KTH ROYAL INSTITUTE
OF TECHNOLOGY

Doctoral Thesis in Information and Communication Technology

Towards Decentralized Graph Learning

LODOVICO GIARETTA

Stockholm, Sweden 2023

Towards Decentralized Graph Learning

LODOVICO GIARETTA

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Friday the 9th June 2023, at 9:00 a.m. in Sal-C, Kistagången 16, Stockholm

Doctoral Thesis in Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden 2023

ISBN 978-91-8040-584-3
TRITA-EECS-AVL-2023:42

Printed by: Universitetsservice US-AB, Sweden 2023

Abstract

Current Machine Learning (ML) approaches typically present either a centralized or federated architecture. However, these architectures cannot easily keep up with some of the challenges introduced by recent trends, such as the growth in the number of IoT devices, increasing awareness about the privacy and security implications of extensive data collection, and the rise of graph-structured data and Graph Representation Learning. Systems based on either direct data collection or Federated Learning contain centralized, privileged systems that may act as scalability bottlenecks and dangerous single points of failure, while requiring users to trust the privacy protections and security practices in place. The combination of these issues ultimately leads to data waste, as opportunities to extract insights from available data are missed and thus the full societal benefits of advanced data analytics and ML are not realized.

In this thesis, we argue for a paradigm shift towards a completely decentralized and trustless architecture for privacy-aware Graph Representation Learning, which employs Gossip Learning and other gossip-based peer-to-peer techniques to achieve high levels of scalability and resilience while reducing the risk of privacy leaks. We then identify and pursue three key research directions necessary to achieve our vision: lifting unrealistic assumptions on Gossip Learning, identifying and developing specific use cases that are enabled or improved by gossip-based decentralization, and overcoming the obstacles to the deployment of decentralized training and inference for Graph Representation Learning models.

Based on these key directions, our contributions are as follows. First, we analyze the robustness of Gossip Learning when several unrealistic but often assumed conditions are lifted. Then, we exploit Gossip Learning and gossip-based peer-to-peer protocols more in general across three use cases: the collaborative training of differentially-private Naive Bayes classifiers across organizations holding sensitive user data; the construction of decentralized, privacy-preserving data marketplaces; and the development and decentralization of early-stage IoT botnet detection systems based on Graph Representation Learning. Finally, we introduce a general framework for the fully-decentralized training of Graph Neural Networks, overcoming the typical requirement of these models to access non-local information during training and inference.

The combination of these contributions removes major roadblocks towards decentralized graph learning, and also opens a new research direction aimed at further developing and optimizing the fully-decentralized training of Graph Representation Learning models.

Sammanfattning

Dagens metoder för maskininlärning (ML) har vanligtvis antingen en centraliserad eller federerad arkitektur. Dessa arkitekturer kan dock inte lätt hålla jämna steg med några av de utmaningar som introducerats av de senaste trenderna, som till exempel ökningen av antalet IoT-enheter, ökad medvetenhet om integritets- och säkerhetskonsekvenserna av omfattande datainsamling samt ökningen av grafstrukturerad data och Graph Representation Learning. System baserade på antingen direkt datainsamling eller federerad inlärning innehåller centraliserade, privilegierade system som kan vara flaskhalsar och riskerar bli kritiska sårbarhetspunkter. Samtidigt måste användarna lita på integritetsskyddet och säkerhetspraxis som finns. Kombinationen av dessa problem leder i slutändan till ett ineffektivt nyttjande av data, eftersom möjligheter att utvinna insikter från tillgänglig data inte utnyttjas och därmed inte realiserar de fulla samhällsnyttorna som är möjliga med avancerad dataanalys och ML.

I denna avhandling argumenterar vi för ett paradigmskifte mot en helt decentraliserad och tillitslös arkitektur för integritetsmedveten Graph Representation Learning, som använder Gossip Learning och andra gossip-baserade peer-to-peer-tekniker för att uppnå höga nivåer av skalbarhet och motståndskraft, samtidigt som den minskar risken för integritetsläckor. Vi identifierar och driver sedan tre viktiga forskningsinriktningar som är nödvändiga för att uppnå vår vision; att lyfta orealistiska antaganden om Gossip Learning, identifiera och utveckla specifika användningsfall som möjliggörs eller förbättras av gossip-baserad decentralisering, samt övervinna hindren för utplacering av decentraliserad utbildning och inferens för Graph Representation Learning modeller.

Baserat på dessa nyckelriktlinjer våra bidrag är följande. Först analyserar vi robustheten i Gossip Learning när flera orealistiska men ofta antagna villkor upphävs. Vi utnyttjar sedan Gossip Learning och gossip-baserade peer-to-peer-protokoll mer generellt i tre användningsfall: kollaborativ inlärning av differentiellt privata Naive Bayes-klassificerare över entiteter med känslig användardata; byggandet av decentraliserade datamarknadsplatser som bevarar integriteten; samt utveckling och decentralisering av IoT-botnätdetekteringssystem i ett tidigt skede baserade på Graph Representation Learning. Slutligen introducerar vi ett allmänt ramverk för helt decentraliserad utbildning av Graph Neural Networks, som eliminerar de typiska kraven för dessa modeller för att få tillgång till icke-lokal information under träning och inferens.

Kombinationen av dessa bidrag tar bort stora hinder mot decentraliserad grafinlärning, och öppnar också en ny forskningsriktning som syftar till att vidareutveckla och optimera den helt decentraliserade utbildningen av Graph Representation Learning modeller.

Acknowledgements

The journey of a doctoral student is always challenging, full of successes and failures, excitements and disappointments, both academically and personally. It certainly was for me, and I am deeply grateful to all the people who helped and supported me along this journey.

First, I would like to thank my main supervisor, Assoc. Prof. Šarūnas Girdzijauskas, who guided me through this journey. In many long conversations across these four years, you always supported my work and directed me on the correct path, helping me grow as a researcher.

I would then like to thank all the colleagues with whom I had the opportunity to collaborate and discuss my work over the years. In particular, Filip Cornell, Susanna Pozzoli and Ahmed Emad from KTH; Thomas Marchioro from FORTH Institute and Ahmed Lekssays from Università dell'Insubria. Thank you for sharing the doctoral journey with me, exchanging invaluable knowledge, and spending a lot of good time together.

I would like to use this occasion to also acknowledge the support I received from the Marie-Curie ITN project *RAIS: Real-time Analytics for the Internet of Sports* (grant number 813162), part of the European Union's Horizon 2020 research and innovation program. Being part of this project gave me the opportunity to share experiences and knowledge with researchers from different backgrounds, exposing me to different perspectives and enriching my journey.

One person without whose support I wouldn't have been able to complete this journey is my wife, Pei Zhang. Your constant help and encouragement motivate me to always push forward, especially in challenging times, and I am deeply grateful to have you by my side every day.

Even from Italy, my family, and in particular my parents, grandparents, and brother, has been following my progress and offering their support. Thank you for always thinking about me and providing words of encouragement in moments of need.

Finally, I am grateful to all my friends who, although scattered across the continent, are always close to me. Thank you for sharing, through frequent conversations, the joys and struggles of this journey.

List of appended papers

Paper A

L. Giaretta and Š. Girdzijauskas.
"GOSSIP LEARNING: OFF THE BEATEN PATH".
2019 IEEE International Conference on Big Data (Big Data).
IEEE, 2019, pp. 1117-1124.
Accepted 2019-10-17

Paper B

T. Marchioro, L. Giaretta, E. Markatos and Š. Girdzijauskas.
"FEDERATED NAIVE BAYES UNDER DIFFERENTIAL PRIVACY".
19th International Conference on Security and Cryptography (SECRYPT).
Scitepress, 2022, pp. 170-180.
Accepted 2022-05-01

Paper C

L. Giaretta, T.Marchioro, E. Markatos and Š. Girdzijauskas.
"TOWARDS A REALISTIC DECENTRALIZED NAIVE BAYES WITH DIFFERENTIAL PRIVACY".
[under review] Springer Lecture Notes in Computer Science (LNCS) Transactions, 2023.

Paper D

L. Giaretta, I. Savvidis, T. Marchioro, Š. Girdzijauskas, G. Pallis, M. D. Dikaiakos and E. Markatos.
"PDS^2: A USER-CENTERED DECENTRALIZED MARKETPLACE FOR PRIVACY PRESERVING DATA PROCESSING".
IEEE 37th International Conference on Data Engineering Workshops (ICDEW).
IEEE, 2021, pp. 92-99.
Accepted 2021-02-22

Paper E

L. Giaretta, T.Marchioro, E. Markatos and Š. Girdzijauskas.
"TOWARDS A DECENTRALIZED INFRASTRUCTURE FOR DATA MARKETPLACES: NARROWING THE GAP BETWEEN ACADEMIA AND INDUSTRY".
Proceedings of the 1st International Workshop on Data Economy.
ACM, 2022, pp. 49-56.
Accepted 2022-10-20

Paper F

L. Giaretta, A. Lekssays, B. Carminati, E. Ferrari and Š. Girdzijauskas.
"LIMNET: EARLY-STAGE DETECTION OF IOT BOTNETS WITH LIGHTWEIGHT MEMORY NETWORKS".
Computer Security-ESORICS 2021: 26th European Symposium on Research in Computer Security, Proceedings, Part I.
Springer, 2021, pp. 605-625.
Accepted 2021-07-19

Paper G

L. Giaretta, A. Lekssays, B. Carminati, E. Ferrari and Š. Girdzijauskas.
"METASOMA: DECENTRALIZED AND COLLABORATIVE EARLY-STAGE DETECTION OF IOT BOTNETS".
[under review] IEEE Internet of Things Journal (IOTJ), 2023.

Paper H

L. Giaretta and Š. Girdzijauskas
"FULLY-DECENTRALIZED TRAINING OF GNNS USING LAYER-WISE SELF-SUPERVISION".
[under review] European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2023.

Contents

Chapter 1

Introduction

Thanks to an increasingly fast pace of digitalization, our society now constantly produces vast troves of data, monitoring many of its aspects, from individual behaviours, to industrial operations, to environmental observations. At the same time, modern semiconductor fabrication techniques have provided us with a previously unthinkable amount of computational power at a relatively low cost. These events set the foundation for a number of research breakthroughs, which led to the flourishing, both in academia and industry, of Big Data processing and Machine Learning techniques.

These *data-driven* techniques are becoming, year after year, more and more important components of our society. They play a key role in an ever-growing number of areas, from how we interact with and consume both online and physical content, to how companies forecast future scenarios and optimize their infrastructure and operations, to how faults or malicious behaviour can be detected and addressed in real-time in a variety of domains. Big Data processing [1], [2] and Machine Learning (ML) [3], [4] allow us to both improve existing services and provide new, innovative ones, which were deemed impossible just a few years ago, boosting the overall well-being of our society. It is therefore imperative for both academia and industry to push forward the research on these techniques, in order to maximize their *sustainable exploitation* to the benefit of all.

Currently, the vast majority of Big Data processing and Machine Learning approaches present a *centralized* architecture, in which the raw data produced by all sources is transferred to a central location (either "the Cloud" or an on-prem datacenter), where it is stored and processed as needed to provide relevant services. This paradigm has been steadily developed and optimized over many years, resulting in a variety of well-established and highly-effective tools and frameworks, such as Apache Flink[1] for streaming data processing and

[1] `https://flink.apache.org/`, accessed 2023-04-17

Pytorch Distributed[2] for large-scale ML training. However, a number of trends have developed in recent years, which create new opportunities for our society, but also highlight *intrinsic limitations* in the centralized paradigm and lead to challenges that threaten its long-term sustainability and call for a *paradigm shift* in this area.

One such trend is the growing deployment of edge and **Internet of Things** (IoT) devices [5], [6]: from personal gadgets like smartphones and smartwatches, to "smart city" and "smart factory" appliances like sensors, cameras, robots and vehicles, these connected devices are forecast to be soon embedded in all aspects of our society. This leads to a growing number of data sources, characterized not only by their large aggregate *volume*, but also by their great *variety* and *velocity* [2], [7], [8]. In turn, this leads to **scalability** issues for centralized solutions: the network bandwidth and computing power required to process communications from IoT devices grow linearly with their number and with the frequency of their updates. Even when technically possible, building, maintaining and scaling centralized Big Data and ML solutions may be financially infeasible, especially when the data is deemed to provide limited or unknown value. Thus, a "chicken and egg" problem may arise, where data collection is only justifiable financially when it leads to significant business value, but the potential business value cannot be easily evaluated, or even discovered, until a significant amount of data has been collected and analyzed.

The issue of scalability is also closely connected to that of *reliability*. Any architecture relying on a central component presents a **single point of failure**. As long as that component is controlled by a single entity, its size or level of internal redundancy hardly matters: an individual server can easily stop working due to a hardware fault; a configuration error can take down an entire datacenter and even a globally-distributed cloud system[3]; a company may take the business decision to completely shut down a key service[4].

Another key trend encompasses the interlinked domains of **data privacy**, **security** and **trust**. As the data being collected covers more and more aspects of our daily lives while at the same time becoming ever more granular, these aspects gain increasing relevance. In some cases, organizations collect sensitive data about their users. In other cases, while the data may not directly appear to be sensitive, advanced analysis techniques and the cross-matching of separate datasets may enable the extraction of sensitive information. Due to the *intangible nature of data*, once an entity has obtained direct access to a dataset, it is nearly impossible to track or limit its usage and dissemination. Combined with

[2] `https://pytorch.org/tutorials/beginner/dist_overview.html`, accessed 2023-04-17

[3] `https://www.bleepingcomputer.com/news/technology/massive-cloudflare-outage-caused-by-network-configuration-error/`, accessed 2023-03-13

[4] `https://thestack.technology/google-cloud-iot-core-retired-killed-by-google/`, accessed 2023-03-13

the recent proliferation of the Data Economy[5] – in which organizations buy and sell data in order to improve their service offering or maximize the return of their data collection efforts, respectively – this leads to significant risks for the users every time they agree to data collection activities.

Whenever a single entity is responsible for the storage and/or processing of a large amount of data, users need to trust its intentions. Our society has already witnessed situations[6,7] in which an organization was found to have used the data at its disposal in ways that the public at large deemed to be "unethical" and a breach, if not of the explicit legal contract in place, at least of the implicit social one. At the same time, users also need to trust the security practices of the entity: we are experiencing more and more frequent cyberattacks[8] against key organizations, that often include data exfiltration efforts, with the risk of sensitive data being leaked online, or secretly sold to malicious entities. As a response to these issues of privacy, security and trust, regulators in various jurisdictions are putting in place more and more stringent statutes on how potentially-sensitive data should be handled.

A third key trend is the growing production of **graph-structured data**: from social media networks, to road traffic patterns, to molecular structures, a significant portion of the data we capture includes not only individual data points with their features, but also the various kinds of relations and interactions between them. However, traditional ML approaches are not capable of modelling these links, and thus miss the crucial information that they contain. This has led to the development of Graph Representation Learning (GRL) [9], a branch of ML that focuses on extracting and processing the key phenomena hidden in graph-structured data, and presenting them in a format amenable to traditional ML workloads. In recent years, Graph Neural Networks (GNNs) [10], a family of GRL techniques based on deep learning, have emerged as the state of the art in this area. Unfortunately, most GRL techniques – mirroring the overall ML trend – are based on a centralized architecture, where the complete graph to be analyzed is stored by a single entity, exacerbating all the issues described above.

Thus, given these trends, it is clear that centralized Big Data processing and ML architectures cannot maximize societal benefits in a *sustainable* way. From the end user perspective, they bring reliability and privacy risks, and require high levels of trust towards the data collectors. From the perspective of the organization deploying these techniques, they introduce significant financial,

[5] https://digital-strategy.ec.europa.eu/en/library/communication-building-european-data-economy, accessed 2023-03-15

[6] https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_data_scandal, accessed 2023-03-10

[7] https://en.wikipedia.org/wiki/Project_Nightingale, accessed 2023-03-10

[8] https://blog.checkpoint.com/2023/01/05/38-increase-in-2022-global-cyberattacks/, accessed 2023-03-10

technical and legal burdens. The combination of these issues inevitably leads to **data waste**: the vast majority of the data being produced every day is not collected nor processed, and thus our society risks missing out on significant benefits that may arise from the exploitation of these data.

Both the academic and industrial communities have identified these issues, and significant efforts have been dedicated to addressing them, across a variety of research areas. For example, best practices for maintaining scalable and reliable cloud infrastructure are better understood and more widely implemented[9]. However, scalable centralized processing techniques cannot match the growth in data, and reliability issues can manifest at different levels and are ultimately intrinsic to any system controlled by a single entity. On another front, several techniques in the area of privacy-preserving computing have gained significant traction, such as Trusted Execution Environments [11], which can prevent direct data exfiltration and remove the need to trust the data-processing entity. But these techniques cannot be used in isolation, as they do not address the challenge of reliably operating on high volumes of data from a large number of sources. Thus, specific approaches for *massively-distributed* Big Data and ML processing are necessary, on top of which privacy-preserving techniques can be integrated.

The most well-known massively-distributed ML approach is Federated Learning [12], which increases scalability by removing the need to transfer raw data from the source devices to the central location. Instead, each device performs a portion of the necessary computations locally, submitting its partial result to a central aggregator, which then computes a final result. In addition to better scalability, this approach can also improve privacy by keeping the raw data on each device secret, and can be combined with privacy-preserving techniques for stronger guarantees [13]. However, Federated Learning is still based on a centralized architecture, and therefore still presents the same intrinsic limitations discussed above. Its scalability, while significantly better than a traditional approach, is still limited by the capabilities of the central aggregator, which also acts as a single point of failure, raising reliability concerns. The central aggregator is also a potential source of trust issues, due to its privileged status in the architecture. Thus, the concerns described above still apply to Federated Learning scenarios.

In this thesis, we instead argue for a paradigm shift towards a *truly decentralized* architecture for data processing, focusing specifically on ML workloads. By using existing *peer-to-peer gossip communication protocols* [14], and in particular **Gossip Learning** [15], the IoT devices become responsible for the entire computation and aggregation process, without the need for any central entity for data collection, aggregation or even coordination. With this architecture,

[9] `https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html`, accessed 2023-04-17

all the issues described above can be more easily tackled. A high degree of scalability is naturally achieved by the system, as every new data source brings not only more data to process, but also more computational power and network bandwidth to perform the decentralized processing and aggregation. As all IoT devices are equal peers in the system, no single point of failure exists, and no privileged entity has full control of the system or needs to be trusted with a higher degree of responsibility than others. Furthermore, because the raw data never needs to leave the source devices, a basic level of data protection is immediately achieved, which can be further enhanced by the addition of privacy-preservation techniques. Finally, new value-creating processing tasks can be trialled without any upfront investment for a centralized infrastructure, as both storage and processing are fully handled by the peer-to-peer network.

We also incorporate GRL techniques – and in particular GNNs – in our decentralized ML architecture, not only due to their general pervasiveness in modern ML, but also due to the particular synergy with decentralized data sources: personal connected devices like smartphones and smartwatches are uniquely suited to capture our position within social graphs; smart sensors spread across a city can be mapped to road connectivity and capture traffic flows; and a decentralized network of connected devices is itself a graph that can be analyzed. In employing GRL techniques, we remain true to our decentralized paradigm, and thus require each device in our architecture to be only responsible for, and aware of, its own position and interactions within the structure of the graph.

Overall, the combination of decentralized, gossip-based learning and GRL leads to our vision for tackling the challenges and trends of modern ML and maximising the sustainable exploitation of our growing data sources:

Vision *A completely decentralized and trustless architecture for privacy-aware Graph Representation Learning applications, in which the use of local computations and gossip-based peer-to-peer communications guarantees virtually unlimited scalability, and where each device only maintains a local view of its surroundings, with raw data never spreading through the network.*

Figure 1.1 contrasts a traditional centralized GRL architecture with our decentralized vision.

Unfortunately, **three challenges** exist that must be addressed before our vision can be achieved, and which are described in detail in section 1.1. **First**, most existing studies on Gossip Learning fail to consider the heterogeneity and complexity of real-world decentralized environments, and thus present unrealistic assumptions and limitations. **Second**, Gossip Learning shows low penetration in the field of applied ML and there appears to be a general lack of awareness in the academic and industrial communities regarding the benefit, chal-

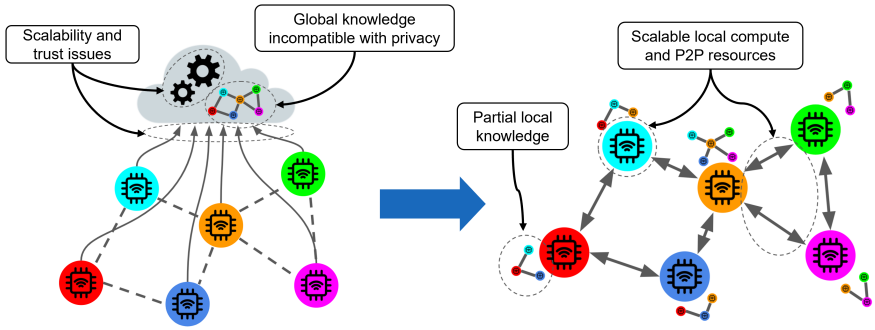


Figure 1.1: Our vision: moving from a centralized architecture to a decentralized one, with scalable peer-to-peer computation and partial local knowledge.

lenges and applicability scope of gossip-based decentralization. **Third**, Graph Representation Learning presents unique decentralization challenges compared to more traditional ML approaches, preventing the direct conversion of existing centralized solutions.

Given these challenges, we set three objectives for this thesis, which we detail in section 1.2: 1) identifying and addressing the unrealistic assumptions often seen in pre-existing Gossip Learning research, 2) proposing specific use cases where Gossip Learning, or gossip-based decentralization more in general can be employed to materialize the key advantages of decentralization, and 3) apply gossip-based decentralization to the training and inference processes of GRL-based models.

Our contributions towards these objectives, listed in section 1.3, lead us to summarize this thesis as follows:

Thesis Statement *Gossip Learning can be adapted to robustly withstand challenging real-world conditions and can be successfully employed in a variety of use cases, such as to decentralize collaborative training on sensitive user data, to build decentralized privacy-preserving data marketplaces, and to enable lightweight, decentralized early-stage botnet detection on industrial IoT devices. Finally, Gossip Learning plays a key role in enabling our vision of achieving fully-decentralized training and inference of Graph Representation Learning models based exclusively on local knowledge.*

1.1 Challenges

Gossip Learning [15] is the key enabling technology that represents the backbone of the decentralized architecture we envision. It is a general framework, based on gossip communication protocols [14], to perform iterative training of ML models in a decentralized setting, where each participating device owns a private set of data that cannot be shared with others. Due to its nature, it is often compared to Federated Learning [12], a framework that performs the same task of iterative training on multiple private data sources, but which employs a central aggregator instead of gossip communications to merge the contributions from individual devices.

However, despite their similarities, Gossip Learning and Federated Learning have received very different amounts of interest in the academic and industrial communities. The latter, publicly backed by large corporations[10,11], has been the focus of significant academic research and has been studied and applied in a variety of different use cases. Thanks to these efforts, many of the challenges in applying Federated Learning are now well understood, and techniques have been developed to address them. These include, among others, dealing with restricted computational and networking capabilities of edge devices [16], learning from heterogeneous data distributions [17], withstanding malicious behaviours from some devices [18] and preventing information leakage [13].

On the other hand, despite showing performance comparable to Federated Learning [19], **Gossip Learning has received significantly less attention** by the broader community. Because of this, its performance characteristics and challenges are not well understood. In fact, very few works have methodically analyzed the potential effects of real-world conditions such as restricted communication capabilities, heterogeneous device speeds or non-IID data distributions. Similarly, works exploring byzantine fault tolerance or strict privacy guarantees have been sporadic and limited in their scope. On the other hand, a large number of studies on Gossip Learning present very strong assumptions in all or several of these aspects, and thus their results risk being deemed unrealistic. Therefore, it is necessary to provide complete, methodical analyses of how various conditions can affect Gossip Learning in a wide variety of settings, thus providing a solid foundation for its application in real-world environments.

The seemingly limited awareness about Gossip Learning in the wider community also leads to significantly **lower penetration of the technique in the area of *applied* ML**. When experts from different domains seek to apply ML techniques to their specific tasks, and assess that they need data-private training

[10] `https://ai.googleblog.com/2017/04/federated-learning-collaborative.html` accessed 2023-03-13

[11] `https://www.microsoft.com/en-us/research/blog/flute-a-scalable-federated-learning-simulation-platform/` accessed 2023-03-13

across a number of entities, Gossip Learning is rarely considered, with Federated Learning attracting the vast majority of studies. This also extends to situations where complete decentralization may be beneficial, as can be inferred by the proliferation of studies seeking to achieve "Decentralized Federated Learning", often by employing complex (relative to Gossip Learning) protocols such as blockchain-based consensus, smart contracts or decentralized file systems [20], [21]. It is therefore necessary to increase awareness about Gossip Learning and gossip-based protocols more in general, so that the research community may consider them as an alternative for decentralized data-private applications and may investigate their suitability in different domains and use cases. The goal of increasing awareness may be achieved by spearheading a search and analysis of relevant use cases, thus providing useful starting points that can be extended to further domains.

The limited understanding of Gossip Learning notwithstanding, the immediate implementation of our vision is also hampered by **the challenge of decentralizing existing GRL techniques**, and in particular GNNs. The vast majority of works on this topic have considered a traditional centralized setting, where the complete training graph is fully contained within either an individual machine or a single cloud infrastructure (e.g. a datacenter). A few studies [22]–[24] have investigated various kinds of federated settings for GNN training, however, they are at odds with our vision in two aspects. First, being federated approaches, they require a central aggregator and are therefore not truly decentralized. Second, they assume that each device stores a large enough subgraph to be able to perform a complete forward and backward step based exclusively on information stored on the device. As the forward (and therefore backward) pass of an L-layers GNN requires the input graph to include the L-hop neighbourhood of each node for which an output should be produced, ensuring that each device has access to all relevant inputs would not be scalable to deep GNNs and would violate our vision, requiring the storage of *non-local* information. Thus, it is necessary to develop techniques to allow fully-decentralized training of GNNs, overcoming the challenge posed by the wide *receptive fields* of deep GNNs while only requiring access to local information.

However, *training* GRL models is only half of the challenge. Once a model is trained, it needs to be deployed for *inference*. While this process is much lighter and does not require backward passes, it still presents the same receptive field issue, as high-quality representations of each node can only be built by taking into account interactions within a large, multi-hop neighbourhood. In some applications, the same solutions developed for centralized training may be reused. But there might also be situations where training can be performed in a traditional, centralized setting, using centrally-available or open-access data, and only the inference itself needs to be performed in a purely decentralized, data-private setting. As such, lightweight decentralized inference techniques

may be developed for these scenarios.

1.2 Research Objectives

In this thesis, we set out to address the limitations of existing Gossip Learning literature and the challenges in decentralizing GRL training and inference, highlighted in section 1.1, with the overall purpose of removing any roadblocks to the implementation and deployment of the vision that we laid out.

We identify 3 key objectives to achieve our purpose.

O1 Our first objective is to identify and address some of the unrealistic assumptions that are often present in previous Gossip Learning studies. Most of these assumptions relate to tangible characteristics of the deployment environment, such as limited computational and bandwidth capabilities of edge devices, restricted communication channels, and non-IID data distributions. Others, however, relate to intangible stakeholder requirements, such as the need for certain degrees of privacy protection. Our approach is to methodically analyze and gradually lift these assumptions, thus removing the artificial limits that they pose to the range of scenarios where Gossip Learning can be applied. Our objective is to quantitatively evaluate how lifting these limitations affects the performance characteristics of Gossip Learning, identify the thresholds after which Gossip Learning ceases to work effectively and, where possible, present protocol variants that further extend its applicable boundaries.

O2 Our second objective is to propose specific use cases where Gossip Learning, or gossip-based protocols more in general, can be employed in place of centralized or federated solutions, identifying and exploiting the key advantages of decentralization. These use cases should cover different domains and applications and thus act as starting points to increase awareness about Gossip Learning and enable its spreading in the wider community. Within the scope of this objective, we therefore identify 3 promising use cases, and expand each into a dedicated sub-objective.

O2.1 Our first use case concerns collaborative training of Naive Bayes classifiers [25] on highly-sensitive user data collected by a number of large and medium organizations. An example of this could be a consortium of health institutions employing ML to improve their diagnosing tools, utilizing medical records from previous and current patients. In this context, the organizations themselves do not require any protection for their overall data distributions, but due

to the sensitive nature of individual data points, it must be impossible to infer the identity of individual users within each organization. As such, Differential Privacy [26] is employed locally at each institution. Our objective is to quantitatively evaluate federated and decentralized training setups for this use case. Particular focus is given to the impact of different phenomena that may arise in a realistic environment, such as different numbers of organizations, different numbers of users at each organization and varying privacy considerations across the organizations.

O2.2 Our second use case concerns the domain of decentralized, privacy-preserving data marketplaces. Data marketplaces are growing in importance, as they improve the exploitation of existing data, allowing the flow of information and business value between those with the means to collect data and those with the business ideas to exploit them. However, traditional marketplaces suffer from severe usability issues, including scalability and data discovery, and social issues, such as the lack of privacy protections and equitable wealth distribution. Decentralized, privacy-preserving data marketplaces attempt to address these issues. Our main objective is to show that Gossip Learning, and more in general gossip-based protocols, can be effectively used to build data marketplace architectures that satisfy all the requirements and address all the challenges mentioned. However, due to the relative infancy of this topic, its interdisciplinary nature, its wide scope and its complex set of stakeholders, a broader analysis is required. We therefore expand our objective to also provide a detailed analysis of the different stakeholders in this field, their often conflicting requirements, the various computer science domains that are relevant to the development of decentralized, data-private marketplaces and the most promising techniques from each of these fields, along with the many remaining open questions in the area.

O2.3 Our third use case comes from the domain of IoT security. In recent years, mirroring the growth in IoT devices, our society has witnessed a sharp increase in the number of IoT botnet attacks. These types of malware first stealthily spread across vulnerable IoT devices and then, once reached a sufficient "mass", order all infected devices to attack a victim system, causing a Distributed Denial of Service (DDoS) attack. As a response, security researchers have employed a variety of tools, including ML, to quickly identify and stop botnet attacks. However, fewer efforts have been dedicated to the detection of IoT botnets during their spreading phase, before any attack is launched. Our main objective is to demonstrate the effectiveness of a decentralized early-stage botnet detection approach, based on

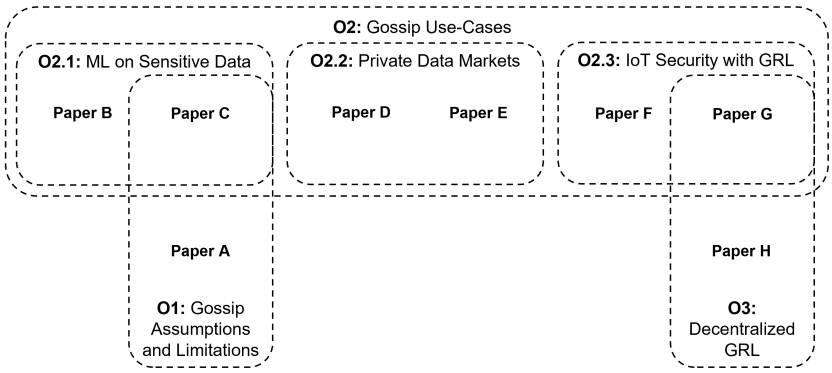


Figure 1.2: Overview of the overlapping research objectives set out in this thesis (see section 1.2), and of how each included paper (see section 1.4) contributes to them.

collaborative inference across IoT devices powered by gossip-based information sharing. But, to achieve this core objective, we first need to develop a centralized technique that is fast, lightweight and extremely accurate, which we can then extend to a decentralized setting.

O3 Our third objective is to combine our advanced study of Gossip Learning, and gossip-based protocols more in general, with GRL techniques, in order to achieve fully-decentralized training and inference at scale on private, local data. Thus, achieving this third objective means enabling the implementation of our vision. More specifically, this objective encompasses two distinct aspects. The first aspect consists of the development of a general framework for the fully-decentralized training of deep GNNs based only on local knowledge. The objective is to demonstrate the feasibility of such a framework and open the road to future, in-depth analysis and deployment of its techniques. The second aspect focuses on decentralized inference, and in particular on those scenarios where decentralized training is not necessary, and where thus a simpler and lighter inference solution may be achievable, compared to the general framework just described. We identify the scenario described in objective **O2.3** as a perfect example of this, and our objective is thus to evaluate the inference performance of decentralized vs centralized GRL in that context.

1.3 Thesis Contributions

The contributions of this thesis are collected in 8 papers, listed in section 1.4. Each of the (sub-)objectives presented in section 1.2 is addressed by two papers, with some papers existing at the overlap of multiple (sub-)objectives. Figure 1.2 summarizes the objectives, highlighting which papers contribute to each and what overlaps exist.

Overall, the contributions of this thesis can be summarized as follows.

- In **Paper A** [27], we study some of the key assumptions present in most pre-existing work on Gossip Learning. We identify three key axes on which these assumptions sit: the data distribution axis, ranging from the typical assumption of fully-distributed data (one data point per node), to non-IID distributed (different numbers of data points per node and/or different distributions of data points at each node); the device speed axis, affected by both the computation and communication capabilities of each device, ranging from the typical assumption of equal speeds to heavily-skewed speeds; and the network topology axis, ranging from the typical assumption of full connectivity (any-to-any communications) to very restrictive topologies, with almost isolated communities and/or power law degree distributions. Our results show that Gossip Learning is quite robust to most of these factors, and thus most assumptions can be *individually* lifted without issues. However, we show that when non-IID data distributions are correlated with either skewed device speeds or restricted topologies, Gossip Learning may present slow convergence, and may even converge to biased models, that do not truly represent the overall data distribution. We identify the root causes of these biases and implement two variants of Gossip Learning to address them. One employs gossip caching to prevent model bias caused by skewed, data-correlated device speeds. Another de-biases the implicit random walks performed by gossiped models in order to prevent model bias towards the data stored in high-degree nodes in restricted network topologies. Thus, **Paper A** [27] provides a significant contribution towards objective **O1**.

- We perform an extensive quantitative analysis of the scenario presented in objective **O2.1**. In **Paper B** [28], we develop a federated, single-step aggregation scheme to train Naive Bayes classifiers on highly-sensitive user data stored by a consortium of medium to large organizations. Our scheme employs Differential Privacy, applied locally by each institution before sending its contribution to the central aggregator, in order to hide the identity of individual users whose data is being exploited by the model. We provide an empirical evaluation of our approach, comparing it with a centralized, differentially-private variant and with a centralized, non-

private baseline. Our results show that the federated approach is highly competitive with the centralized, differentially-private one. Furthermore, we highlight how, due to the different privacy sensitivities of numerical and categorical features, a federated aggregation across a large number of organizations may even outperform a centralized one when numerical features are dominant in a dataset, while categorical ones favor centrally-applied differential privacy. In **Paper C** [29], we extend our previous work in two directions. First, we replace our federated, single-step aggregation scheme with a completely decentralized, iterative one based on gossip communication protocols. We show that the decentralized approach quickly converges to the same results as the federated one, thus negating any benefit of the latter. We also focus on a more realistic analysis of privacy budgets and dataset sizes across organizations. We show that our federated and decentralized solutions are resilient to the existence of very different dataset sizes across organizations, and even to different privacy budgets being set by each organization when applying differential privacy. Thus, by ensuring that realistic privacy conditions do not hamper the decentralized training process, our work in this area also contributes to objective **O1**.

- We provide a detailed review and analysis of the decentralized, privacy-preserving data marketplaces domain introduced in objective **O2.2**. In **Paper D** [30], we perform an in-depth analysis of the key stakeholders and conflicting requirements in such a marketplace. We then develop a modular, highly-flexible, user-centered architecture for a decentralized, privacy-preserving data marketplace. We review a number of state-of-the-art techniques and tools in the areas of blockchain, privacy-preserving computing, and decentralized ML, identifying Ethereum, Trusted Execution Environments (TEEs) and Gossip Learning as the most suitable approaches to implement the components of our modular architecture. Finally, we review several key open questions in the area, including reward distribution schemes, data authenticity checks, data discovery and filtering tools, and advanced privacy protections. In **Paper E** [31], we provide a critical look at the current state of industrial and academic progress in the area of data marketplaces. Specifically, we identify a significant gap between academic research and industrial applications, which we analyze with the goal of identifying key academic advancements that can be feasibly ported to the industry in the short term, to bridge this gap. This leads to a re-evaluation of blockchain technologies, privacy-preserving computation techniques and data valuation approaches.

- In **Paper F** [32], we develop a lightweight deep learning model for early-stage detection of IoT botnets, employing GRL techniques and a *memory*

network architecture. In our approach, a central monitoring system collects headers for all packets exchanged in an industrial IoT environment and feeds them to a pair of mutually-recurrent RNN cells to build and update device "memories" (i.e. embeddings) that can then be used by down-stream classifiers to perform both device-level and packet-level detection of botnets. Our results show that our approach not only provides significantly higher detection accuracy than existing recurrent models for this early-stage detection, but is also orders of magnitude smaller and much faster during inference. Building on these characteristics, in **Paper G** [33], we extend our previous solution to remove the need for a central monitoring system to perform the inference task. Instead, each IoT device is responsible for monitoring only its local traffic and building partial memories. These are then gossiped and merged across different devices to achieve global knowledge. Thus, this line of work not only presents a third practical use case for Gossip Learning (**O2.3**), but also demonstrates an effective decentralized GRL inference approach (**O3**), in a scenario where decentralized training is not necessary.

- In **Paper H** [34], we develop and evaluate a general framework for fully-decentralized training of GNN models, thus completing objective **O3**. Our solution combines several techniques – including decoupled layer-wise training, self supervision, gossip-based negative sampling and Gossip Learning – to ensure training can be performed even though each device is only aware of its direct neighbours and is not aware of the raw features, nor the edges, of any other node in the training graph. Our results show that our solution is close to the performance of a centralized model with full global knowledge, and is thus a feasible alternative in scenarios where decentralization and privacy are key factors. Furthermore, we highlight a wide variety of research questions that are opened by these promising results, and we thus trace an interesting and novel research direction that we hope will be followed by several future studies.

1.4 List of Included Papers

This thesis is supported by the following 8 papers.

Paper A **L. Giaretta** and Š. Girdzijauskas, *Gossip Learning: Off the Beaten Path*, IEEE BigData 2019

Contribution: The author of this thesis implemented all the required code, performed all the experimental analysis, designed the proposed bias mitigations and contributed a majority of the written paper.

Paper B T. Marchioro, **L. Giaretta** et al., *Federated Naive Bayes Under Differential Privacy*, SECRYPT 2022

Contribution: The author of this thesis implemented a majority of the required code, run and collected results for a majority of the experiments, and was responsible for the analysis of the impact of local sensitivity of numerical features on the accuracy of the model. The paper was written jointly by T. Marchioro and the author of this thesis, with both actively participating and reviewing each other's work during all phases of the research.

Paper C **L. Giaretta**, T. Marchioro, et al., *Towards a Realistic Decentralized Naive Bayes with Differential Privacy*, **under review** in Springer LNCS Transactions

Contribution: The author of this thesis designed, implemented and collected evaluation results for the gossip-based implementation of the model, and implemented, evaluated and analyzed the model behaviour with varying dataset sizes and privacy budgets. The paper was written jointly by T. Marchioro and the author of this thesis, with both actively participating and reviewing each other's work during all phases of the research.

Paper D **L. Giaretta**, I. Savvidis, T. Marchioro, et al., *PDS^2: A User-Centered Decentralized Marketplace for Privacy Preserving Data Processing*, ICDEW 2021

Contribution: The author of this thesis provided the original vision that led to this paper, performed the bulk of the stakeholders and requirements analysis and led the overall architectural design, which was jointly performed with the other authors. The author of this thesis proposed and developed the requirement of flexible ownership boundaries for end users and the concept of decentralized executors. The author of this thesis performed the literature review and analysis of techniques in the areas of decentralized ML, data discovery and filtering, and privacy leaks. The paper was written jointly by I. Savvidis, T. Marchioro and the author of this thesis, with the three actively participating and reviewing each other's work during all phases of the research.

Paper E **L. Giaretta**, T. Marchioro, et al., *Towards a Decentralized Infrastructure for Data Marketplaces: Narrowing the Gap between Academia and Industry*, Data Economy 2022

Contribution: The author of this thesis performed the analysis of the openness and transparency challenges in data marketplaces, including the analysis of blockchain solution and of data discovery and standardization,

and provided and equal contribution with T. Marchioro to the analysis of the challenge of data valuation. The paper was written jointly by T. Marchioro and the author of this thesis, with both actively participating and reviewing each other's work during all phases of the research.

Paper F **L. Giaretta**, A. Lekssays, et al., *LiMNet: Early-Stage Detection of IoT Botnets with Lightweight Memory Networks*, ESORICS 2021

Contribution: The author of this thesis designed, implemented and evaluated all aspects of the proposed ML architecture. The paper was written jointly by A. Lekssays and the author of this thesis, with both actively participating and reviewing each other's work during all phases of the research.

Paper G **L. Giaretta** and Š. Girdzijauskas, *Metasoma: Decentralized and Collaborative Early-Stage Detection of IoT Botnets*, **under review** in IEEE IoTJ

Contribution: The author of this thesis proposed the idea of decentralized early-stage IoT detection, designed the additional ML components necessary and implemented and evaluated the ML performance of the system. The author of this thesis also performed the part of the security analysis related to potential memory forgery, designing, implementing and evaluating the countermeasures based on forgery detection. The paper was written jointly by A. Lekssays and the author of this thesis, with both actively participating and reviewing each other's work during all phases of the research.

Paper H **L. Giaretta** and Š. Girdzijauskas, *Fully-Decentralized Training of GNNs*, **under review** in ECML-PKDD 2023

Contribution: The author of this thesis developed the idea of decentralized GNN training, identified all key challenges and relevant techniques, implemented all the required code, performed all the experimental analysis and contributed a majority of the written paper.

1.5 List of Excluded Papers

The work performed by the author while pursuing this thesis is not limited to the work presented in the included papers (see section 1.4), but also comprises participation in the development of additional manuscripts and publications, which are *not* included in support of this thesis, but are listed as follows.

- D. Garcia Bernal, **L. Giaretta** and Š. Girdzijauskas, *Federated Word2Vec: Leveraging Federated Learning to Encourage Collaborative Representation Learning*, ArXiV 2021

Contribution: The author of this thesis supervised a student working on their Master's Degree projects, focusing on the federated training of language models. The author of this thesis participated in shaping the goals of the work and helped and guided the student from the initial literature review, to the implementation, evaluation and writing of the final Master's Degree project report and of the resulting manuscript listed here.

- A. A. Alkathiri, **L. Giaretta** and Š. Girdzijauskas, *Decentralized Word2Vec using Gossip Learning*, NoDaLiDa 2021

  Contribution: The author of this thesis supervised a student working on their Master's Degree projects, focusing on the decentralized training of language models. The author of this thesis participated in shaping the goals of the work and helped and guided the student from the initial literature review, to the implementation, evaluation and writing of the final Master's Degree project report and of the resulting paper listed here.

- A. E. Samy, **L. Giaretta**, Z. T. Kefato and Š. Girdzijauskas, *SchemaWalk: Schema Aware Random Walks for Heterogeneous Graph Embedding*, WWW Companion 2022

  Contribution: The author of this thesis implemented part of the required code, collected results for part of the experiments, provided guidance and feedback during the implementation and analysis of the proposed method, and participated in drafting the paper.

1.6 Outline

The rest of this thesis is organized as follows. In chapter 2 we provide background information relevant to the contents of the thesis. In chapter 3 we expand on section 1.3 with an extended summary of the papers included in this thesis. In chapter 4 we describe potential future work and provide concluding remarks. Afterwards, the complete publications listed in section 1.4 are appended.

Chapter 2

Background

In this chapter, we briefly introduce some of the key topics and technologies that are mentioned multiple times throughout this thesis. First, we describe Federated and Gossip Learning, two techniques for massively-distributed ML on private data. We then introduce the general area of Graph Representation Learning (GRL), with a particular focus on Graph Neural Networks (GNNs) and dynamic graphs. Finally, we discuss several techniques that can provide strong data privacy guarantees for ML training scenarios.

2.1 Massively-Distributed ML on Private Data

Recent years have seen a quick growth not only in the number of connected devices, but also in their computational capabilities. This is especially visible in the smartphone market, with new devices including powerful CPUs and even AI accelerators. At the same time, as already discussed in chapter 1, the data collected by these devices have become more and more granular and sensitive.

This has led to the development of various techniques to train ML models directly on the edge devices, therefore achieving two goals: 1) exploiting the computational capabilities of the devices to reduce the load on the central infrastructure and increase scalability and 2) providing a certain degree of privacy, as the raw data is not disclosed by the devices, which only share model updates computed on the local data.

However, it must be noted that – despite providing an important, straightforward, and sometimes sufficient first line of protection – sharing local updates instead of the raw data does not completely prevent a determined attacker from extracting sensitive information from that data. Even without seeing the raw data, an attacker with access to the individual updates shared by a device may obtain significant insights about the data stored in that device [35]. And, even when access to the individual updates is removed, an attacker that can

access the weights of the trained model, or even just query it, may be able to identify individual devices that participated in its training, and even extract certain information about their raw private data [36]. Therefore, it is often necessary to compound massively-distributed data-private ML techniques with other privacy-preservation approaches, such as the ones that will be introduced in section 2.3.

We use the term *massively-distributed data-private techniques* to collectively refer not only to multiple different techniques, but also to multiple possible scenarios where these techniques can be applied. Sometimes, the term *decentralized ML* is also employed to collectively refer to all techniques in this area, but it is important to note that not all techniques are truly decentralized, in that some of them employ centralized orchestration and/or aggregation. Thus, the use of the term most likely derives from the fact that these techniques are often used to operate on *decentralized data*, that is, data that is not owned nor controlled by a single entity, but rather by a number of different entities.

Compared to centralized data, decentralized data comes with a number of challenges: while the former can be shuffled across storage nodes – all owned by a single entity – to ensure IID distribution, the latter cannot, thus leading to data imbalances. Furthermore, decentralized data is stored across a number of devices, or more in general systems, which may have different performance, networking and reliability characteristics, and which are typically spread geographically and connected via the public Internet. Thus, massively-distributed ML techniques must account for these aspects, accommodating for heterogeneous environments and lifting many of the reliability and performance assumptions that hold in traditional, datacenter-scale distributed learning.

However, it is possible to further subdivide the concept of decentralized data into two distinct scenarios: the *federated* and the *massively decentralized* scenarios. In the former, the data is controlled by a relatively small number of medium to large organizations, each of which owns a large number of data points with significant variety. These organizations store and process the data on capable and reliable dedicated hardware. Thus, this scenario is closer to the traditional centralized one. On the other hand, the massively decentralized scenario is more extreme. The number of devices can reach millions, each owned by a separate individual, with very limited and biased data. Furthermore, each device may be highly unreliable and may only be available intermittently.

We now introduce the two most important techniques for massively-distributed ML on private data: Federated Learning [12] and Gossip Learning [15] The former is a centralized approach, while the latter is completely decentralized. However, both have been successfully applied to both federated and massively decentralized data sources.

2.1.1 Federated Learning

Federated Learning [12] is the most popular technique in this area. It is based on the well-known *parameter server* architecture often employed in traditional, datacenter-scale data-parallel ML training, but adapts it to the less controlled environment that we just described. In Federated Learning, a central entity, logically separate from all data sources, acts as coordinator and aggregator for the entire system and controls the authoritative copy of the model. In each iteration of the training process, this central entity broadcasts the current authoritative model weights to each participant, or to a subset of them selected to be active in this iteration. Each of the chosen participants will then perform a local training step, which typically consists of computing the gradient of the received model on the local private data. The computed gradients are then submitted to the central entity, which aggregates all of them (or all those received within a pre-specified time window) and proceeds to update the authoritative copy of the model. Then, the process is repeated for the following iteration.

As mentioned in section 1.1, Federated Learning has received a significant amount of attention in recent years, and can now be considered a well-established and well-understood technique. Each of the challenges brought by the decentralized data setting has been extensively analyzed, and numerous extensions to the basic protocol just described have been proposed to improve its behaviour in complex realistic environments. This includes dealing with the limited networking and computational capabilities of many edge devices [16] and learning from non-IID data distribution across the participants [17].

As the popularity of Federated Learning grows, so does the risk that malicious actors, aware of its characteristics, attempt to exploit it or circumvent the weak privacy protections provided by the base protocol. As such, a significant amount of work has been devoted to improving the security and privacy characteristics of Federated Learning. Privacy-preserving techniques have been applied to prevent information leakage, from the individual model updates and/or from the final trained model [13], [37]. At the same time, byzantine fault-tolerance approaches have been utilized [18] to ensure that malicious participants cannot prevent or slow down the convergence of the model, or bias its predictions to render it unreliable.

In this thesis: **Paper B** [28] introduces a federated, differentially-private Naive Bayes classifier. Federated Learning is also discussed in the context of decentralized, privacy-preserving data marketplaces in **Paper D** [30] and **Paper E** [31].

Algorithm 1 Skeleton of the Gossip Learning protocol

```
procedure MAIN
    currentModel ← INITMODEL()
    lastModel ← currentModel
    loop
        WAIT(Δ)
        p ← RANDOMPEER()
        SEND(p, currentModel)
    end loop
end procedure
procedure ONMODELRECEIVED(m)
    mergedModel ← MERGE(m, lastModel)
    currentModel ← UPDATE(mergedModel, localData)
    lastModel ← m
end procedure
```

2.1.2 Gossip Learning

Gossip Learning [15] approaches the problem of training on private data from a different angle: that of *peer-to-peer gossip communication protocols* [14]. Gossip protocols are a broad family of techniques that allow a network of devices to efficiently spread and aggregate information until convergence is reached, at which point all devices share the same, complete knowledge of the information [38]. They achieve this using *asynchronous*, peer-to-peer communications, without the need for any central orchestration.

More specifically, each device participating in Gossip Learning, at regular intervals, shares its current local model with a randomly-selected peer. At any point in time, upon receiving a model from a peer, each device merges the newly-received model with the last one that was previously received, trains the combined model on its local data, and stores the resulting model as its new current model, which it will gossip at the next occasion. Algorithm 1 captures the high-level steps performed by each participating device.

It is interesting to also look at this process from the perspective of the models, rather than the devices. Each model can be described as performing a *random walk* over the set of devices. At each step of the walk, the model not only learns about the information stored in that device (through the local training step), but also receives all the knowledge that was collected by the previous "visitor" of the device (through the model merging step). Intuitively, this leads to each device effectively "doubling" the number of data points it has been trained on at each step of the random walk, thus explaining the fast convergence rate of Gossip Learning protocols.

While Gossip Learning has been shown to be an effective alternative to Federated Learning [19], it has unfortunately not received the same attention

as the latter, especially with regards to exploring its robustness to the challenges of massively-distributed settings, as will be extensively discussed in section 3.1.

In this thesis: Many of the assumptions and limitations of Gossip Learning are discussed in **Paper A** [27]. Gossip Learning and/or gossip-based protocols are employed throughout this thesis to build decentralized, differentially-private Naive Bayes classifiers (**Paper C** [29]), as building blocks for decentralized, privacy-preserving data marketplaces (**Paper D** [30] and **Paper E** [31]), to decentralize an early-stage detector for IoT botnets (**Paper G** [33]) and to build a framework for fully decentralized training of GNNs (**Paper H** [34]).

2.2 Graph Representation Learning

As discussed in chapter 1, Graph Representation Learning (GRL) [9], [39] is a branch of ML that focuses on extracting information from graph-structured data, exploiting both the structure of the graph and the implicit meaning of the relationships between nodes, and any available node- or edge-level features.

Graph structures show up in a wide variety of scenarios, ranging from social networks and other human interactions, to bonds within molecules and interactions between them [40], to the organization of knowledge bases in terms of relations between concepts [41]. This large variety in the application domains and in the kinds of data being encoded by graphs translates to an equal variety in their characteristics: nodes and edges in a graph may be homogeneous (that is, all represent the same kind of entities and interactions) or heterogeneous; features may be associated with nodes, edges, both or none; the graphs may be static, or may evolve dynamically, with nodes and edges appearing (and, in some cases, disappearing) over time. These are only some of the many axes among which graph structures, and in turn GRL techniques, may be classified [42].

As such, it is not possible to effectively summarize the entirety of this field in a concise manner. We instead choose to focus on those aspects that will be relevant to the papers included in this thesis, and refer the readers to a variety of surveys on the broader topic.

One key concept that exists in the vast majority of GRL approaches is that of *node embeddings* [43] These are low-dimensional vectors, one for each node in the graph, that are often produced by these approaches, either as intermediate results or as final outputs. Node embeddings are typically computed by considering both the features of the nodes and their connections to other nodes in the graph. As such, they perform the key task of encoding the *unstructured* information contained in the edges of the graph into a *structured* embedding space. As such, node embeddings present the graph information in a format

that can be fed into existing ML architectures for tasks such as classification and regression.

2.2.1 Graph Neural Networks

Graph Neural Networks (GNNs) [10] are a broad family of techniques that represent the state of the art in most GRL applications. While not clearly defined, the term GNN typically refers to any deep learning GRL technique based on a message-passing architecture [44], where a new embedding for each node is computed by using a trainable ML component to process "messages" received from its neighbours; the messages themselves are typically produced by passing the existing embeddings of the neighbours and any edge information through a trainable component.

Often, the term GNN is specifically associated with convolutional GNNs [10], which are the most popular and successful sub-family of GNNs. As the name suggests, these can be intuitively seen as performing the same kind of operation as image-based CNNs [45]: embedding each pixel by aggregating its features with those of all the pixels that are close to it. While each layer in a CNN only aggregates information from a small area, stacking them allows the *receptive field* of each pixel to increase layer after layer, thus capturing larger and larger patterns in the input image. Convolutional GNNs perform the same operation, but instead of considering close-by pixels in an image, they consider directly-connected nodes in a graph structure.

However, not all GNN approaches fit into this sub-family. For example, some of the first GNN architectures were based on Recurrent Neural Networks (RNNs) [10]. More recently, some of the approaches proposed for GRL on dynamic graphs have also employed architectures that are not based on traditional ML convolutions [46]–[48].

In this thesis: **Paper H** [34] introduces a decentralized framework for training generic convolutional GNN models. **Paper F** [32] and **Paper G** [33] employ deep GRL techniques on dynamic graphs.

2.2.2 Learning on Dynamic Graphs

Traditionally, GRL techniques have focused on *static* graphs. However, many real-world graphs are in fact dynamic, with new edges and nodes appearing and, in some cases, old ones disappearing or becoming meaningless. For example, new social connections between people are formed every day and old connections, while not suddenly disappearing, may wane if not renewed.

Thus, an increasing amount of attention has been dedicated in recent years to the sub-field of *dynamic* GRL, in which a model is provided temporal in-

formation about the evolution of the graph and can use that to evolve node embeddings over time and solve tasks such as predicting future changes in the graph.

While there is substantial variety among dynamic GRL techniques, two main research directions can be identified, based on different architectures for capturing the changes happening in the graph. The first consists of creating snapshots of the graph structure at different points in time. The advantage of this approach is that a large number of changes can be processed in bulk, and that traditional static GRL techniques can be applied to each snapshot to produce time-specific embeddings, with additional ML components being added to model the changes between snapshots [49], [50]. However, this approach suffers from low granularity: as the entire graph is reprocessed at each snapshot, reducing the time interval between them incurs significant increases of computational requirements.

This issue can be avoided by instead feeding the GRL model with a continuous stream of changes: every time a change (such as the addition of a new edge) happens, the model performs a partial update of the node embeddings, based on the intuition that each individual change only affects a small portion of the nodes that are directly adjacent or otherwise very close to it [46]–[48]. Approaches of this type can detect model changes in the graph structure much quicker, but, due to their highly-sequential nature, are more challenging to train, due to the risk of vanishing or exploding gradients.

In this thesis: A dynamic GRL approach based on analyzing a continuous stream of graph changes is introduced in **Paper F** [32] and adapted to a decentralized environment in **Paper G** [33].

2.3 Privacy-Preserving ML

As already mentioned, data-private ML approaches like Federated and Gossip Learning cannot alone guarantee that information about the participants will not be leaked. To achieve this, it is necessary to combine them with one or more techniques that are able to provide strong, mathematical guarantees. However, it is important to note that information leakages can happen in different phases of the ML lifecycle, and that sensitive knowledge can be extracted from different types of processed data. As such, no single solution can provide all-around protection from all kinds of leakages. Instead, it is necessary to identify, on a case-by-case basis, what information is sensitive and which part of the ML lifecycle leak that information, so that the correct combination of approaches can be used to achieve the necessary privacy guarantees.

2.3.1 Blind Computation Techniques

In a massively-distributed setting, not only in the context of ML but more in general for data processing workloads, one of the main sources of information leakage consists of the local computation results that need to be exchanged across the participants or with a central aggregator. An attacker with access to this local results can easily extract significant amounts of sensitive knowledge about the data owned by each participant. External attackers can be easily locked out of this knowledge by the use of encryption, but this leaves the door open to malicious, or even just "curious" participants (or, in Federated Learning, central aggregators).

Fortunately, several techniques exist that enable "blind" computing, i.e. the ability to perform computation on data without the ability to read that data, and thus without the possibility of using that data for any activity other than the agreed-upon computation. The three most popular techniques in this domain are Homomorphic Encryption (HE), Trusted Execution Environments (TEEs) and Secure Multi-Party Computation (SMC).

Homomorphic Encryption Homomorphic Encryption schemes [51] are a family of cryptographic approaches that allow specific computations to be performed on encrypted inputs, and produce encrypted outputs, so that the entity performing the computation need not be able to access either inputs or outputs. More formally, given a suitable function f, Homomorphic Encryption allows the construction of a corresponding function f' and encryption/decryption functions enc/dec, such that $dec(f'(enc(x))) = f(x)$. Unfortunately, Homomorphic Encryption schemes typically incur high computational overheads [52] and the set of functions that can be computed exactly is limited. This is particularly problematic in ML settings, as several widely-used non-linear activation functions (such as sigmoid and softmax) can only be approximated, while those based on ordinal operations (such as max pooling) cannot be computed at all [53], [54].

Trusted Execution Environments Trusted Execution Environments [11] are special enclaves within a system, often implemented in hardware, that are isolated from the rest of the system, such that no observer with hardware and/or software access to the system is able to monitor or modify the operations being performed and the data being stored and manipulated within the environment. As such, it is possible to perform blind computation by feeding the TEE with encrypted data and have the TEE decrypt it, perform the necessary calculations and re-encrypt the output before it leaves it [55]. TEEs typically include embedded cryptographic keys that they can use for secure communications, and that are also employed to provide remote attestation, i.e. the ability to verify that a TEE has not been tampered with and ensure that the intended

operations are being performed properly. One of the main limitations of TEEs is that they typically only have access to limited hardware resources and cannot, for example, employ external AI accelerators, as that would require the unencrypted data to leave the CPU and move through a (potentially unsafe) system bus to reach the (potentially compromised) accelerator. However, due to the growing importance of data security not only in edge use cases, but also in cloud-based scenarios, it appears that major hardware producers are looking into lifting some of these limitations [56], [57].

Secure Multi-Party Computation The term Secure Multi-Party Computation [58] refers to a large family of cryptographic techniques that enable multiple participants to collaboratively perform an aggregate computation, without any of them ever revealing their specific contribution. The term SMC does not have the same clear-cut definition as the previous ones, because SMC solutions can be built using a variety of underlying protocols, including Homomorphic Encryption [59] However, Homomorphic Encryption and TEEs are general approaches to perform blind computations in any context and are not limited to, nor specifically designed for, distributed data sources. On the other hand, SMC specifically targets the latter scenario, and focuses on multiple parties communicating with each other to perform a joint computation.

In this thesis: Blind computation techniques are extensively discussed as building blocks for decentralized, privacy-preserving data marketplaces in **Paper D** [30] and **Paper E** [31].

2.3.2 Differential Privacy

Unfortunately, even if the updates contributed by individual participants to the massively-distributed training of an ML are protected through any of the blinded computation techniques described above, sensitive information can still be leaked through other means. Studies have shown that it is, in fact, possible to extract sensitive information by analyzing or even just querying the final trained model [60].

That is because blind computation techniques, in general, only hide the intermediate information necessary to produce an output, but do not affect the output itself, which is indistinguishable from – and can leak as much information as – what would be obtained without any blinding technique.

One well-known approach to prevent this kind of leakage is Differential Privacy [26]. At its core, Differential Privacy considers the output of a query Q that, when run on a dataset D, produces a numerical result. Random noise is then added to the result before disclosing it to the entity that submitted the query. The distribution from which the noise is sampled is specifically crafted so

that the probability of $Q(D)$ returning v differs from the probability of $Q(D')$ returning v by at most a factor e^{ε}, where ε is a tunable *privacy budget* and where D' is any dataset that differs from D in only a single data point.

Intuitively, this mathematical definition guarantees that a curious or malicious entity querying a dataset may not use the output of the query to confidently determine whether a specific data point exists in the dataset or not, as the datasets with and without that specific data point would have very similar chance of returning the observed output. The smaller the privacy budget, the higher the similarity in the output distributions of similar datasets, and thus the lower the confidence of the malicious observer in the exact composition of the queried dataset.

It is important to note that, if the same dataset is queried multiple times, it is possible to neutralize the added noises and pinpoint the true answer to the queries, which could then be used to identify the exact composition of the dataset and thus potentially leak sensitive information. As such, if n (not necessarily identical) queries need to be performed on a single dataset, while maintaining an overall privacy budget ε, then the actual privacy budget to set for each query is ε/n.

Differential Privacy has been extensively employed in privacy-preserving ML, not only in the context of massively-distributed training, but also in traditional centralized scenarios, in order to enable the trained model to be disclosed without endangering sensitive training data.

In this thesis: **Paper B** [28] and **Paper C** [29] introduce differentially-private Naive Bayes classifiers trained with federated and gossip-based aggregation schemes respectively. Differential Privacy is also discussed in the context of decentralized, privacy-preserving data marketplaces in **Paper D** [30] and **Paper E** [31].

Chapter 3

Summary of Appended Papers

This thesis is supported by 8 publications, as briefly described in section 1.3. In this chapter, we expand on those previous descriptions, providing additional context and an extended summary of the key contributions of these papers. In doing so, we group our papers according to their main topics, loosely matching with the research objectives defined in section 1.2.

3.1 Paper A: Pushing the Limits of Gossip Learning

In **Paper A** [27], we focus on objective **O1**: identifying and addressing the unrealistic assumptions that are often present in pre-existing research on Gossip Learning.

We argue that the assumptions that we identify limit the applicability of Gossip Learning to real-world environments because they do not sufficiently account for the *heterogeneity* that is often displayed in decentralized edge environments. We identify three axes of heterogeneity on which these restrictive assumptions are made.

- On the **data distribution** axis, many Gossip Learning studies, possibly owing to the traditional setting of gossip-based aggregation protocols, assume a fully-distributed setting, where only a single data point is stored in each device. But in ML applications, it is often the case that each device contains multiple data points. For example, a smartphone may store multiple photos that can be used for image classification purposes, or many messages to train predictive text models. Furthermore, these data are often non-IID distributed in two ways: first, different devices may have significantly different amounts of data; second, these local datasets may be drawn from different underlying distributions.

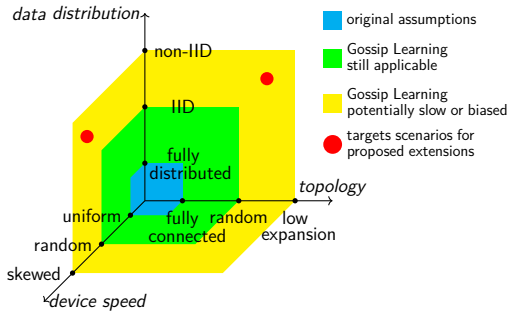


Figure 3.1: The three axes of unrealistic assumptions analyzed in **Paper A** [27]. Image adapted from **Paper A**.

- On the **device speed** axis, it is often assumed that all devices are capable of gossiping at the same frequency. However, some devices may have limited computational power and struggle to perform the required merging and local training steps in the given interval, and/or may have low network bandwidth and take too long to transfer the updated models. It is possible to overcome this issue by setting a very long interval between gossiping steps, but that would slow down the training process in terms of wall-clock time. As such, it may be beneficial to lift the assumption of identical device speeds and evaluate the behaviour of Gossip Learning when some devices gossip models significantly less frequently than others.

- Finally, on the **topology** axis, the traditional setting for Gossip Learning – and, more in general, gossip-based protocols – assumes that every device is able to gossip its update to any other device in the network. However, IoT devices deployed in different environments around the world may not be able to do that, due to security restrictions or more general network policies. The devices might instead be forced to gossip over a fixed topology, which may present characteristics that are suboptimal for the fast and unbiased spreading of information, such as the presence of tight communities or a power-law degree distribution.

On each of these axes, we *gradually* lift the typical assumptions, and quantitatively measure the behaviour of Gossip Learning, in order to identify at what point it no longer satisfies the requirements of fast and unbiased convergence to the global model, and therefore map the space of scenarios where Gossip Learning can be safely employed, as shown in fig. 3.1.

For our evaluation, we train a logistic regression model and employ both typical, well-known ML datasets and a simple, synthetic one. We build the latter in such a manner that the weights of the regressor can be easily plotted and

interpreted to identify the presence of model bias on each of the participating devices.

Our results show that Gossip Learning is generally robust to a variety of real-world conditions and can therefore often be employed safely. However, we found that issues arise when **non-IID data distributions** are correlated with other heterogeneous characteristics.

- One such issue arises when the data distribution is correlated with the **community structure** of a restricted network topology, that is, different communities in the topology have different data distributions. As each device is exposed almost exclusively to information coming from within its own community, it takes a large number of iterations for the gossiping and merging process to propagate the information across the communities, and therefore Gossip Learning requires a large number of iterations to converge in this setting.

- A more serious issue arises when the data distribution is correlated with the **device speeds**, that is, when high-speed devices present different data distribution compared to low-speed devices. As the former gossip their updates more often, their influence on the overall training process is higher than it should be in a fair environment, and therefore the overall model is biased towards their data distribution and may fail to learn the patterns present in the data on low-speed devices.

  We propose and evaluate an effective technique to counter this bias: model caching. By modifying the merging process to employ two random cached models, rather than the latest two received ones, we ensure that models gossiped by low-speed devices are as likely to be merged and propagated by their receivers as those gossiped by high-speed devices, despite the latter being larger in number. Through this technique, high-speed devices effectively boost the propagation of models generated by low-speed ones.

- Another issue of model bias arises when the data distribution is correlated with the **degree distribution** of the restricted network topology, that is, when high-degree nodes have a different data distribution than low-degree ones. As high-degree nodes receive a large portion of all gossiped models and can train them on local data before gossiping them further, they have an undue influence on the overall training process and result in model bias.

  We mitigate this bias by employing a "pass-through" gossiping technique: instead of always performing model merging and local training, nodes may randomly choose to simply forward the received models as they are, with the chance of this happening depending on the degrees of sender and

receiver. As such, high-degree nodes will sometimes forward the models of low-degree nodes without local training.

3.1.1 Summary of Paper A

Relevant Objectives: O1

Key Contributions:

- Identified key assumptions in pre-existing Gossip Learning studies.
- Quantitatively evaluated the behaviour of Gossip Learning when these assumptions are lifted, showing its general robustness and highlighting specific scenarios in which it stops meeting its speed and fairness requirements.
- Proposed and evaluated effective techniques to prevent model bias in the highlighted scenarios and therefore extend the scope of applicability of Gossip Learning.

3.2 Papers B & C: Collaborative Training of Naive Bayes Classifiers on Sensitive User Data

Within the scope of objective **O2**, the first use case we consider is the collaborative training of Naive Bayes classifiers on highly-sensitive user data. Before delving into the details of our contribution and findings later in this section, it is helpful to first better define the characteristics of this use case.

3.2.1 Scenario

We investigate a *federated data* scenario, as described in section 2.1. More specifically, we consider a consortium of medium to large organizations, each of which is in possession of highly-sensitive data collected from a number of users/customers of that organization. These organizations wish to collaborate to train an ML classifier, exploiting the large size of their combined datasets to maximize the predictive capacity of the model. Having the organizations pool their datasets and train a single, global model can often provide significant advantages compared to each organization training its own private model using only local data. This is especially true when the data present significant amounts of variability or specific patterns that may cause the local data at each institution to not be representative of the overall population, thus potentially leading to biased models, or when the amount of data at each institution is small compared

to the complexity of the model or of the behaviours to be modelled, potentially leading to overfitting.

A key motivating example for this scenario is that of hospitals, or more in general health institutions, which hold a number of patient records, and may want to use them to train models capable of improving the diagnosing process for a certain set of diseases or conditions. In case of uncommon diseases, an individual hospital may not have sufficient records to build a reliable ML model capable of generalizing to new patients.

As mentioned, we consider the data stored by each organization to be highly sensitive, thus requiring us to take protective steps. More specifically, what we consider to be sensitive is whether each specific individual is included in the dataset of an organization. Going back to the health institutions example, we want to ensure that no one is able to tell, with good confidence, whether any of the institutions included the patient records of Bob in the dataset they used for training the collaborative model. This protection applies not only to the training process but also to the resulting model. That is, we want to reasonably hide the presence of Bob's records not only to other institutions and other facilitating entities (e.g. central aggregators) participating in the training process, but also to any third parties that might, at a later point, receive a copy of the trained model. Thus, not only the information exchanged during training, but also the final model parameters, must not betray Bob's participation.

This requirement immediately excludes the possibility of using SMC, TEEs or Homomorphic Encryption for this purpose, as these techniques shield access to the information exchanged during the training process, but typically guarantee that the final output of the computational process is indistinguishable from what has been obtained without any privacy-preserving technique. Thus, if an attacker can confidently discern the presence of Bob's patient record when all the training data is pooled in a central location and trained without any protective steps, by analyzing the final model weights, then the same attacker can also confidently discern, with the same approach, the presence of Bob's record when the model training is performed using SMC, TEEs or Homomorphic Encryption.

Alas, the correct technique to employ in this context is Differential Privacy. When applied locally at each institution, it guarantees that adding or removing Bob's records from the dataset of any institution does not significantly change the contribution of that institution to the model. As such, even with unrestricted access to these individual contributions from each institution, an attacker cannot confidently say whether Bob's records are part of the training set or not.

On the other hand, this application of Differential Privacy only protects the privacy of individual users. It does not provide any privacy guarantees to whole organizations. That is, an attacker with access to the final trained model may still be able to confidently discern whether a specific institution participated or

not in the collaborative training. Furthermore, an attacker with access to the individual contributions of a single organization may extract information about the overall distribution of the private dataset of that organization, although they cannot discern individual users. To shield entire organizations from membership inference and data leakage attacks, additional layers of differential privacy and/or privacy-preserving computation techniques must be employed.

However, we do not consider this necessary in our scenario, as we make the assumption that organization-level information is not sensitive. In our health-related scenario, for example, this would translate to considering the incidence of specific conditions, or the distribution of certain patient features, within an institution, not being sensitive.

Finally, it is also necessary to introduce the ML model that we focus on in this scenario. Federated Learning and Differential Privacy have been combined in several studies focused on Deep Learning models, but little attention has been given to more shallow ML models, despite their significant use in industrial environments. In particular, Naive Bayes classifiers are known to often provide robust results, while at the same time being lightweight and relatively interpretable. These classifiers are trained not via an iterative optimization process, but via a single pass on the dataset, by computing simple statistics, such as class counts and feature distributions. This makes the process extremely fast, requiring a single communication pass in a federated setting, and also allows to approach the topic of Differential Privacy from the traditional perspective of protecting sensitive database queries, rather than the more recent perspective of protecting ML updates.

3.2.2 Federated Differentially-Private Naive Bayes

In **Paper B** [28], we introduce and evaluate a *federated* approach for training a differentially-private Naive Bayes classifier in the scenario just described.

As mentioned, the parameters of a Naive Bayes classifier are certain key statistical information collected directly from the dataset. More specifically, they are:

- for each class, the *count* of data points belonging to that class;
- for each categorical feature, the *count* of data points presenting each of the possible categories of that feature; the count must be provided separately for each class;
- for each numerical feature, the mean and standard deviation of the feature distribution, computed separately for data points belonging to each class.

In our federated approach, the central aggregator must compute these statistics based on information provided by each participant. For *count* queries, the

local counts provided by each participant can be trivially summed. Numerical features are instead more challenging to compute, especially because the formula for computing the standard deviation requires knowledge of the mean. As such, a naive solution would require two rounds of communication, with the aggregator computing the global mean and then broadcasting it to all participants, allowing each of them to compute its contribution to the standard deviation. Instead, we utilize an alternative formulation that reduces the problem to the computation of two independent *sum* queries. More specifically, each participant computes and sends to the aggregator the sum of each numerical feature and the sum of the squares of each numerical feature (both separately for each class, as mentioned above). The central aggregator then utilizes these two values, together with the class-wise counts, to compute the mean μ and the second moment ς. The standard deviation can then be computed as $\varsigma - \mu^2$. Thus, only a single communication round is needed to compute all Naive Bayes parameters.

Before sending their local query results, all devices apply Differential Privacy locally. To do so, they sample noise from a Laplacian Distribution proportional to the overall *privacy budget* and the *local sensitivity* of the query. The overall privacy budget is fixed at all nodes and depends on an application-specific basic privacy budget, which is chosen by the consortium, and on the number of numerical features and categorical values in the datasets. The local sensitivity, on the other hand, depends on how much the result of a query would change if the local dataset were to be modified by modifying exactly one row. As such, for *count* queries, the local sensitivity is always 1, as changing one row can change each count by at most one. On the other hand, for *sum* queries, the local sensitivity depends on the local dataset distribution: the larger the values stored in a dataset, the larger the difference in the overall sum when one row in the dataset is modified.

This observation leads to one of the most interesting findings in the experimental evaluation of our approach. When a dataset is dominated by categorical features, our federated differentially-private approach provides slightly worse results compared to a centralized one under the same Differential Privacy conditions, with the gap increasing with the number of devices participating. That intuitively makes sense, as combining multiple noise contributions can lead to degradation in the quality of the parameters compared to a single, centralized application of Differential Privacy. On the other hand, on datasets dominated by numerical features, our approach can sometimes outperform a centralized solution. This is because, while our approach requires the combination of multiple noise contributions, in the case of numerical features, many of these noise contributions have significantly lower scales compared to a single, centralized one, as they are computed on smaller datasets with less variance between the rows.

Overall, we show that a federated solution can compete with, and in some cases outperform, a centralized one for the training of Naive Bayes classifiers in a differentially-private setting, while providing additional protections by not requiring a trusted entity to collect and query all the data.

3.2.3 Decentralized Differentially-Private Naive Bayes

In **Paper C** [29], we extend the previous approach by replacing the central aggregator with gossip-based decentralized aggregation protocols.

Following the random-walk interpretation of Gossip Learning presented in section 2.1.2, each participant spawns a Naive Bayes model that performs a random walk over the set of all participants, each time collecting local query results and merging with previous models. We apply correction factors to prevent the model parameters from exploding or vanishing over many random walk steps and show that convergence can be achieved in a small number of steps, even with a large number of participants. We thus achieve objective **O2.1**: the completely decentralized collaborative training of Naive Bayes classifiers on highly-sensitive data.

In this work, we also contribute to objective **O1**, by lifting the unrealistic assumption that different participants must have similarly-sized datasets and that all participants must agree on a common privacy budget. When these conditions are not true, the noise contributions of different participants are no longer uniform: for participants with smaller datasets and/or smaller privacy budgets, the "noise-to-signal ratio" in query results is significantly higher. However, our results show that our federated and decentralized approaches are robust to these skewed noise contributions and can therefore be safely employed in a variety of heterogeneous settings.

3.2.4 Summary of Papers B & C

Relevant Objectives: **O1** and **O2.1**

Key Contributions:

- Designed and implemented federated and decentralized approaches to collaboratively train differentially-private Naive Bayes classifiers on highly-sensitive user data stored across multiple organizations.
- Evaluated the proposed approaches against centralized counterparts, showing competitive, and sometimes superior, performance; analyzed the results and provided insights into the effect of local query sensitivity on these performance figures.

- Addressed the possibility of heterogeneous scenarios, where privacy budgets and dataset sizes are not uniform across the organizations, and showed that the proposed approaches can robustly handle these scenarios.

3.3 Papers D & E: Decentralized Privacy-Preserving Data Marketplaces

The second use case we develop within the scope of objective **O2** is that of data marketplaces, which sit at the core of the growing Data Economy. They represent a key step in the journey to achieve maximum exploitation of the available data sources to create advanced services to improve our society.

Data marketplaces present three key players: data providers, data consumers, and marketplace operators. Data consumers typically consist of organizations that believe they can use data to create business value. For them, the marketplace provides a way to quickly obtain large amounts of data that would otherwise be hard, if not impossible, to collect. An extensive data collection may in fact require a significant investment of time and resources, and may be hard to achieve if an organization does not already have a strong presence in the domain where that data belongs. This problem is further exacerbated by the fact that advanced ML services often benefit from collecting and matching data from a variety of domains. This lowering of the barriers necessary to obtain data and build services is particularly useful to new players, who do not have the wide presence and collection capabilities of more established organizations in their domain.

On the other hand, data providers may belong to two broad categories. For the majority of current well-established industrial marketplaces, these are other organizations, that collected data in their respective domains and most likely use them internally to provide their own services. For them, data marketplaces offer a low-risk opportunity to extract additional value from their data and capitalize on the innovative services that third parties can provide using their data. However, as most data ultimately derives from the monitoring of the activity of individual users, the idea of directly incorporating them in the Data Economy, bypassing traditional data-collecting organizations, has gained some popularity, especially in the research community. This would have the societal benefit of directly rewarding individuals for the business value they create through their daily activities. However, it also adds significant complexity to the design of a marketplace, due to the sheer number and heterogeneity of the involved players.

Finally, it is important to highlight that data marketplaces, and the Data Economy more in general, are still in their infancy, and that many technical and societal questions remain open. These include questions about governance, trust and data management structures, business value quantification and reward

distribution. However, the most important open question is arguably that of data privacy and security.

3.3.1 PDS^2: Privacy-Preserving Decentralized Data Sharing System

In **Paper D** [30], we attempt to tackle many of the challenges and questions just described, by designing a decentralized, privacy-preserving data marketplace. Our design focuses on the scenario where the data providers are the individual users with their smart, connected devices. As such, our system must be able to scale to a large number of players and must provide sufficient flexibility and modularity to accommodate the need of a heterogeneous crowd.

We therefore start by analyzing what are the key, and often conflicting, requirements of the various stakeholders in a user-centered data marketplace. From the user perspective, these include:

- the ability to maintain **full control** of which consumers use the data and for what purposes the data is used;
- **strong privacy guarantees**, especially when the data includes sensitive personal information which should not be leaked;
- **fair rewarding** schemes.

A traditional data marketplace, where data consumers pay to download a copy of the data, cannot fulfil these requirements. In fact, once an entity has obtained a copy of some data, it is impossible to prevent them from utilizing those data for any purpose they wish or sharing those data with any other entity, nor it is possible to effectively monitor such activities. Thus, a key aspect of our marketplace architecture is the idea that data consumers should not have access to the data, but should simply be allowed to submit tasks to the marketplaces, which will be executed on the available data in a blinded manner, with only the output released to the consumer. Thus, the users are aware, and can explicitly authorize or not, each individual use of their data, which will never leave the marketplace, and are rewarded for every such use.

At the same time, the requirements of the data consumers must be considered, including the need to protect any intellectual property (IP) that might be embedded in those tasks, such as the specific characteristics of an advanced ML model.

Given all these requirements, we build a data marketplace architecture comprising various modules, each responsible for a specific aspect. For each component, we investigate multiple existing technologies that can be employed in its implementation. At the core of our architecture are the *executors*, which

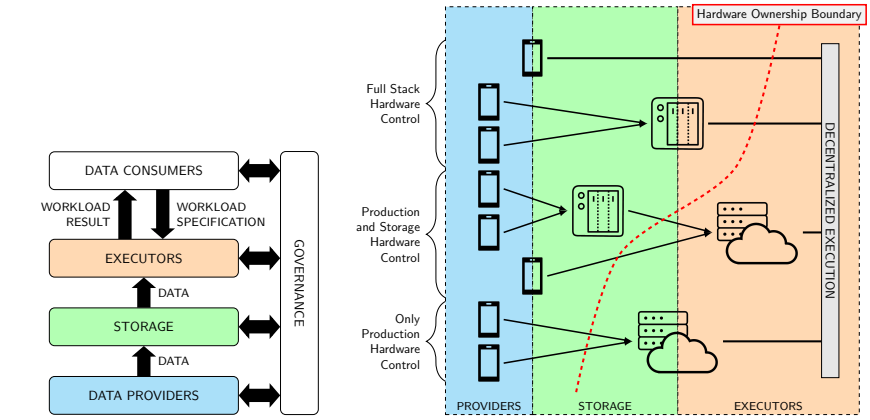


Figure 3.2: Diagrams showcasing the modular architecture (a) and flexible user-centric control structure (b) of PDS^2. Images adapted from **Paper D**.

are responsible for receiving data from the providers and tasks from the consumers, performing the tasks on the data, and returning a result. Gossip-based protocols, and in particular Gossip Learning, are employed by multiple executors holding separate data partitions to compute a global task output, ensuring robustness and scalability to large numbers of participants. Different blinding techniques (see section 2.3.1) are investigated to ensure that the executors can perform their operations on encrypted data and tasks, thus protecting the privacy of the users and the IP rights of the consumers. In the end, Trusted Execution Environments (TEEs) are chosen as the most suitable solution. Another key module of the architecture is the *decentralized governance* layer, which is responsible for tracking available data and pending tasks, for distributing rewards, and for auditing the behaviour of the marketplace. Here we employ the Ethereum blockchain, which provides powerful smart contracts to automate complex tasks, such as reward distribution.

The strong privacy and security guarantees provided by our architecture also contribute to its flexibility. End users are free to choose whether to store their data locally on their devices, or whether to rely on third-party storage solutions. Similarly, users can choose to have their own devices act as executors, thus ensuring that their data never leaves their physical control, or can employ third-party executors from the marketplace, knowing that their data is safely processed within TEEs. Figure 3.2 provides an overview of the marketplace architecture and of the various options for participating end users.

Finally, **Paper D** [30] also explores several key unanswered challenges in the

area of data marketplaces, reviewing state of the art solutions and providing directions for future research.

3.3.2 Narrowing the Gap between Academia and Industry

In **Paper E** [31], we review the field of data marketplaces from a different perspective. We observe that there is a growing gap between the advanced solutions developed by academic researchers and the real-world implementations deployed by most industrial marketplaces. While the former often employ complex architectures and cutting-edge technologies, with the goal of solving certain key issues that are identified in this domain, the latter seem to rely on more traditional architectures. We therefore set out to list these key issues that academic studies seek to solve, review the proposed state-of-the-art solutions for each of them, and identify those that could be *feasibly* implemented by the industry in the *short term*, with the goal of narrowing the gap between academia and industry.

First, we observe the current level of fragmentation in the data marketplace landscape, which causes issues of data penetration and data discovery. At the same time, we argue that a consolidation process would lead to new issues of accountability and trust, and therefore argue in favor of a *decentralized consortium* of players, based on blockchain technologies. After reviewing the available options, we consider *permissioned* blockchains to be the most feasible for this task in the short term. We also emphasize how a decentralized, open and standardized metadata ledger could enable efficient data discovery powered by third-party players.

Second, we consider the challenges brought by data privacy regulations and review the possibility of employing a range of techniques to ensure compliance and reduce leakage risks. A review of existing solutions in this area again points to TEEs being the most realistic approach, as their support is growing among major hardware and cloud service providers, while Homomorphic Encryption and SMC still present significant drawbacks. As discussed in section 2.3, we argue for combining blinded computation techniques with Differential Privacy, to prevent indirect information leakage from the computed results. We also suggest employing federated or decentralized approaches to ensure scalability, an important aspect considering that individual TEEs often present limited computational capabilities.

Finally, we consider the issue of data valuation, from two perspectives. First, a consumer must be able to decide which datasets are valuable for training a certain ML model, and therefore which datasets should be purchased on the marketplace. Then, after an ML model has been trained and its performance measured, the consumer must be able to quantify how each of the datasets employed affected that performance metric, in order to fairly reward the provider

of each dataset.

Overall, we see significant potential for different key technologies to be transferred from the academic to the industrial world, and we hope to soon see a narrowing of the current gap between the two.

3.3.3 Summary of Papers D & E

Relevant Objectives: O2.2

Key Contributions:

- Analyzed key players and requirements in the design of a user-centered data marketplace.
- Designed a decentralized, privacy-preserving data marketplace that is modular, scalable, highly flexible and user-centered.
- Identified a growing gap between academic and industrial developments in the area of data marketplaces.
- Thoroughly reviewed key enabling technologies for developing decentralized, privacy-preserving data marketplaces and identified those technologies that can be feasibly implemented in industrial designs in the short term.
- Reviewed and discussed several open questions in the area of data marketplace design.

3.4 Papers F & G: Early-Stage IoT Botnet Detection

The third and final use case that we consider within the scope of objective **O2** is that of early-stage botnet detection.

In recent years, our society has seen a quick growth in the number of cyberattacks and malware deployments. IoT devices are a particularly sensitive area in this regard, as their large number and significant variety in terms of hardware and software capabilities creates a large attack surface for malicious entities. The situation is further exacerbated by the limited security features of these devices and by a general lack of awareness and understanding of their risks.

Botnets are a specific type of malware that has been particularly successful in exploiting IoT devices. After infecting a device, botnets use its network interfaces to spread to other devices, exploiting known vulnerabilities to gain

access and infect them. Once a sufficient number of devices has been infected, the entity controlling the botnet can order all of them to perform a Distributed Denial of Service (DDoS) attack against a victim system, using the combined networking capabilities of the infected devices to overwhelm the target and prevent its normal operation.

Significant research has been dedicated to the detection of botnets. As botnets make heavy use of the network both during the spreading and the attack phase, network traffic analysis has emerged as one of the most promising directions in the implementation of botnet detection systems. However, most work in this area has focused on the attack phase, detecting the botnet when it initiates a DDoS attack and attempting to mitigate it. We believe that it would be beneficial to instead focus on the spreading phase, detecting and eradicating the botnet before it has the chance to perform any attacks. A few studies have explored this early-stage detection scenario, employing either Recurrent Neural Networks (RNNs) or graph-theoretical approaches [61].

3.4.1 LiMNet: a Centralized Solution

In **Paper F** [32], we develop and evaluate LiMNet, a novel deep learning approach for early-stage IoT botnet detection, building a recurrent GRL model that significantly outperforms existing solutions.

The scenario we focus on is that of an Industrial IoT (IIoT) deployment, such as an individual smart city or smart factory, with a significant number of devices, all connected to the same local network. In this context, we model the problem of detecting infected devices as a GRL *node classification* problem. We build a dynamic graph, where each node represents an IoT device and each edge is an individual packet exchanged. We use the streaming scenario described in section 2.2.2: smart access points and switches within the network forward a copy of the headers of each packet to a central monitoring server in real time. Running on this server, LiMNet processes each packet in sequence, building and updating node embeddings for each device.

Every time a new packet header is received, LiMNet updates the embeddings of the source and destination devices using a pair of *mutually-recurrent RNN cells*. In general, an RNN cell takes an existing embedding and updates it by combining it with additional input information. In LiMNet, one RNN cell updates the embedding of the source device using the previous embedding of the destination device, while simultaneously another RNN cell performs the equivalent update on the destination device embedding. The embeddings of all devices are stored in the system memory and can be queried at any point in time to extract useful information. In our evaluation, we utilize three ML classifiers: two device classifiers, fed directly with the individual embeddings, detect whether each device is infected or is currently under attack from in-

fected devices. A packet classifier is instead fed the concatenation of packet header features and source and destination device embeddings, with the goal of detecting whether an individual packet is part of a malicious communication flow.

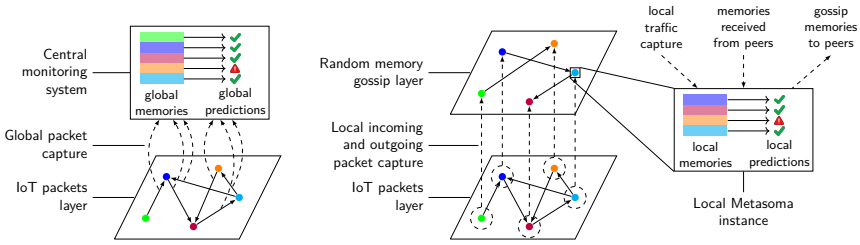
Our results show that LiMNet performs significantly better compared to previous recurrent deep learning models for early-stage botnet detection. Additionally, the LiMNet model is orders of magnitude smaller in terms of model weights, to the point of easily fitting in the L2 cache of a modern CPU core. And it is also much faster than previous approaches during inference, processing over 3000 packet headers per second when running on a single CPU core. Due to these characteristics, LiMNet can therefore be easily deployed directly on smart network infrastructure, such as programmable routers, without the need for a dedicated server or specialized AI accelerators.

3.4.2 Metasoma: a Decentralized Solution

However, despite its lightweight nature, LiMNet might not be suitable for very large deployments – where capturing and streaming all network traffic to a central system may be infeasible – or for deployments where no central entity controls the entirety of the network infrastructure or where peer-to-peer routing is employed. Thus, in **Paper G** [33], we build Metasoma: a decentralized botnet detection system, based on the LiMNet architecture, which runs directly on the IoT devices themselves, without any central system.

Figure 3.3 summarizes the main components of Metasoma and compares them to LiMNet. Instead of employing a central system for building embeddings and detecting botnets, in Metasoma each individual IoT device runs its own local copy of a modified LiMNet architecture, building local embeddings and performing local detection. Each device builds its own local embeddings by monitoring all traffic originating from, or directed to, itself. As such, these embeddings only provide a partial view of the behaviour of other devices. Therefore, Metasoma employs a gossiping scheme that allows the IoT devices to share and merge the embeddings they produce, leading to each device having a global picture of the behaviour of the network. To achieve this, Metasoma extends the LiMNet architecture with an additional trainable ML component, which learns the best strategy to merge the local embeddings with the received ones, in order to preserve all the complementary information that is captured by the two input embeddings.

However, running the botnet detection on the IoT devices themselves, rather than on separate, *trusted* hardware, significantly increases the attack surface of the IoT deployment. An attacker aware of the inner workings of Metasoma may exploit weaknesses in its design to perform malicious activities undetected. To prevent this, we perform an extensive security analysis of Metasoma, which



(a) LiMNet (b) Metasoma

Figure 3.3: Comparison of the centralized LiMNet architecture introduced in **Paper F** and the decentralized Metasoma architecture introduced in **Paper G**. Figure adapted from **Paper G**.

allows us to identify potential attack vectors and propose countermeasures to ensure our approach can be safely deployed.

One key attack vector is **embedding forgery**: an infected device may modify the behaviour of its local Metasoma instance to forge embeddings that do not reflect the true behaviour of other devices, and may gossip these forged embeddings to honest devices in order to perform a *label flipping* attack and prevent the detection of the botnet. However, due to the way Metasoma is trained, enacting this attack is challenging. As already mentioned, when an honest device merges its local embedding with a received one, it preserves the complementary information of both. Thus, if the local embedding of a device encodes a malicious behaviour, this will not be erased by receiving an embedding that encodes honest behaviour. So the only way an infected device can successfully perform a label flipping attack is by forging an embedding that significantly deviates from what Metasoma was trained on, thus confusing the ML component responsible for the merging and resulting in the erasure of important information. This analysis leads us to enhance Metasoma with a *forgery detector* component: a separate deep learning model trained to detect and reject anomalous embeddings that significantly deviate from those seen during the training process.

Another attack vector consists of performing the opposite kind of label flipping: instead of forcing the classification of a malicious device as benign (to prevent detection), an attacker may instead choose to force the classification of a benign device as malicious, with the goal of having that device excluded from the network, and thus potentially achieve a Denial of Service (DoS) attack, if the victim device performs an important function in the network. Unfortunately,

this does not require the forging of anomalous embeddings, as it can be simply achieved by injecting forged packets, seemingly from the victim device, into the local network stack. These are then picked up by Metasoma and encoded in the embedding of the victim, which is then gossiped as usual, quickly resulting in all devices believing the victim to have sent those malicious packets. While this attack cannot be fully prevented, its potential impact on an IoT deployment can be significantly reduced by restricting device-to-device communications, so that, at any point in time, each device only has a certain number of neighbours in the network. Thus, an infected device can only flag as malicious those devices that it is connected to, and therefore the "blast radius" of a security breach is limited to a small neighbourhood of the overall network.

Our results show that Metasoma provides competitive performance when compared to centralized solutions such as LiMNet. Furthermore, we also evaluate our security countermeasures and find them to be effective in protecting our detection system.

One final note is that, due to its decentralized nature and employment of GRL techniques, Metasoma contributes not only to objective **O2.3**, but also to objective **O3**. However, Metasoma only performs decentralized *inference*. Its *training* process is still performed offline, on a centralized system, and due to its computational requirements, necessitates the use of AI accelerators. This will motivate our next and final work on decentralized training.

3.4.3 Summary of Papers F & G

Relevant Objectives: **O2.3** and **O3**

Key Contributions:

- Developed and evaluated a lightweight GRL-based model for early-stage detection of IoT botnets that is more accurate, smaller and faster than pre-existing approaches.

- Developed and evaluated a decentralized variant of the previous approach, which achieves competitive results without the need for any centralized infrastructure.

- Performed an extensive security analysis of our decentralized botnet detection system, identifying potential vulnerability, proposing relevant countermeasures and evaluating the effectiveness of those countermeasures.

3.5 Paper H: Fully-Decentralized Training of GNNs

Our last contribution, **Paper H** [34], completes the work on objective **O3**, by presenting a solution for the fully-decentralized training of convolutional GNNs with limited local knowledge.

When discussing decentralized GRL, it is important to make a clear distinction between two different graphs. One is the *communication network* that connects the devices participating in the training process, where the nodes are hardware devices and the edges are network connections. The other is the *target graph*, the one on which the ML model is being trained, in which the meaning of nodes and edges is specific to the given application domain.

In this work, we consider a *fully-decentralized* scenario, where each device in the communication network matches one-to-one with a node in the target graph. This is a very common scenario: personal connected devices such as smartphones and smartwatches typically match one-to-one with individuals, while traffic monitoring devices in a smart city deployment may match with roads or road intersections. We also make an additional assumption of the communication network being fully connected, knowing that our approach will be robust to more complex scenarios thanks to the results from **Paper A** [27].

We also require that each device must possess minimal knowledge of the target graph, in order to preserve privacy. More specifically, each device may only be aware of the feature of its own node and of the nodes directly connected to it in the target graph, and may only be aware of those edges in the target graph that are adjacent to its own node. In other words, each device only knows about the one-hop neighbourhood of its corresponding node in the target graph.

This requirement, however, introduces significant challenges: as mentioned in section 2.2.1, deep convolutional GNNs build wide *receptive fields*, and thus to produce the output embedding for an individual node they must take as input its entire L-hop neighbourhood, where L is the number of GNN layers. This goes against our requirement for local knowledge only. A naive workaround would consist in having each node broadcast its embedding to all neighbours, which would then use it to build higher-level embeddings, directly implementing over the network the logical message passing architecture described in section 2.2.1. However, this would lead to a large number of embedding transfers being performed in each forward pass, each of which would need to be followed up by a matching gradient transfer in the backward pass, leading to prohibitive bandwidth requirements. It would also lead to a significant amount of dependencies between the computations carried about by the different devices, and therefore introduce synchronization issues and potentially long idle times for a majority of the nodes, while waiting for a few stragglers.

Our approach instead removes all synchronization requirements across devices and significantly limits the bandwidth required by the system. To achieve

this, we combine several key enabling technologies in a novel way.

- We employ **decoupled, layer-wise learning**, inserting gradient stops between each layer in the GNN. In this way, each layer treats its inputs as if they were fixed features, rather than trainable embeddings from the previous layer. Thus, costly backward-pass communications to exchange gradients are no longer needed. Conversely, these gradient stops prevent any training signal from flowing through the network, and therefore prevent all layers from learning.

- We solve this issue by equipping each layer with a **self-supervised loss** function. This provides a local source of gradients, and thus training signal, independent of any other layers, thus enabling each layer to learn in a standalone way. While a task-specific loss function (such as a classification loss based on ground-truth labels) would lead the model to capture specific phenomena in the graph that explain the given labels, a self-supervised loss forces the model to capture all phenomena that are relevant to reconstructing the original graph, and thus lead to the creation of more general and flexible node embeddings. However, no self-supervised loss can function with a single, local embedding as input. Instead, they all require the local embedding to be compared with other, randomly-chosen embeddings from the graph.

- Thus, we utilize a **decentralized, push-based negative sampling** approach, whereby each device, at every iteration, will select several random peers from the whole communication network, and send to them its local node embeddings. The receiving devices will use these as negative samples for their local self-supervised losses.

- We make extensive use of **embedding buffering**: each device maintains a fixed-size FIFO buffer containing the latest K embeddings received through negative sampling, and uses the entire buffer in each of its self-supervised training iterations. Thus, it does not need to receive K fresh negatives at each iteration, instead reusing the same multiple times until they are eventually replaced. As such, each device can enjoy a large number K of negative samples, while only needing to push its embeddings to $K' \ll K$ peers at each iteration, significantly reducing the network pressure. Similarly, the embeddings of neighbours are also cached and do not need to be broadcasted after every iteration.

- While each device performs its training steps independently, we employ **Gossip Learning** to share the partially-trained models across the network and achieve convergence of the model weights to the global optimum.

- Finally, we employ gossip-based **decentralized membership management** techniques to ensure that each device possesses contact information (e.g. IP addresses and port numbers) of a *uniform random sample* of the other devices in the network. This is necessary to allow each device to choose, at every iteration, random peers to whom to push its embeddings as negative samples, and to whom to gossip its model weights according to the Gossip Learning protocol.

We build two implementations of our system. One is a full, multi-core simulator that captures all of the aforementioned aspects of our solution, and can provide accurate predictions of its behaviour. Another is an approximate, GPU-accelerated emulator that only reproduces some aspects of the architecture, but that allows for the quick tuning and analysis of different design choices, enabling us to better understand the characteristics of our solution.

We employ these two implementations to evaluate our approach. As it is not possible to directly assess the quality of self-supervised embeddings, we apply them to a down-stream supervised classification task, and use the accuracy on that task as a proxy metric for the quality of the embeddings. The results show that the performance of our approach is relatively close to that obtained by a centralized architecture, despite the severe limitations that our scenario poses on the distribution of global graph knowledge. Furthermore, our emulator enables us to provide an extensive analysis of the behaviour of our approach. Overall, we demonstrate that our solution is a feasible alternative to existing centralized training architecture, especially for privacy-conscious scenarios where centralized knowledge collection is not feasible.

Due to the many different building blocks that form our solution, our promising results open several research questions, relating to the optimization of the different components employed and the exploitation of our architecture in more complex scenarios. We extensively discuss these open questions and suggest a path for future studies to tackle them, closing the gap between decentralized and centralized training and bringing the former into industrial applications.

3.5.1 Summary of Paper H

Relevant Objectives: O3

Key Contributions:

- Designed a novel approach for the fully-decentralized training of convolutional GNNs with limited local knowledge, based on the combination of several key enabling technologies.
- Implemented and evaluated the proposed approach, analyzing its behaviour and showcasing promising results.

- Discussed key new research questions arising from this study, highlighting a new, promising research direction for future works.

Chapter 4

Conclusions and Future Work

Advanced, data-intensive ML techniques play a key role in the development of novel digital services, and the improvement of existing ones, that can positively affect our society. Thus, it is fundamental to maximize our ability to exploit the vast troves of data that our society produces every day. This requires us to overcome the issues that are becoming more and more apparent in our traditional, centralized data processing and ML architectures, including scalability, reliability, security, privacy and trust.

In this thesis, we argued that these issues can be sustainably solved by performing a paradigm shift, moving towards a completely decentralized and trustless architecture for privacy-aware ML applications, and in particular for Graph Representation Learning (GRL). We then identified the key roadblocks that prevent this decentralized vision from being immediately applied, and we set out to remove them.

This thesis first discussed the unrealistic assumption of previous works in the area of Gossip Learning, regarding heterogeneous device capabilities, non-IID data and varying privacy budgets. These assumptions were lifted and Gossip Learning was shown to be robust to a wide variety of realistic conditions and therefore safely applicable in many real-world scenarios.

Then, the limited penetration of gossip-based approaches in the domain of applied ML was addressed. This thesis showed the suitability of gossip-based aggregation for the collaborative training of Naive Bayes classifiers on highly-sensitive user data owned by a consortium of organizations. It discussed the use of Gossip Learning as a building block for a decentralized, privacy-preserving data marketplace that is highly flexible, modular and user-centric. Gossip communications were also employed to distribute GRL node embeddings across IoT devices at inference time, enabling the decentralization of a powerful system for the early-stage detection of IoT botnets.

Finally, this thesis tackled the unique challenges that prevent the naive decentralization of GNN training processes. Through a novel combination of sev-

eral techniques, including Gossip Learning and push-based gossiping of negative samples, a framework was designed that efficiently trains self-supervised GNNs in a fully-decentralized setting, achieving very promising results and opening a broad new research direction concerned with tuning and exploiting this novel approach.

Overall, through this work, we removed major obstacles to the implementation of our vision, enabling the development of completely decentralized architectures for graph learning.

Future Work Our work, especially in the area of decentralized GNN training, gives rise to several interesting questions for future studies to explore. The combination of many different components in our solution calls for in-depth studies into how each of them impacts the overall performance of the system, and for the development of tailored solutions that can replace some of the off-the-shelf implementations employed and narrow the gap to centralized alternatives. Privacy-preserving techniques should also be explored, to provide stronger privacy guarantees. Finally, more complex models and scenarios should be tested, to catch up with the latest advancements in centralized GRL applications.

The broader domain of Gossip Learning also presents open research directions for future studies. Many of the scenarios explored in the context of Federated Learning should be investigated from a purely decentralized perspective. Of particular note is the lack of in-depth studies on byzantine behaviours in Gossip Learning applications. Byzantine-resilient gossip protocols need to be developed and thoroughly examined to guarantee the robustness of decentralized ML applications to malicious attackers and achieve the goal of a completely trustless architecture.

Bibliography

[1] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," *Journal of Big data*, vol. 2, no. 1, pp. 1–32, 2015.

[2] P. Russom *et al.*, "Big data analytics," *TDWI best practices report, fourth quarter*, vol. 19, no. 4, pp. 1–34, 2011.

[3] T. M. Mitchell *et al.*, *Machine learning*. McGraw-hill New York, 2007, vol. 1.

[4] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[5] S. Li, L. D. Xu, and S. Zhao, "The internet of things: A survey," *Information systems frontiers*, vol. 17, pp. 243–259, 2015.

[6] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: An overview," *The internet society (ISOC)*, vol. 80, pp. 1–50, 2015.

[7] B. Furht, F. Villanustre, B. Furht, and F. Villanustre, "Introduction to big data," *Big data technologies and applications*, pp. 3–11, 2016.

[8] J. Černiauskas. "Understanding the 4 v's of big data." (Aug. 23, 2022), [Online]. Available: `https://www.forbes.com/sites/forbestechcouncil/2022/08/23/understanding-the-4-vs-of-big-data` (visited on 03/10/2023).

[9] F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "Graph representation learning: A survey," *APSIPA Transactions on Signal and Information Processing*, vol. 9, e15, 2020.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[11] P. Jauernig, A.-R. Sadeghi, and E. Stapf, "Trusted execution environments: Properties, applications, and challenges," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 56–60, 2020.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[13] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[14] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Transactions on Computer Systems (TOCS)*, vol. 23, no. 3, pp. 219–252, 2005.

[15] R. Ormándi, I. Hegedűs, and M. Jelasity, "Gossip learning with linear models on fully distributed data," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 4, pp. 556–571, 2013.

[16] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *arXiv preprint arXiv:2109.04269*, 2021.

[17] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[18] J. So, B. Güler, and A. S. Avestimehr, "Byzantine-resilient secure federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2168–2181, 2020.

[19] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Distributed Applications and Interoperable Systems: 19th IFIP WG 6.1 International Conference, DAIS 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17–21, 2019, Proceedings 19*, Springer, 2019, pp. 74–90.

[20] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Network*, vol. 35, no. 1, pp. 234–241, 2020.

[21] C. Pappas, D. Chatzopoulos, S. Lalis, and M. Vavalis, "Ipls: A framework for decentralized federated learning," in *2021 IFIP Networking Conference (IFIP Networking)*, IEEE, 2021, pp. 1–6.

[22] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "Fedgnn: Federated graph neural network for privacy-preserving recommendation," *arXiv preprint arXiv:2102.04925*, 2021.

[23] M. Jiang, T. Jung, R. Karl, and T. Zhao, "Federated dynamic gnn with secure aggregation," *arXiv preprint arXiv:2009.07351*, 2020.

[24] C. He, K. Balasubramanian, E. Ceyani, *et al.*, "Fedgraphnn: A federated learning system and benchmark for graph neural networks," *arXiv preprint arXiv:2104.07145*, 2021.

[25] D. Berrar, "Bayes' theorem and naive bayes classifier," *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 403, p. 412, 2018.

[26] C. Dwork, "Differential privacy," in *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, Springer, 2006, pp. 1–12.

[27] L. Giaretta and Š. Girdzijauskas, "Gossip learning: Off the beaten path," in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 1117–1124.

[28] T. Marchioro, L. Giaretta, E. Markatos, and Š. Girdzijauskas, "Federated Naive Bayes under Differential Privacy," in *19th International Conference on Security and Cryptography (SECRYPT), JUL 11-13, 2022, Lisbon, Portugal*, Scitepress, 2022, pp. 170–180.

[29] L. Giaretta, T. Marchioro, E. Markatos, and Š. Girdzijauskas, *Towards a Realistic Decentralized Naive Bayes with Differential Privacy*, under review.

[30] L. Giaretta, I. Savvidis, T. Marchioro, *et al.*, "PDS^2: A user-centered decentralized marketplace for privacy preserving data processing," in *2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW)*, IEEE, 2021, pp. 92–99.

[31] L. Giaretta, T. Marchioro, E. Markatos, and Š. Girdzijauskas, "Towards a decentralized infrastructure for data marketplaces: narrowing the gap between academia and industry," in *Proceedings of the 1st International Workshop on Data Economy*, 2022, pp. 49–56.

[32] L. Giaretta, A. Lekssays, B. Carminati, E. Ferrari, and Š. Girdzijauskas, "LiMNet: Early-Stage Detection of IoT Botnets with Lightweight Memory Networks," in *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I*, Springer, 2021, pp. 605–625.

[33] L. Giaretta, A. Lekssays, B. Carminati, E. Ferrari, and Š. Girdzijauskas, *Metasoma: Decentralized and Collaborative Early-Stage Detection of IoT Botnets*, under review.

[34] L. Giaretta and Š. Girdzijauskas, *Fully-Decentralized Training of GNNs using Layer-wise Self-Supervision*, under review.

[35] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[36] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[37] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multi-party computing," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6178–6186, 2020.

[38] A. Montresor, "Gossip and epidemic protocols," *Wiley encyclopedia of electrical and electronics engineering*, vol. 1, 2017.

[39] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[40] M. M. Li, K. Huang, and M. Zitnik, "Graph representation learning in biomedicine and healthcare," *Nature Biomedical Engineering*, pp. 1–17, 2022.

[41] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[42] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy, "Machine learning on graphs: A model and comprehensive taxonomy," *Journal of Machine Learning Research*, vol. 23, no. 89, pp. 1–64, 2022.

[43] A. Dalmia and M. Gupta, "Towards interpretation of node embeddings," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 945–952.

[44] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, PMLR, 2017, pp. 1263–1272.

[45] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[46] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *Proceedings of the 25th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2019.

[47] Y. Ma, Z. Guo, Z. Ren, J. Tang, and D. Yin, "Streaming graph neural networks," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 719–728.

[48] Z. T. Kefato, S. Girdzijauskas, N. Sheikh, and A. Montresor, "Dynamic embeddings for interaction prediction," in *Proceedings of The Web Conference 2021*, 2021.

[49] A. Pareja, G. Domeniconi, J. Chen, *et al.*, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 5363–5370.

[50] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dysat: Deep neural representation learning on dynamic graphs via self-attention networks," in *Proceedings of the 13th international conference on web search and data mining*, 2020, pp. 519–527.

[51] H. Yousuf, M. Lahzi, S. A. Salloum, and K. Shaalan, "Systematic review on fully homomorphic encryption scheme and its application," *Recent Advances in Intelligent Systems and Smart Applications*, pp. 537–551, 2021.

[52] V. Haralampieva, D. Rueckert, and J. Passerat-Palmbach, "A systematic comparison of encrypted machine learning solutions for image classification," in *Proceedings of the 2020 workshop on privacy-preserving machine learning in practice*, 2020, pp. 55–59.

[53] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE symposium on security and privacy (SP)*, IEEE, 2017, pp. 19–38.

[54] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via minionn transformations," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 619–631.

[55] O. Ohrimenko, F. Schuster, C. Fournet, *et al.*, "Oblivious {multi-party} machine learning on trusted processors," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 619–636.

[56] F. Mo, Z. Tarkhani, and H. Haddadi, "Sok: Machine learning with confidential computing," *arXiv preprint arXiv:2208.10134*, 2022.

[57] A. C. Elster and T. A. Haugdahl, "Nvidia hopper gpu and grace cpu highlights," *Computing in Science & Engineering*, vol. 24, no. 2, pp. 95–100, 2022.

[58] Y. Lindell, "Secure multiparty computation," *Communications of the ACM*, vol. 64, no. 1, pp. 86–96, 2020.

[59] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Annual Cryptology Conference*, Springer, 2012, pp. 643–662.

[60] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[61] D. Zhuang and J. M. Chang, "Peerhunter: Detecting peer-to-peer botnets through community behavior analysis," in *2017 IEEE Conference on Dependable and Secure Computing*, IEEE, 2017, pp. 493–500.

Appended Papers