the experimental results demonstrate that the federated approach achieves performance levels comparable to the centralized model (AUC $> 0.97$), reinforcing the robustness and effectiveness of the proposed distributed learning framework.

It is important to emphasise that the objective of this comparison is not to benchmark model architectures, but rather to evaluate whether a distributed, privacy-preserving learning paradigm can achieve high detection performance under realistic deployment conditions. The results therefore reflect not only the capability of the models, but also the suitability of

the overall system design for 5G environments, where centralized data aggregation is often infeasible due to latency, privacy, scalability, and regulatory constraints.

Furthermore, these findings are grounded in datasets that reflect real-world telecom scenarios, characterised by large-scale, highly imbalanced, and behaviourally complex data distributions. As such, the observed performance is indicative of practical applicability, demonstrating that federated learning can operate effectively in environments that closely resemble operational telecom networks.

### 7.6.2 Credit authorship contribution statement

**Paper:** A Federated Machine Learning Approach to Detect International Revenue Share Fraud on the 5G Edge

**Doi:**https://doi.org/10.1145/3477314.3507322 SAC 22: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing
**Pedro Fidalgo**: Investigation, Software
**Paulo Cortez**: Investigation, Methodology ,Software, Writing – original draft & editing.
**Leopoldo Silva**: Investigation, Methodology ,Software, Writing – original draft & editing.
**Francisco Morais**: Investigation, Methodology ,Software, Writing – original draft & editing.
**Pedro Pires**: Investigation, Software
**Carlos Manuel Martins**: Investigation
**Helena Rodrigues**: Investigation, Methodology, Supervision, Writing – original draft & editing.
**Luis Ferreira**: Investigation, Methodology, Supervision, Writing – original draft & editing.

# Chapter 8

# Conclusions

## 8.1 Overview

Telecom fraud poses a critical threat to the telecommunications industry, with annual losses estimated at $28.3 billion USD [26]. The transition to 5G networks has amplified the complexity and scale of this threat, increasing the attack surface and enhancing the tools and techniques available to fraudsters. These changes demand a deeper, more nuanced understanding of network behaviour, particularly the dynamics of key actors and nodes within telecom infrastructures.

This thesis investigates Telecom Fraud not as a generic anomaly detection problem, but as a domain-specific challenge rooted in complex and evolving systems. A significant gap in the literature is the lack of domain knowledge applied to fraud detection; generic models often miss subtle, yet critical, indicators of malicious behaviour. By integrating operational knowledge of telecom environments with advanced analytical techniques, this work emphasises the importance of understanding fraud from within the system rather than as an isolated anomaly.

Central to this thesis is the exploration of the *capability* dimension in traditional fraud theory. While the classic fraud triangle highlights pressure, opportunity, and rationalization, this work builds upon the "fraud diamond" model, emphasizing the critical role of an actor's ability to commit fraud. In the context of Telecom networks, this capability is often embodied by *hubs* and *gatekeepers*: highly connected or strategically positioned nodes that play pivotal roles in data traffic and control. This thesis analyses the influence, behaviour, and characteristics of these nodes, proposing that their network roles make them uniquely capable of facilitating or detecting fraud.

### 8.1.1 Integrated Analysis of the Main Contributions

This thesis develops its contributions across five interconnected chapters. Chapter 3 investigates malware techniques used in Telecom fraud, examining how malicious software enables access, persistence, and control over telecom infrastructure. It establishes the technical foundation for understanding how sophisticated attackers operate.

Chapter 4 focuses on fraud detection frameworks, showing that traditional static models fail to capture the adaptive and contextual nature of telecom fraud. It argues for the incorporation of behavioural evolution and domain context in the identification of high-capability actors. This leads into Chapter 5, which introduces dynamic network analysis as a methodological solution. By modelling telecom environments as temporal graphs, the chapter reveals how the roles of hubs and gatekeepers emerge and evolve, enabling the detection of structural shifts linked to fraud.

Chapter 6 presents advanced visualization techniques that translate complex graph and behavioural data into accessible and interpretable formats. These visualisations support human-in-the-loop analysis and help domain experts identify fraud scenarios in real-time. Finally, Chapter 7 introduces federated learning as a scalable and privacy-preserving approach to fraud detection. This chapter shows how Telecom providers can collaboratively build models without sharing raw data, overcoming regulatory and operational constraints.

Together, these chapters form a comprehensive approach that fuses technical analysis, domain knowledge, and adaptive modeling to improve telecom fraud detection in modern, distributed networks.

### 8.1.2 Methodological Frameworks and Technical Highlights

This thesis employs a multi-methodological approach, each tailored to address specific challenges in telecom fraud detection while collectively reinforcing the roles of hubs and gatekeepers within complex, dynamic networks.

**MONDEO-Tactics5G: Multistage Malware Detection and Mitigation.** MONDEO-Tactics5G presents a layered approach to detecting and mitigating mobile malware in 5G networks. It identifies threats like FluBot by leveraging DNS-based traffic analysis and a feedback loop for adaptive blacklisting. A core strength of MONDEO-Tactics5G lies in its integration with 5G network functions (e.g., User Plane and Session Management Functions), enabling in-network traffic monitoring and responsive control. Additionally, it introduces tactical mitigation strategies that balance effectiveness with operational constraints, such as minimizing disruptions to customer service. This framework establishes foundational insights into how malware-enabled fraud actors gain capability—particularly by occupying gatekeeper-like roles that manipulate traffic flows and persist within the network.

**STARBRIDGE: Identifying Influential and Controlling Nodes.** STARBRIDGE introduces a topology-driven method for uncovering nodes that hold strategic control in fraud networks. These nodes, often situated in star-bridge-star configurations, exhibit high values in control centrality, bridging centrality, and influence, three complementary metrics that quantify a node's capacity to dominate, connect, and impact the flow of information across distinct communities. By ranking nodes based on their topological features and benchmarking against real-world fraud management systems, STARBRIDGE proves that high-influence nodes are significantly more likely to be involved in fraudulent activities. This work reinforces the conceptual link between fraud capability and network centrality, offering a scalable, generalizable tool to prioritize interventions against high-risk nodes.

**TgraphSpot: Interactive and Explainable Analysis of Temporal Graphs.** TgraphSpot addresses the temporal and evolving nature of fraud by enabling scalable, explainable, and interactive analysis of time-evolving call graphs. Its visual analytics tools allow fraud analysts to identify suspicious nodes based on intuitive features such as call duration, in-degree, and temporal patterns. By supporting lasso-based selection in multidimensional spaces and allowing deep dives into node behaviour over time, TgraphSpot positions analysts to trace the emergence and evolution of potential gatekeepers and hubs, making it a human-centric complement to automated methods like STARBRIDGE.

**CALLMINE: Scalable Detection in Billion-Scale Call Graphs.** CALLMINE scales fraud detection to billion-entry datasets by integrating density-based subgraph detection (e.g., FRAUDAR [46], M-Zoom [94]) with feature-based visualizations. It isolates both known and novel fraud patterns, such as ghost destinations and duration-based anomalies, with no need for parameter tuning. This method's flexibility in supervised and unsupervised contexts, coupled with its speed, makes it a powerful asset for detecting hubs in sprawling telecom graphs. CALLMINE advances the notion of fraud capability through statistical and topological anomaly detection in real-world, large-scale environments.

**SAC Federated Learning: Privacy-Aware Distributed Modeling.** SAC addresses the challenge of privacy and decentralization in telecom data by implementing a federated learning framework. It allows for collaborative training of detection models across distributed environments, preserving data locality and reducing latency. The framework supports neural architecture search and automated ML to tune models per node. Importantly, it lays the groundwork for adaptive, cross-organizational fraud detection strategies that respect regulatory constraints. SAC complements the other methodologies by operationalising detection at scale while safeguarding sensitive network data.

Together, these methodologies offer a complementary and layered toolkit for telecom fraud detection. They span low-level detection (MONDEO-Tactics 5G), mid-level network structure

analysis (STARBRIDGE), user-interactive exploration (TgraphSpot), large-scale pattern discovery (CALLMINE), and distributed learning (SAC), each reinforcing the critical role of hubs and gatekeepers in understanding and disrupting fraudulent behaviour.

### 8.1.3 Operational Relevance and Industry Implications

This research was developed with a strong emphasis on practical deployment and adaptability within real-world telecom infrastructures. The frameworks proposed in this thesis are not only theoretically sound, but also aligned with current and emerging industry needs, particularly in the context of 5G networks, fraud management systems, and data privacy regulations. Two of the methodologies developed in this thesis, STARBRIDGE and MONDEO-Tactics 5G, are currently being evaluated experimentally in collaboration with three telecommunications operators. These real-world deployments are providing valuable insights into their operational viability, integration complexity, and effectiveness in detecting fraudulent activity.

The MONDEO-Tactics 5G system demonstrates direct compatibility with 5G core network functions such as the User Plane Function (UPF) and Session Management Function (SMF). This integration allows for near-real-time malware detection and tactical mitigation embedded within network operations, offering telecom operators a scalable and low-latency defence mechanism.

STARBRIDGE, by revealing topologically influential and controlling nodes within fraud networks, provides a robust decision support layer for telecom analysts and fraud investigation teams. Its influence ranking metrics can augment or even guide prioritisation strategies in fraud management platforms, helping target actors with the highest impact potential.

TgraphSpot and CALLMINE further this operational relevance through their user-centred design and massive-scale capabilities. TgraphSpot facilitates intuitive and interpretable fraud exploration, offering analysts the tools to visualise node evolution, detect unusual call patterns, and investigate network anomalies in a guided, interactive manner. CALLMINE delivers high-throughput, automated anomaly detection on billion-scale graphs, ensuring operational viability in large telecom environments without compromising on explainability or effectiveness.

SAC Federated Learning ensures that machine learning-based detection solutions remain privacy-preserving and deployment-friendly. It enables multiple telecom entities to benefit from shared learning without compromising customer data confidentiality, aligning with General Data Protection Regulation (GDPR) and other global data governance policies. This not only improves collaborative fraud detection, but also reduces operational risk.

Collectively, these methodologies bring tangible improvements in detection accuracy, interpretability, scalability, and privacy, ensuring their direct applicability in modern telecom fraud

defence systems. They enable telecom operators to respond proactively to evolving threats while maintaining system reliability and user trust.

### 8.1.4 Limitations and Challenges

While the methodologies presented in this thesis advance the field of telecom fraud detection, several limitations and challenges remain.

First, many of the detection mechanisms rely on partially labelled datasets or heuristic rules, which can limit generalisability and introduce biases. The effectiveness of supervised models, such as those in SAC, depends on the availability and quality of the labelled data, which is often scarce or inconsistently annotated in operational environments.

Second, the trade-off between scalability and interpretability presents ongoing challenges. For instance, while CALLMINE achieves high performance on billion-scale datasets, its effectiveness can be constrained in environments requiring strict real-time processing or integration with low-latency systems. Similarly, the interpretability of complex models—particularly deep learning approaches used in federated settings—can hinder trust and adoption among telecom analysts.

Third, while temporal analysis is a strength of tools like TgraphSpot, dynamic fraud behaviours can sometimes evolve in ways that detection systems may not immediately capture. This highlights the need for models capable of continual learning and adaptation without extensive retraining. Moreover, although adversarial behaviours such as fraudsters altering call patterns, mimicking benign behaviour, or applying camouflage techniques have been significantly mitigated by the approaches proposed in this thesis, the possibility of evasion remains and demands ongoing vigilance.

Finally, the integration of these methods into production systems involves technical, organisational, and regulatory barriers. Issues such as cross-departmental coordination, explainability requirements, and evolving compliance standards must be addressed for full-scale deployment. These limitations suggest several avenues for future research, particularly in enhancing robustness, automation, and human-in-the-loop interpretability.

### 8.1.5 Future Directions and Research Opportunities

One promising direction is the development of Federated Generative Knowledge Graphs (FGKGs), which combine federated learning principles with generative knowledge graph construction. FGKGs can unify distributed data sources across telecom operators while respecting privacy constraints, offering a global, yet secure perspective on fraud ecosystems. This approach supports real-time updates, proactive anomaly detection, and decentralised model evolution.

Furthermore, future research should explore dynamic knowledge embedding, self-updating reasoning agents, and human-in-the-loop architectures to ensure that GenAI remains aligned with expert expectations and evolving fraud landscapes. Integrating these intelligent systems into risk-aware environments will be key to unlocking their full potential.

In summary, future research should pursue the convergence of GenAI, agentic systems, and federated semantic infrastructures. Together, these paradigms offer a vision of telecom fraud detection that is proactive, explainable, collaborative, and robust to rapid change.

Looking ahead, a key direction lies in moving from monolithic fraud detection pipelines toward modular agent-based systems, where each agent acts as a specialist—focused on a specific analytical task within a broader, collaborative fraud investigation framework.

This vision sees Generative AI (GenAI) not merely as a passive tool for data augmentation, but as a proactive analytical agent—capable of reasoning, adaptation, and cooperation. Inspired by recent architectures such as ReAct [117] and AgentTuning [66], we propose a structured configuration of domain-specific GenAI agents orchestrated through a summarization mechanism.

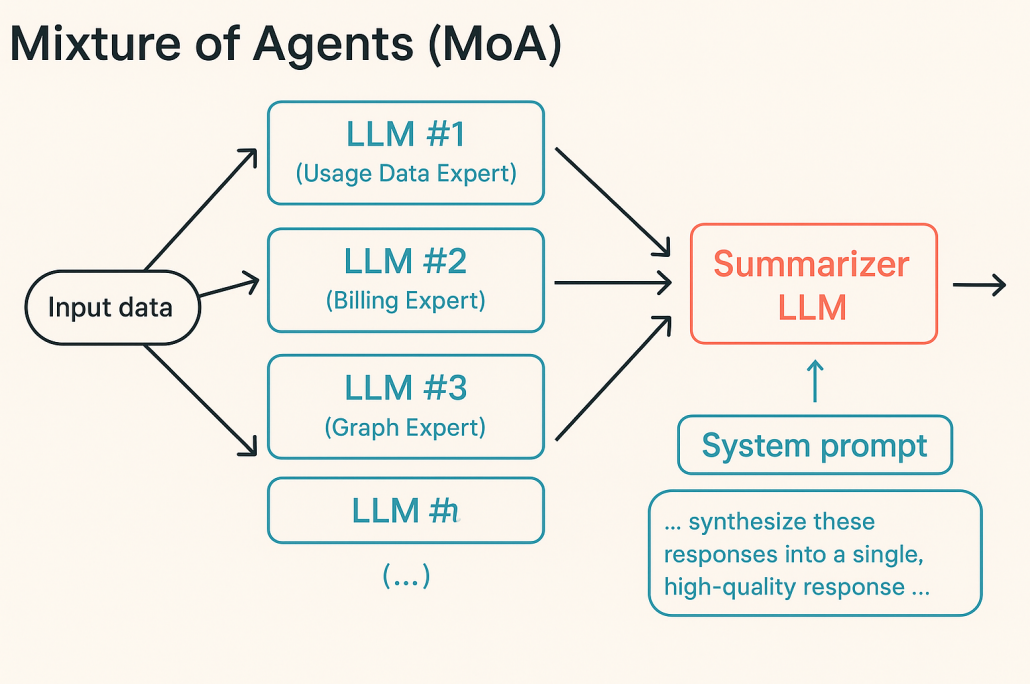



**Figure 8.1:** Mixture of Agents (MoA) Specialized LLM agents.

As illustrated in Figure 8.1, incoming data streams are directed to a set of expert agents each tailored to interpret a specific domain such as usage behaviour, billing anomalies, or graph based relationships. These agents operate in parallel and submit their analyses to a summarization agent, which merges their outputs into a unified, high-confidence response. When coordinated via a Model Context Protocol (MCP), these agentic models can serve in roles such as synthesis of fraud patterns, context sensitive alert triage, and collaborative case

management analyst support.

Paired with domain specific reasoning structures such as knowledge graphs, this architecture enables explainable and semantically rich fraud investigation. Knowledge graphs act as structured representations of telecom entities and behaviours, capturing relationships among users, devices, services, and transactions. Their integration with GenAI unlocks hybrid reasoning pipelines that combine symbolic and statistical inference [50], supporting real-time insight generation across decentralised data landscapes.

Multiple roles can be defined depending on the specific use case. We suggest that these agent roles include:

- **Pattern Synthesizer Agent** — generates and refines candidate fraud typologies from labeled and unlabeled data.
- **Triage Agent** — contextualizes and ranks alerts, prioritizing those that align with evolving threat signatures.
- **Case Management Agent** — summarizes entity behavior, correlates evidence, and recommends analyst actions.
- **Compliance-Aware Agent** — ensures policy and regulatory adherence by validating recommended actions.

To coordinate these agents and ensure consistency, a shared semantic backbone is required. This is enabled via *knowledge graphs*—structured representations of telecom entities, relationships, and behavioural patterns [50]. These graphs provide memory, reasoning grounding, and cross-agent synchronization.

A particularly promising direction involves **Federated Generative Knowledge Graphs (FGKGs)**, which blend federated learning with generative graph construction. FGKGs allow:

- **Privacy-preserving collaboration** across operators,
- **Continuous fraud topology updates** based on learned patterns,
- **Real-time enrichment** of symbolic and neural representations.

Further research should also explore:

- Dynamic knowledge embeddings for real-time updates,
- Self-updating reasoning loops across agents,
- Hybrid neural architectures to balance explainability and learning capacity.

By embedding GenAI into a modular, policy aware architecture, telecom systems can evolve into truly agentic, transparent, and human aligned security ecosystems where collaborative AI agents augment fraud analysts with scalable, explainable, and adaptive reasoning capabilities.

# Bibliography


[1] Martín Abadi Abadi et al. “TensorFlow: A System for Large-Scale Machine Learning.” In: *OSDI*. Vol. 16. 2016, pp. 265–283.

[2] Martín Abadi Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/lite.

[3] Martín Abadi Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/federated.

[4] Nasir Abbas et al. “Mobile Edge Computing: A Survey”. In: *IEEE Internet of Things Journal* 5.1 (2018), pp. 450–465. URL: https://doi.org/10.1109/JIOT.2017.2750180.

[5] Rabiâu ABDULLAHI and Noorhayati Mansor. “Fraud Triangle Theory and Fraud Diamond Theory. Understanding the Convergent and Divergent For Future Research”. In: *International Journal of Academic Research in Accounting, Finance and Management Sciences* 5 (Oct. 2015). DOI: 10.6007/IJARAFMS/v5-i4/1823.

[6] Abuse.ch. *MalwareBazaar Database*. https://bazaar.abuse.ch/browse/tag/flubot/, Last Visit: 2022-08-08. 2022.

[7] Yuan Ai, Mugen Peng, and Kecheng Zhang. “Edge computing technologies for Internet of Things: a primer”. In: *Digital Communications and Networks* 4.2 (2018), pp. 77–86. URL: https://doi.org/10.1016/j.dcan.2017.07.001.

[8] Nida Akhtar et al. “Android malware detection using machine learning techniques”. In: *2017 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2017, pp. 7–13.

[9] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. “oddball: Spotting Anomalies in Weighted Graphs”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Mohammed J. Zaki et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 410–421. ISBN: 978-3-642-13672-6.

[10] Leman Akoglu, Pedro O. S. Vaz de Melo, and Christos Faloutsos. “Quantifying Reciprocity in Large Weighted Communication Networks”. In: *PAKDD (2)*. Vol. 7302. Lecture Notes in Computer Science. Springer, 2012, pp. 85–96.

[11] Leman Akoglu, Hanghang Tong, and Danai Koutra. *Graph-based Anomaly Detection and Description: A Survey*. 2014. arXiv: 1404.4679 [cs.SI].

[12] Masood Akram, Mumtaz Khan, and Muhammad Usman. “Malware detection techniques: A brief review and new perspective”. In: *arXiv preprint arXiv:1307.1506* (2013).

[13] Suzan Almutairi et al. “Hybrid Botnet Detection Based on Host and Network Analysis”. In: *Journal of Computer Networks and Communications* 2020 (2020). ISSN: 2090715X.

[14] Suzana Andova, Holger Hermanns, and Joost-Pieter Katoen. “Discrete-Time Rewards Model-Checked”. In: *Formal Modeling and Analysis of Timed Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 88–104. ISBN: 978-3-540-40903-8.

[15] Paolo Arcaini, Elvinia Riccobene, and Patrizia Scandurra. “Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation”. In: *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. 2015, pp. 13–23. DOI: 10.1109/SEAMS.2015.10.

[16] Frank Bauer and Joseph Lizier. “Identifying influential spreaders and efficiently estimating infection numbers in epidemic models: A walk counting approach”. In: *EPL (Europhysics Letters)* 99 (Mar. 2012). DOI: 10.1209/0295-5075/99/68007.

[17] Andrea Bianco and Luca de Alfaro. “Model Checking of Probabalistic and Nondeterministic Systems”. In: *Foundations of Software Technology and Theoretical Computer Science, 15th Conference, Bangalore, India, December 18-20, 1995, Proceedings*. 1995.

[18] Stephen P. Borgatti and Martin G. Everett. “A Graph-theoretic perspective on centrality”. In: *Social Networks* 28.4 (2006), pp. 466–484. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2005.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0378873305000833.

[19] Francesco Calderoni. “Strategic positioning in mafia networks”. In: *Crime Networks*. Ed. by Carlo Morselli. New York, NY, USA: Routledge, 2014, pp. 163–182.

[20] Javier Cámara et al. “Evaluating Trade-Offs of Human Involvement in Self-Adaptive Systems”. In: *Managing Trade-Offs in Self-Adaptive Systems*. Ed. by Ivan Mistrik et al. Elsevier, Sept. 2016.

[21] Qi Alfred Chen, Zhiyun Qian, and Z. Morley Mao. “Peeking into Your App without Actually Seeing It: UI State Inference and Novel Android Attacks”. In: *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, Aug. 2014, pp. 1037–1052. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/chen.

[22] Shang-Wen Cheng, David Garlan, and Bradley Schmerl. “Architecture-based Self-adaptation in the Presence of Multiple Objectives”. In: *Proceedings of the 2006 International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. http://acme.able.cs.cmu.edu/pubs/uploads/pdf/seams06.pdf. Shanghai, China, May 2006.

[23] Communications Fraud Control Association. *2023 Global Fraud Loss Survey*. Available online: https://www.cfca.org/fraud-loss-survey/. 2023.

[24] Communications Fraud Control Association. *CFCA 2021 Fraud Loss Survey*. Technical Report. Accessed: 2025-03-09. Dec. 2021. URL: https://cfca.org/wp-content/uploads/2021/12/CFCA-Fraud-Loss-Survey-2021-2.pdf.

[25] Communications Fraud Control Association (CFCA). *Fraud Loss Survey*. Version 1.0. 2019. URL: https://cfca.org/wp-content/uploads/2021/02/CFCA-2019-Fraud-Loss-Survey.pdf.

[26] Communications Fraud Control Association (CFCA). *Fraud Loss Survey*. Version 1.0. 2021. URL: https://cfca.org/wp-content/uploads/2021/12/CFCA-Fraud-Loss-Survey-2021-2.pdf.

[27] Corinna Cortes, Daryl Pregibon, and Chris Volinsky. "Communities of Interest". In: *Advances in Intelligent Data Analysis*. Ed. by Frank Hoffmann et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 105–114. ISBN: 978-3-540-44816-7.

[28] Marco Cremonini and Francesca Casamassima. "Controllability of Social Networks and the Strategic Use of Random Information". In: *Computational Social Networks* 4 (Oct. 2017). DOI: 10.1186/s40649-017-0046-2.

[29] Anupam Das et al. "Assisting Users in a World Full of Cameras: A Privacy-Aware Infrastructure for Computer Vision Applications". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1387–1396. URL: https://doi.org/10.1109/CVPRW.2017.181.

[30] Daniela Oliveira Elie Bursztein. *Deconstructing the Phishing Campaigns that Target Gmail Users*. techreport. Aug. 2019. URL: https://www.fastcompany.com/90387855/we-keep-falling-for-phishing-emails-and-google-just-revealed-why.

[31] William Enck et al. "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones". In: *Proceedings of the 9th USENIX conference on Operating systems design and implementation*. 2010, pp. 1–6.

[32] ETSI. *Multi-access Edge Computing (MEC); Framework and Reference Architecture*. Tech. rep. ETSI GS MEC 003 V2.1.1. European Telecommunications Standards Institute (ETSI), 2019. URL: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf.

[33] Romano Fantacci and Benedetta Picano. "Federated learning framework for mobile edge computing networks". In: *CAAI Transactions on Intelligence Technology* 5.1 (2020), pp. 15–21. URL: https://doi.org/10.1049/trit.2019.0049.

[34] Parvez Faruki et al. "Android Security: A Survey of Issues, Malware Penetration, and Defenses". In: *IEEE Communications Surveys and Tutorials* 17.2 (2015), pp. 998–1022. DOI: 10.1109/COMST.2014.2386139.

[35] Parvez Faruki et al. "Andromaly: a behavioral malware detection framework for android devices". In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. 2013, pp. 180–187.

[36] Princy Faruki et al. "AndroSimilar: Robust signature for detecting variations of android malware". In: *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE. 2015, pp. 1–6. DOI: 10.1109/CCECE.2015.7129366.

[37] Miroslav Fiedler. “Algebraic connectivity of graphs”. eng. In: *Czechoslovak Mathematical Journal* 23.2 (1973), pp. 298–305. URL: http://eudml.org/doc/12723.

[38] Karl Flinders. *UK consumer trust in banks, retailers and telcos declines as scams increase*. techreport. Sept. 2021. URL: https://www.computerweekly.com/news/252507268/UK-consumer-trust-in-banks-retailers-and-telcos-declines-as-scams-increase.

[39] Ahmed Ghoneim et al. “Medical Image Forgery Detection for Smart Healthcare”. In: *IEEE Communications Magazine* 56.4 (2018), pp. 33–37. URL: https://doi.org/10.1109/MCOM.2018.1700817.

[40] M. Girvan and M. E. J. Newman. “Community structure in social and biological networks”. In: *Proceedings of the National Academy of Sciences* 99.12 (June 2002), pp. 7821–7826. DOI: 10.1073/pnas.122653799. URL: https://doi.org/10.1073%2Fpnas.122653799.

[41] Shi Gu et al. “Controllability of structural brain networks”. In: *Nature Communications* 6 (Oct. 2015), p. 8414. DOI: 10.1038/ncomms9414.

[42] Zhang Hanlin et al. “ScanMe Mobile: A Cloud-Based Android Malware Analysis Service”. In: *Journal of Cybersecurity* 10.2 (2020), pp. 135–148.

[43] E. Harper. *Exodus Spyware Targets iPhones & Android Phonesto Collect All of Your Personal Data*. online. Oct. 2019. URL: https://www.techlicious.com/blog/exodus-spyware-android-iphone/#google_vignette.

[44] Ryo Hasegawa and Hitoshi Iyatomi. “A Lightweight Android Malware Detection Method using One-Dimensional Convolutional Neural Network”. In: *Proceedings of the 13th International Conference on Signal-Image Technology & Internet-Based Systems*. ACM. 2018, pp. 54–61.

[45] Ying He et al. “Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach”. In: *IEEE Communications Magazine* 55.12 (2017), pp. 31–37. DOI: 10.1109/MCOM.2017.1700246. URL: https://doi.org/10.1109/MCOM.2017.1700246.

[46] Bryan Hooi et al. “FRAUDAR: Bounding Graph Fraud in the Face of Camouflage”. In: *KDD*. ACM, 2016, pp. 895–904.

[47] Mianning Hu et al. “A Framework for Analyzing Fraud Risk Warning and Interference Effects by Fusing Multivariate Heterogeneous Data: A Bayesian Belief Network”. In: *Entropy* 25.6 (2023). ISSN: 1099-4300. DOI: 10.3390/e25060892. URL: https://www.mdpi.com/1099-4300/25/6/892.

[48] Shin-Yuan Hung, David C. Yen, and Hsiu-Yu Wang. “Applying data mining to telecom churn management”. In: *Expert Systems with Applications* 31.3 (2006), pp. 515–524. URL: https://doi.org/10.1016/j.eswa.2005.09.080.

[49] IEEE. “IEEE 5G and Beyond Technology Roadmap”. In: (2017). URL: https://futurenetworks.ieee.org/images/files/pdf/ieee-5g-roadmap-white-paper.pdf.

[50] Shaoxiong Ji et al. “A survey on knowledge graphs: Representation, acquisition and applications”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (2021), pp. 494–514.

[51] Meng Jiang et al. “CatchSync: catching synchronized behavior in large directed graphs”. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’14. New York, New York, USA: Association for Computing Machinery, 2014, pp. 941–950. ISBN: 9781450329569. DOI: 10.1145/2623330.2623632. URL: https://doi.org/10.1145/2623330.2623632.

[52] Kapersky. *Mobile malware evolution 2021*. techreport. Feb. 2022. URL: https://securelist.com/mobile-malware-evolution-2021/105876/.

[53] Rasha Kassem and Andrew Higson. “The New Fraud Triangle Model”. In: *Journal of Emerging Trends in Economics and Management Studies* 3 (June 2012).

[54] Koodous. *Collective Intelligence Against Android Malware*. https://koodous.com/, Last Visit: 2022-08-08. 2022.

[55] Valdis Krebs. “Uncloaking Terrorist Networks”. In: *First Monday* 7.4 (Apr. 2002). DOI: 10.5210/fm.v7i4.941. URL: https://journals.uic.edu/ojs/index.php/fm/article/view/941.

[56] M. Kwiatkowska, G. Norman, and D. Parker. “PRISM 4.0: Verification of Probabilistic Real-time Systems”. In: *Proc. 23rd International Conference on Computer Aided Verification (CAV’11)*. Ed. by G. Gopalakrishnan and S. Qadeer. Vol. 6806. LNCS. Springer, 2011, pp. 585–591.

[57] Arkose Labs. “2023 Cybercrime Prevention Playbook”. In: (2023). URL: https://www.arkoselabs.com/resource/2023-cybercrime-prevention-playbook.

[58] Glenn Lawyer. “Understanding the influence of all nodes in a network”. In: *Scientific reports* 5 (Mar. 2015), p. 8665. DOI: 10.1038/srep08665.

[59] Jia Li et al. “Attributed Network Embedding for Learning in a Dynamic Environment”. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press. 2017, pp. 422–428. URL: https://arxiv.org/abs/1706.01860.

[60] Jiaxi Li et al. “HRGCN: Heterogeneous Graph-level Anomaly Detection with Hierarchical Relation-augmented Graph Neural Networks”. In: *DSAA’ 2023: 10th IEEE International Conference on Data Science and Advanced Analytics*. 2023, pp. 1–10. DOI: 10.1109/DSAA60987.2023.10302626. URL: https://hdl.handle.net/10453/176090.

[61] Jing Li and Han Jin. “Android Malware Detection Based on Feature Codes”. In: *Symmetry* 11.10 (2019), p. 1302. DOI: 10.3390/sym11101302.

[62] Jundong Li, Leman Akoglu, et al. “A Comprehensive Survey on Graph Anomaly Detection with Deep Learning”. In: *arXiv preprint arXiv:2106.07178* (2021).

[63] Jundong Li et al. “Change Point Detection in Dynamic Networks”. In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM. 2016, pp. 210–218.

[64] Yanfang Li and Zhigang Jin. “Android malware detection using machine learning techniques”. In: *2018 15th International Computer Conference on Wavelet Active Media*

*Technology and Information Processing (ICCWAMTIP)*. IEEE. 2018, pp. 131–134. DOI: 10.1109/ICCWAMTIP.2018.8481227.

[65] Zhongmou Li, Hui Xiong, and Yanchi Liu. "Detecting Blackholes and Volcanoes in Directed Networks". In: (May 2010).

[66] Shuohang Liu et al. "AgentTuning: Teaching LLMs to plan and act". In: *arXiv preprint arXiv:2402.05670* (2024).

[67] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-Laszlo Barabasi. "Controllability of complex networks". In: *Nature* 473 (May 2011), pp. 167–73. DOI: 10.1038/nature10011.

[68] Zhe-Ming Lu and Xin-Feng Li. "Attack Vulnerability of Network Controllability". In: *PloS one* 11 (Sept. 2016), e0162289. DOI: 10.1371/journal.pone.0162289.

[69] Rushil Mallarapu. https://github.com/sudo-rushil/dgaintel, Last Visit: 2022-08-08. 2020.

[70] Emaad A. Manzoor et al. *Fast Memory-efficient Anomaly Detection in Streaming Heterogeneous Graphs*. 2016. arXiv: 1602.04844 [cs.SI].

[71] Carlo Morselli. "Assessing Vulnerable and Strategic Positions in a Criminal Network". In: *Journal of Contemporary Criminal Justice* 26.4 (2010), pp. 382–392. DOI: 10.1177/1043986210377105. eprint: https://doi.org/10.1177/1043986210377105. URL: https://doi.org/10.1177/1043986210377105.

[72] Cooperation Group on Network and Information Security. *Cybersecurity of 5G networks - EU Toolbox of risk mitigating measures*. techreport. Jan. 2020. URL: https://www.politico.eu/wp-content/uploads/2020/01/POLITICO-Cybersecurity-of-5G-networks-EU-Toolbox-January-29-2020.pdf.

[73] paloalto Networks. *Stop Attackers from Using DNS Against You*. techreport. Apr. 2022. URL: https://www.paloaltonetworks.com/resources/whitepapers/stop-attackers-from-using-dns-against-you.

[74] Caleb C. Noble and Diane J. Cook. "Graph-based anomaly detection". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: Association for Computing Machinery, 2003, pp. 631–636. ISBN: 1581137370. DOI: 10.1145/956750.956831. URL: https://doi.org/10.1145/956750.956831.

[75] Stuart Oldham et al. "Consistency and differences between centrality measures across distinct classes of networks". In: *PLOS One* 14.7 (July 2019). Ed. by Satoru Hayasaka, e0220061. DOI: 10.1371/journal.pone.0220061. URL: https://doi.org/10.1371.

[76] Opensignal. *5G users on average consume up to 2.7x more mobile data compared to 4G users*. consultation. Oct. 2020. URL: https://www.opensignal.com/2020/10/21/5g-users-on-average-consume-up-to-27x-more-mobile-data-compared-to-4g-users.

[77] Seyed Ali Ossia et al. "Private and Scalable Personal Data Analytics Using Hybrid Edge-to-Cloud Deep Learning". In: *Computer* 51.5 (2018), pp. 42–49. URL: https://doi.org/10.1109/MC.2018.2381113.

[78] Hamed Haddad Pajouh et al. “A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks”. In: *IEEE Transactions on Emerging Topics in Computing* 7.2 (2019), pp. 314–323. URL: https://doi.org/10.1109/TETC.2016.2633228.

[79] D. Palmer. *Smartphone Malware is on the Rise, Here’S Whatto Watch Out for. ZDNET*. online. July 2022. URL: https://www.zdnet.com/article/smartphone-malware-is-on-the-rise-heres-what-to-watch-out-for.

[80] JongChan Park et al. “Fraud Detection with Multi-Modal Attention and Correspondence Learning”. In: Jan. 2019, pp. 1–7. DOI: 10.23919/ELINFOCOM.2019.8706354.

[81] Ami Pedahzur and Arie Perliger. “The Changing Nature of Suicide Attacks: A Social Network Perspective”. In: *Social Forces* 84.4 (June 2006), pp. 1987–2008. ISSN: 0037-7732. DOI: 10.1353/sof.2006.0104. eprint: https://academic.oup.com/sf/article-pdf/84/4/1987/6520845/84-4-1987.pdf. URL: https://doi.org/10.1353/sof.2006.0104.

[82] Yulong Pei, Tianjin Huang, and Werner van Ipenburg. “ResGCN: Attention-based Deep Residual Modeling for Anomaly Detection on Attributed Networks”. In: *Proceedings of the 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2021. DOI: 10.1109/DSAA53316.2021.9564225. URL: https://arxiv.org/abs/2009.14738.

[83] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “DeepWalk: Online Learning of Social Representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2014, pp. 701–710.

[84] B. Aditya Prakash et al. “EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Mohammed J. Zaki et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 435–448. ISBN: 978-3-642-13672-6.

[85] TM Forum - Security & Privacy Project. “GB954 Fraud Classification Guide v2.4”. In: *TM Forum Best Practices* (Aug. 2013).

[86] Murali Ramanathan et al. “Bridging Centrality: Identifying Bridging Nodes in Scale-free Networks”. In: *Bridging Centrality: a Concept and Formula to Identify Bridging Nodes in Scale-free Networks*. 2006.

[87] Jianji Ren et al. “Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things”. In: *IEEE Access* 7 (2019), pp. 69194–69201. URL: https://doi.org/10.1109/ACCESS.2019.2919736.

[88] Tiago Gama Rodrigues et al. “A PSO model with VM migration and transmission power control for low Service Delay in the multiple cloudlets ECC scenario”. In: *IEEE International Conference on Communications (ICC)*. 2017, pp. 1–6. URL: https://doi.org/10.1109/ICC.2017.7996358.

[89] Josep A. Rodríguez and José A Rodrigués. “The March 11 th Terrorist Network: In its weakness lies its strength”. In: *CiteSeerX* (2005). DOI: 10.1.1.98.4408.

[90] Ryan A Rossi et al. “Fast maximum clique algorithms for large graphs”. In: *Proceedings of the 21st international conference companion on World Wide Web*. ACM. 2012, pp. 365–366.

[91] RSA. *RSA Quarterly Fraud Report Volume 3, Issue 2 Q2 2020*. techreport. Oct. 2020. URL: https://www.infopoint-security.de/media/RSA_Fraud_Report_Q2_2020.pdf.

[92] Oleg Rybakov et al. “Streaming Keyword Spotting on Mobile Devices”. In: *Interspeech 2020*. ISCA, Oct. 2020.

[93] Chuan Shi et al. “HeteSim: A general framework for relevance measure in heterogeneous networks”. In: *IEEE Transactions on Knowledge and Data Engineering* 26.10 (2014), pp. 2479–2492.

[94] Kijung Shin, Bryan Hooi, and Christos Faloutsos. “M-Zoom: Fast Dense-Block Detection in Tensors with Quality Guarantees”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 264–280. DOI: 10.1007/978-3-319-46128-1_17. URL: http://dx.doi.org/10.1007/978-3-319-46128-1_17.

[95] Mile Sikic et al. “Epidemic centrality — is there an underestimated epidemic impact of network peripheral nodes?” In: *The European Physical Journal B* 86 (2013), pp. 1–13.

[96] Monetary Authority of Singapore. *Consultation Paper on Proposed Shared Responsibility Framework*. consultation. Oct. 2023. URL: https://www.mas.gov.sg/publications/consultations/2023/consultation-paper-on-proposed-shared-responsibility-framework.

[97] Manmeet Singh, Maninder Singh, and Sanmeet Kaur. “Issues and challenges in DNS based botnet detection: A survey”. In: *Computers & Security* 86 (2019), pp. 28–52. ISSN: 0167-4048.

[98] Jiasong Song et al. “Andromaly: Large scale android malware detection through static analysis”. In: *Acm Transactions on Intelligent Systems and Technology (TIST)* 5.4 (2014), p. 61.

[99] Jimeng Sun et al. “GraphScope: parameter-free mining of large time-evolving graphs”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 687–696. ISBN: 9781595936097. DOI: 10.1145/1281192.1281266. URL: https://doi.org/10.1145/1281192.1281266.

[100] Yizhou Sun et al. “PathSim: Meta path-based top-k similarity search in heterogeneous information networks”. In: *Proceedings of the VLDB Endowment* 4.11 (2011), pp. 992–1003.

[101] Muhammad Talha and Khaled Salah. “Detecting malicious android applications using permissions and api calls”. In: *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE. 2016, pp. 92–97.

[102] Hacker Target. https://hackertarget.com/top-million-site-list-download/, Last visit 2021-09-08. July 2020.

[103] SciKit Team. https://scikit-learn.org/stable/, Last Visit: 2022-08-08. 2007.

[104] Wenqiang Tian and Kevin Lin. “Chapter 2 - Requirements and scenarios of 5G system”. In: *5G NR and Enhancements*. Ed. by Jia Shen et al. Elsevier, 2022, pp. 41–52. ISBN: 978-0-323-91060-6.

[105] Saverio Trotta et al. “2.3 SOLI: A Tiny Device for a New Human Machine Interface”. In: *2021 IEEE International Solid- State Circuits Conference (ISSCC)*. Vol. 64. 2021, pp. 42–44.

[106] Véronique Van Vlasselaer et al. “Guilt-by-Constellation: Fraud Detection by Suspicious Clique Memberships”. In: 2015 (Mar. 2015), pp. 918–927. DOI: 10.1109/HICSS.2015.114.

[107] Bingbo Wang, Lin Gao, and Yong Gao. “Control range: a controllability-based index for node significance in directed networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2012.04 (Apr. 2012), P04011. DOI: 10.1088/1742-5468/2012/04/P04011. URL: https://dx.doi.org/10.1088/1742-5468/2012/04/P04011.

[108] Kuochen Wang et al. “Behavior-based botnet detection in parallel”. In: *Security and Communication Networks* 7.11 (2014), pp. 1849–1859.

[109] Shiqiang Wang et al. “Adaptive Federated Learning in Resource Constrained Edge Computing Systems”. In: *IEEE Journal on Selected Areas in Communications* 37.6 (2019), pp. 1205–1221. URL: https://doi.org/10.1109/JSAC.2019.2904348.

[110] Wei Wang et al. “BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors”. In: *Information Sciences* 511 (2020), pp. 284–296. ISSN: 00200255.

[111] Majda Wazzan et al. “Internet of things botnet detection approaches: Analysis and recommendations for future research”. In: *Applied Sciences (Switzerland)* 11.12 (2021). ISSN: 20763417.

[112] Dong-Jie Wu et al. “DroidMat: Android Malware Detection through Manifest and API Calls Tracing”. In: *2012 Seventh Asia Joint Conference on Information Security*. IEEE, 2012, pp. 62–69. DOI: 10.1109/AsiaJCIS.2012.18. URL: https://ieeexplore.ieee.org/document/6298136.

[113] Sheng Wu et al. “Permission-based android malware detection using feature selection”. In: *2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*. IEEE. 2016, pp. 201–205.

[114] Ying Xing et al. “Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation”. In: *Mathematical Problems in Engineering* 2021 (2021). Ed. by Jude Hemanth, p. 6640499. ISSN: 1024-123X.

[115] Jiawei Xu et al. “AddGraph: Anomaly detection in dynamic attributed graphs”. In: *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM. 2018, pp. 171–179.

[116] Elena Wu-Yan et al. “Benchmarking Measures of Network Controllability on Canonical Graph Models”. In: *Journal of Nonlinear Science* 30 (Oct. 2020). DOI: 10.1007/s00332-018-9448-z.

[117] Shinn Yao et al. “ReAct: Synergizing reasoning and acting in language models”. In: *arXiv preprint arXiv:2210.03629* (2023).

[118] Yunfan Ye et al. “EdgeFed: Optimized Federated Learning Based on Edge Computing”. In: *IEEE Access* 8 (2020), pp. 209191–209198. URL: https://doi.org/10.1109/ACCESS.2020.3038287.

[119] Suleiman Yerima, Madge Moh, and Nura Turemiren. “A novel ensemble learning approach for android malware detection”. In: *2019 International Conference on Computing, Networking and Informatics (ICCNI)*. IEEE. 2019, pp. 1–6. DOI: 10.1109/ICCNI.2019.8785631.

[120] Shuai Yu et al. “When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network”. In: *IEEE Internet of Things Journal* 8.4 (2021), pp. 2238–2251. URL: https://doi.org/10.1109/JIOT.2020.3026589.

[121] Wenchao Yu et al. “NetWalk: A Flexible Deep Embedding Approach for Anomaly Detection in Dynamic Networks”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2018, pp. 2672–2681. DOI: 10.1145/3219819.3220024. URL: https://dl.acm.org/doi/10.1145/3219819.3220024.

[122] Zhengxin Yu et al. “Federated Learning Based Proactive Content Caching in Edge Computing”. In: *IEEE Global Communications Conference (GLOBECOM)*. 2018, pp. 1–6. URL: https://doi.org/10.1109/GLOCOM.2018.8647616.

[123] Xiao Yuan, Peng Lu, and Xiaoguang Li. “Deep learning for android malware detection using dynamic analysis”. In: *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE. 2017, pp. 50–54. DOI: 10.1109/CSE-EUC.2017.102.

[124] Pengju Zhang, Quan Liu, and Zheng Zhao. “Artificial Immune System-Based Approach for Android Malware Detection”. In: *IEEE Access* 6 (2018), pp. 27834–27847.

[125] Xizhe Zhang, Jianfei Han, and Weixiong Zhang. “An efficient algorithm for finding all possible input nodes for controlling complex networks”. In: *Scientific Reports* 7 (Sept. 2017), p. 10677. DOI: 10.1038/s41598-017-10744-w.

[126] Panpan Zheng et al. “One-Class Adversarial Nets for Fraud Detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 1286–1293. DOI: 10.1609/aaai.v33i01.33011286. URL: https://ojs.aaai.org/index.php/AAAI/article/view/3924.

[127] Yajin Zhou and Xuxian Jiang. “Dissecting Android Malware: Characterization and Evolution”. In: *2012 IEEE Symposium on Security and Privacy*. 2012, pp. 95–109. DOI: 10.1109/SP.2012.16.