UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Pedro Henrique Silva Souza Barros

Uncertainty quantification in Adversarial Federated Learning

Belo Horizonte
2026

Pedro Henrique Silva Souza Barros

Uncertainty quantification in Adversarial Federated Learning

Final Version

Dissertation propostal presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Heitor Soares Ramos Filho
Co-Advisor: Fabrício Murai Ferreira & Amir Houmansadr

Belo Horizonte
2026

Barros, Pedro Henrique Silva Souza.

B277u Uncertainty quantification in Adversarial Federated Learning [recurso eletrônico] / Pedro Henrique Silva Souza Barros – 2026.
1 recurso online (165 f. il., color.) : pdf.

Orientador: Heitor Soares Ramos Filho.
Coorientador: Fabrício Murai Ferreira.
Coorientador: Amir Houmansadr

Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciências da Computação.
Referências: f.134-151

1. Computação – Teses. 2. Aprendizado federado (Aprendizado do computador) – Teses. 3. Redes neurais (Computação) – Teses. I. Ramos Filho, Heitor Soares. II. Ferreira, Fabrício Murai. III. Houmansadr, Amir. IV. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. V. Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Uncertainty quantification in Adversarial Federated Learning

PEDRO HENRIQUE SILVA SOUZA BARROS

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Documento assinado digitalmente
gov.br HEITOR SOARES RAMOS FILHO
Data: 12/02/2026 12:13:22-0300
Verifique em https://validar.iti.gov.br

PROF. HEITOR SOARES RAMOS FILHO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. FABRÍCIO MURAI FERREIRA - Coorientador
Departamento de Ciência da Computação - UFMG

PROF. AMIR HOUMANSADR - Coorientador
Faculdade de Ciências da Informação e da Computação - Universidade de Massachusetts Amherst

Documento assinado digitalmente
gov.br ALEJANDRO CESAR FRERY ORGAMBIDE
Data: 28/01/2026 23:53:13-0300
Verifique em https://validar.iti.gov.br

PROF. ANTONIO ALFREDO FERREIRA LOUREIRO
Departamento de Ciência da Computação - UFMG

Documento assinado digitalmente
gov.br ALEJANDRO CESAR FRERY ORGAMBIDE
Data: 22/01/2026 17:23:05-0300
Verifique em https://validar.iti.gov.br

PROF. ALEJANDRO CÉSAR FRERY ORGAMBIDE
School of Mathematics and Statistics - Victoria University of Wellington

Documento assinado digitalmente
gov.br MARCOS OLIVEIRA PRATES
Data: 23/01/2026 11:34:42-0300
Verifique em https://validar.iti.gov.br

PROF. MARCOS OLIVEIRA PRATES
Departamento de Estatística - UFMG

Documento assinado digitalmente
gov.br AMAURI HOLANDA DE SOUZA JUNIOR
Data: 10/02/2026 00:31:35-0300
Verifique em https://validar.iti.gov.br

PROF AMAURI HOLANDA DE SOUZA JÚNIOR
Departamento de Ciência da Computação - IFCE

Belo Horizonte, 21 de janeiro de 2026.

Agradecimentos

Primeiramente, agradeço a Deus por todas as graças alcançadas em minha vida.

Gostaria de agradecer a minha família, especialmente meus pais, Rafael e Vera, bem como meu irmão João, por todo apoio (seja ele financeiro, emocional ou educacional) que me foi oferecido durante todo o meu período acadêmico. Tenho certeza que sem esse apoio, não teria conseguido finalizar essa dissertação. Vocês são minha inspiração e motivação da minha vida.

Agradeço também a Joyce Elisa Herédia. Muito obrigado por toda paciência, apoio, conversas, sushis e conselhos durante todo esse ano.

Além disso, agradeço ao meu mentor, amigo e orientador Dr. Heitor Ramos, por todo apoio que venho recebendo nesses últimos anos, desde minha graduação. Através de sua sugestão, comecei a analisar a hipótese de realizar mestrado na UFMG.

Meus grandes amigos do Conversa Mole, por todas brincadeiras e resenhas do dia-a-dia. Em especial, meus parceiros do return (Demetrios, Alvino e Matheus), que mesmo longe, ainda mantemos a amizade construída durante todo período de graduação. Além disso, agradeço meu parceiro Cristopher por todas conversas sobre o mestrado, bem como eventos e viagem acadêmicas.

Meus companheiros de UFMG, que encararam participar dessa jornada comigo durante esses quatros anos, além de me apresentar os quatros gatos mais legais de BH: Jujuba, Rocky Balboa, Miguel e Nina.

E por fim, agradeço a todos que contribuíram de alguma forma neste trabalho, bem como a banca avaliadora pela leitura e comentários acerca desta tese.

"*I was here. I lived, I loved. I was here. I did, I've done, everything that I wanted and it was more than I thought it would be.*"

(Beyoncé *in "I Was Here"*)

Resumo

Esta tese investiga novas metodologias em Aprendizado Federado (FL), um paradigma que permite múltiplos dispositivos desenvolverem colaborativamente um modelo de aprendizado de máquina compartilhado, mantendo todos os dados de treinamento locais, assim aprimorando a privacidade dos clientes. O FL opera treinando modelos locais em dispositivos individuais, que são então agregados em um servidor central. Apesar de suas vantagens, o FL é vulnerável a ataques de envenenamento de modelo, onde nós maliciosos injetam atualizações maliciosas de modelo, comprometendo a integridade do modelo global. Esta tese, intitulada "Quantificação de Incerteza em Aprendizado Federado Adversarial", introduz novas abordagens para melhorar a privacidade e a segurança dos modelos de aprendizado de máquina distribuídos contra tais ameaças. Para alcançar esse objetivo, a pesquisa explora três métodos distintos para quantificar a incerteza em modelos de FL. O primeiro método, aproximação de Laplace usando a matriz Hessiana em redes neurais, é aplicado especificamente na detecção de ataques de Negação de Serviço Distribuído (DDoS) em cenários de FL. Este método aproveita as derivadas de segunda ordem da função de perda para aproximar a incerteza nas previsões do modelo, proporcionando uma compreensão refinada da confiança do modelo na presença de ataques adversariais e aprimorando a detecção e mitigação de ataques DDoS. O segundo método introduz uma abordagem ad-hoc usando uma técnica de aprendizado de métrica profunda, denominada "SMELL". Este método define um espaço de similaridade (S-Space) para representar dados de forma mais eficaz, mapeando pares de elementos do espaço de *features* original para este novo espaço auxiliar. A similaridade entre pares de dados é quantificada usando marcadores dentro do S-Space, permitindo uma detecção intuitiva e flexível de anomalias e ameaças potenciais em ambientes de FL. O terceiro método estende a abordagem ad-hoc empregando redes neurais bayesianas com inferência variacional. Esta extensão utiliza princípios bayesianos para modelar a incerteza tratando os pesos da rede como distribuições em vez estimativas pontuais, permitindo uma interpretação probabilística das saídas do modelo e uma resiliência aprimorada contra ataques maliciosos. Integrando esses métodos de quantificação de incerteza, a tese visa mitigar os riscos de ataques de envenenamento de modelo, aprimorando assim a robustez, confiabilidade e segurança das aplicações de FL.

Palavras-chave: aprendizado federado; redes neurais bayesianas; aprendizado federado adversarial; quantificação de incerteza.

Abstract

This thesis investigates novel methodologies in Federated Learning (FL). This paradigm enables multiple devices to collaboratively develop a shared machine learning model while keeping all training data localized, thus enhancing client privacy. FL trains local models on individual devices, which are then aggregated on a central server. Despite its advantages, FL is vulnerable to model poisoning attacks, where malicious nodes inject fake model updates, jeopardizing the integrity of the global model. This thesis, titled "Uncertainty quantification in Adversarial Federated Learning", introduces novel approaches to improve the privacy and security of distributed machine learning models against such threats.

We explore three distinct methods for quantifying uncertainty in FL models to achieve this goal. The first, Laplace approximation using the Hessian matrix in neural networks, is explicitly applied to detect Distributed Denial of Service (DDoS) attacks within FL settings. This method leverages the second-order derivatives of the loss function to approximate the uncertainty in model predictions, providing a refined understanding of model confidence in the presence of adversarial attacks and enhancing the detection and mitigation of DDoS attacks. The second method introduces an ad-hoc approach using a deep metric learning technique, namely SMELL. This novel method defines a similarity space (S-Space) to represent data more effectively by mapping pairs of elements from the original feature space into this new auxiliary space. The similarity between data pairs is quantified using markers within the S-Space, allowing for intuitive and flexible detection of anomalies and potential threats in FL environments. The third method extends the ad-hoc approach by employing Bayesian neural networks with variational inference. This extension uses Bayesian principles to model uncertainty by treating network weights as distributions rather than point estimations, allowing for a probabilistic interpretation of model outputs and improved resilience against malicious attacks.

By integrating these uncertainty quantification methods, the thesis aims to mitigate the risks of model poisoning attacks, thereby enhancing the robustness, reliability, and security of FL applications. Experimental results demonstrate the effectiveness of these approaches in reinforcing the integrity of distributed machine learning models under adversarial conditions.

Keywords: federated Learning; bayesian neural network; adversarial federated learning; uncertainty quantification.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

The Internet of Things (IoT) has become increasingly impactful, empowering diverse applications [Li et al., 2020a]. Approximately 5.8 billion IoT devices are estimated to be in use in 2022 [Dao et al., 2022]. Moreover, privacy issues are becoming increasingly relevant for distributed applications, as seen recently in the General Data Protection Regulation (GDPR). In this scenario, Federated Learning (FL) has been shown to be a promising approach for training machine learning models collaboratively on distributed devices that share data while addressing privacy restrictions. FL was proposed to guarantee that training data remains on personal devices and facilitates complex models of collaborative ML on distributed devices.

In FL, edges (a.k.a. clients) use local data to train a local model. In the most usual model, combining several local models estimates an ML model cooperatively in an FL server. The server receives the update of the local models, i.e., the model weights or gradients, and aggregates these models. The steps are repeated in multiple rounds until the model reaches a desirable key performance indicator (KPI). Compared to conventional cloud-centric training approaches, the FL model trained at mobile edge networks offers the following advantages [Lim et al., 2020]: Highly efficient use of network bandwidth, Privacy, and Low latency.

In FL, users train a machine learning model collaboratively without sharing their raw data. The process consists of three main steps, as can be seen in Figure 1.1:

– (Step 1) **Task initialization:** The system initializes the local models and necessary hyperparameters for the learning task. Each user prepares their local model using their respective dataset.

– (Step 2) **Local model training and update**: Each user independently trains their local model using their local data. The goal is to find the optimal parameters

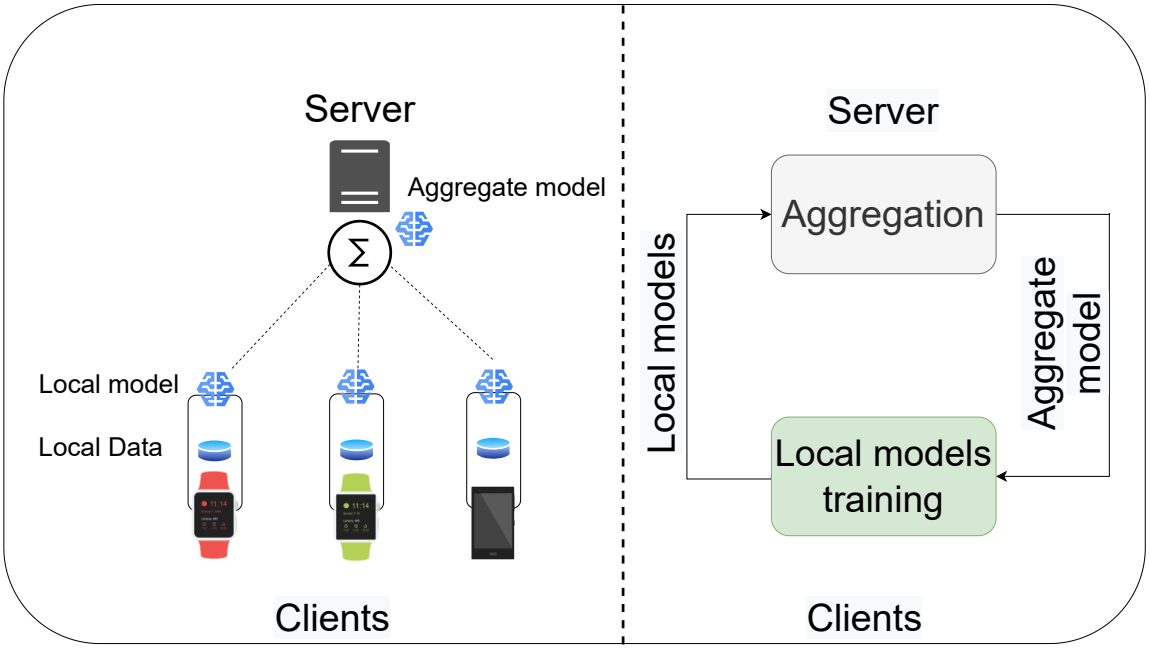


Figure 1.1: Left: Conventional (client-server) federated learning scheme. Right: Training of a simple data flow model in FL.

that minimize the loss function specific to their dataset. This step ensures that each user's model is tailored to their data and captures their local patterns.

– (Step 3) **Model aggregation and update**: The server aggregates the local models from selected participants and generates an updated global model, often referred to as the federated model. The aggregation process typically involves combining the model parameters of the local models. The global model is then sent back to the users for further iterations.

We can implement model aggregation in several ways; for example, FedAvg [McMahan et al., 2017] performs the "arithmetic average" on the local models' weights to obtain the model aggregate in the server. Steps 2 and 3 are repeated iteratively until the aggregate loss function converges or the desired training metric is reached. The iterative nature of the process allows the global model to improve over time by leveraging the collective knowledge of all participating users while maintaining data privacy.

The vulnerability of the system can be exploited by malicious participants, which control the result of the aggregation server model [Chen et al., 2021]. For example, the FL framework is vulnerable to model poisoning attacks in which some malicious end devices try to compromise model security by uploading fake model weights [Chen et al., 2021]. Intuitively, attacks can be defended if the aggregator checks the training process and detects anomalous local models. However, inspecting the entire training process for each end device is hard, especially when the devices are deployed on a large scale.

1.2 Thesis Statement

Considering the context previously described, the main objective of this thesis proposal is to answer the following question:

Research problem: *Can we use uncertain quantification from the Bayesian neural network framework to build robust federated learning applications against model attacks?*

Key Idea: Even presenting interesting results, Neural Networks (NNs) have shortcomings that can limit their applications, such as poor calibration and overconfidence, mainly when the data distribution shifts between training and testing [Guo et al., 2017]. Thus, the Bayesian model is a framework that can quantify uncertainty and use this formalization to develop other learning algorithms [Ghahramani, 2015]. Based on this premise, Bayesian neural networks (BNN) provide a natural way to quantify uncertainty in deep learning. They are less likely to be overconfident or underconfident in their predictions [Gal and Ghahramani, 2016]. A BNN is an exciting approach for FL because the models use Bayesian inference to make predictions based on uncertain inputs [Immer et al., 2021]. Therefore, we can formalize some concepts, such as quantifying the model's knowledge, theoretical misclassification risk, and other properties.

Objectives: The primary objective of this thesis is to analyze FL models against adversarial attacks. To achieve this, we propose the following specific objectives:

- Utilize BNN inference techniques to create quantifiers capable of detecting and mitigating adversarial attacks in FL by identifying malicious model updates.
- Theoretically demonstrate the convergence of FL models in the presence of uncertainty quantifiers, ensuring that the integrity and accuracy of the global model are maintained even under adverse conditions.
- Propose solutions to effectively distinguish between legitimate variations in client data distributions and malicious actions. This is particularly important in FL settings, where data heterogeneity is common, and only a small subset of clients may present data models that significantly differ from the majority due to the heterogeneity.

1.3 Thesis Outline

We organize the remainder of this thesis proposal as follows. First, Chapter 2 presents a novel FL framework for a robust defense against network attacks, with a focus on label poisoning attacks within a supervised learning framework. Chapter 3 proposes a new similarity function for FL applications to tackle model poisoning vulnerability (model replacement). Chapter 4 presents a personalized FL framework for a robust classification of sedentary behavior that explicitly addresses heterogeneous feature distributions and adversarial clients (backdoor attacks). Chapter 5 presents a novel approach to Personalized Federated Learning (PFL) by integrating BNNs to address non-IID data distributions and prediction overconfidence. Chapter 6 combines adaptive confidence-weighted mean aggregation with server momentum to provide convergence guarantees and improve personalization and robustness to heterogeneity and malicious updates. Finally, Chapter 7 offers the concluding remarks and the future directions we intend to explore in this thesis.

1.4 Expected Contributions

The following section summarizes the contributions we expect to present throughout this thesis.

- Advancing the theoretical and practical understanding of adversarial attacks in federated learning. We conjecture that uncertainty quantification can help to detect attacks that are difficult to identify due to their adherence to the model's expected data distribution and lower complexity. The contribution will include a comprehensive framework for executing and evaluating such attacks in federated environments, thereby contributing to the development of more robust defense mechanisms against malicious tactics.

1.4.1 Towards Robust Federated system using Marginal Likelihood to counter label poisoning attacks

Over the last decade, the rapid growth of Internet-connected devices has introduced new security challenges, particularly for low-power IoT devices that remain vulnerable to large-scale attacks such as the Mirai botnet [Antonakakis et al., 2017]. Intrusion Detection Systems (IDS) have thus become essential for network protection [Eliyan and Di Pietro, 2021]; however, traditional signature-based methods have difficulty detecting emerging challengers [Barros et al., 2022a], motivating the use of ML approaches capable of learning complex patterns from network traffic [Khraisat et al., 2019].

Recent advances in FL can enable the training of collaborative models under privacy restrictions [Li et al., 2020a, Barros and Ramos, 2022]. Unlike centralized approaches, FL preserves local data privacy but introduces new challenges, including heterogeneous clients, limited computational resources, and vulnerability to adversarial threats such as Distributed Denial of Service (DDoS) [Li et al., 2022a] and label-poisoning attacks [Toldinas et al., 2022].

In this direction, we propose a novel distributed framework for detecting network attacks in federated environments that considers both privacy and adversarial constraints. Our contributions comprise two levels of defense

(i) Network Security Level: A distributed ML model to detect abnormal traffic patterns in federated networks;

(ii) Model Integrity Level: A robust aggregation method based on marginal likelihood, used to quantify uncertainty and distinguish honest from malicious clients in the presence of label-poisoning attacks.

The primary focus of this chapter is to evaluate if marginal likelihood can effectively detect and mitigate malicious clients in federated settings. To this end, we employ DDoS attacks as a representative use case to validate our approach.

Although the experiments focus on DDoS detection, the proposed framework is designed to be scalable and adaptable to other network scenarios, showing an alternative to mitigate problems in secure FL environments [Li et al., 2021c, Tian et al., 2021, Dao et al., 2022].

1.4.2 Ad-Hoc uncertainty quantification via metric learning to mitigate malicious clients replacement in federated learning

In the previous section, we adapted an existing probabilistic approach from the literature to measure uncertainty in FL scenarios [Immer et al., 2021]. This method provided an approach for evaluating the reliability of local model updates before aggregation. In this chapter, we propose a novel ad-hoc approach designed explicitly for FL environments with heterogeneous data and potential model attacks (model replacement) [Bagdasaryan et al., 2020]. Unlike the previous method, our goal is to develop an uncertainty quantifier, integrated into the aggregation process, to improve both performance and robustness against malicious clients.

Recent work has shown that Deep Metric Learning (DMeL) can capture complex similarity relationships in data [Wang et al., 2019a, Yu et al., 2020], a property especially useful in FL where classes may be missing in local datasets [Chen et al., 2021]. However, existing DMeL methods often face issues such as slow convergence and local optima [Wang et al., 2019a], limiting their effectiveness in secure FL scenarios.

To address these challenges, we introduce a new auxiliary representation space, called *S-space*, and propose an interpretable uncertainty measure tailored for FL aggregation. This ad-hoc quantifier enables the detection and mitigation of malicious clients while preserving accuracy in benign settings.

1.4.3 From Ad-Hoc quantification to reputation-based personalized federated learning under adversarial threats with a sedentary behavior Use Case

In this contribution, we extend our previous analysis to detect backdoor attacks in FL, i.e., where malicious clients embed hidden triggers into local models to manipulate the global model's predictions. Unlike model poisoning, backdoor attacks are more complex to detect because the global model often performs well on clean data while being compromised on specific targeted inputs [Bagdasaryan et al., 2020].

To address this, we conducted experiments in an FL setting using wearable sensor data for detecting sedentary behavior [Li et al., 2021b, Xiao et al., 2021b]. Our goal is

(i) to evaluate the robustness of our ad-hoc uncertainty quantifier in this adversarial environment;

(ii) to propose a reputation-based measure, derived from our uncertainty scores, to detect and mitigate malicious clients performing backdoor attacks.

In this direction, we focus on sedentary behavior as a use case to evaluate our hypothesis, as it exemplifies the challenges of the label concept skew in FL. Sedentary behavior is not defined by a single, uniform activity but by a large clustering class formed by diverse patterns of physical activity. For example, jogging and soccer activities are classified in the same sedentary category (high intensity), but they exhibit different walking patterns. This makes sedentary behavior a particularly suitable case for examining how heterogeneous data distribution affects model performance. This variability reflects the unique ways individuals perform the same activities, making it challenging to develop a single global model that captures sedentary behaviors across all users [Zhu et al., 2023].

Figure 1.2 illustrates the evolution of F1-scores for Human Activity Recognition (HAR) and Sedentary recognition in an FL model (cf. Li et al. [2021b]). The difference in performance between the two tasks highlights the impact of data heterogeneity on model performance.

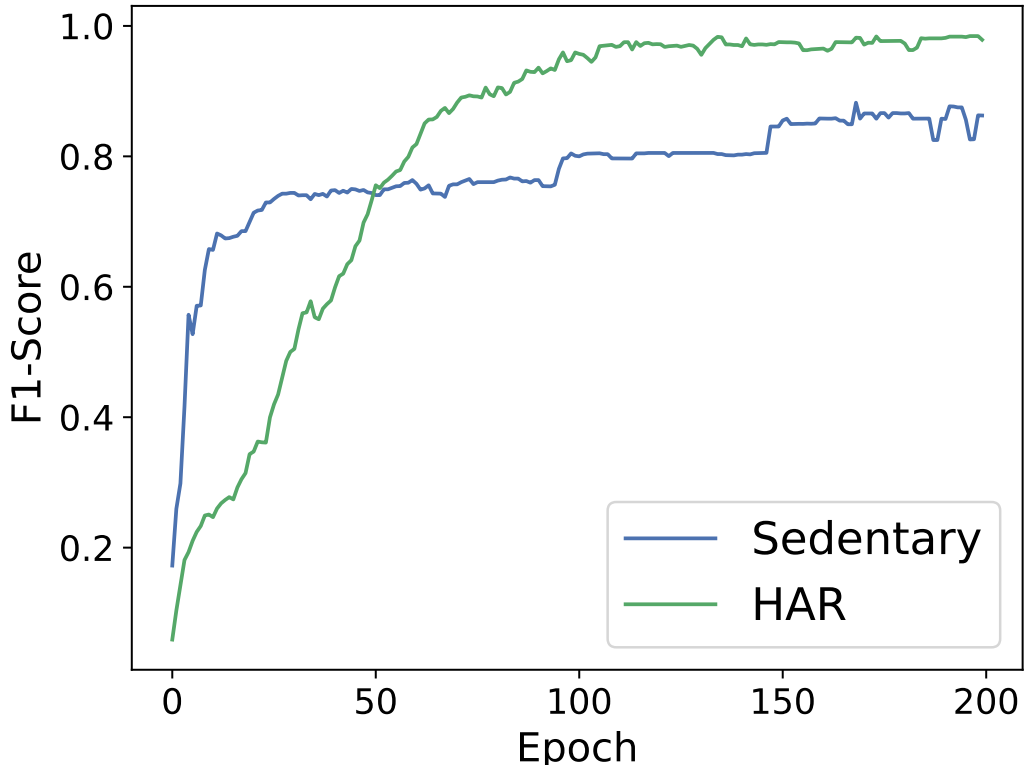
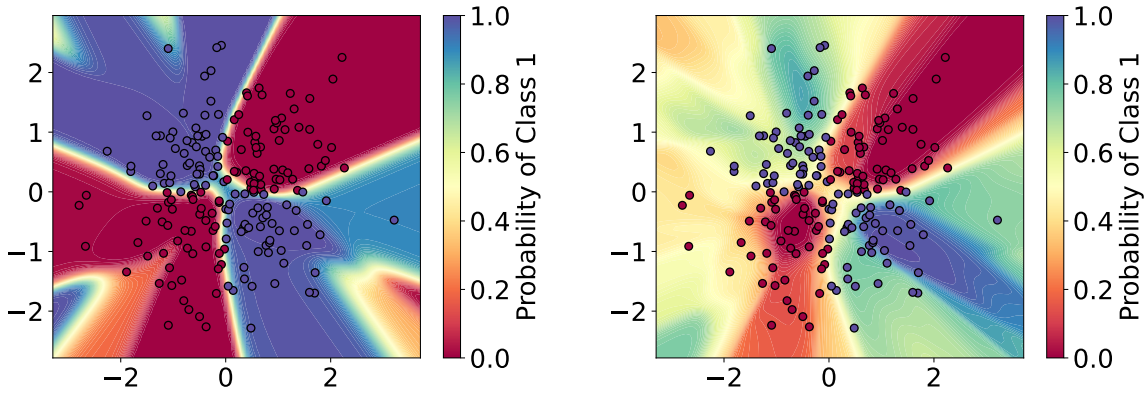


Figure 1.2: F1-Score evolution over epochs for Sedentary and HAR classes for Meta-HAR model [Li et al., 2021b].



(a) Neural Network (b) Bayesian Neural Network

Figure 1.3: Decision regions for a standard neural network vs. a Bayesian neural network on a synthetic dataset. Bayesian models yield smoother boundaries, reflecting better uncertainty calibration.

1.4.4 Personalized federated learning with variational inference for uncertainty quantification

In previous contributions, we progressively advanced uncertainty quantification in FL to proposing an ad-hoc uncertainty quantification [Barros and Ramos, 2022, Barros et al., 2024b, 2025a]. Now, in this chapter, we take a further step by formalizing uncertainty quantification through Variational Inference (VI) to obtain theoretical insights into model confidence in PFL.

FL enables collaborative model training on decentralized data while preserving privacy [McMahan et al., 2017], yet real-world FL applications often suffer from non-IID data distributions that degrade both global and local model performance [McMahan et al., 2017, Huang et al., 2021]. PFL mitigates this by tailoring models to local data using methods such as global model personalization [Karimireddy et al., 2020, Li et al., 2021d], federated meta-learning [Fallah et al., 2020, Dinh et al., 2020], and parameter decoupling [Arivazhagan et al., 2019, Liang et al., 2020].

However, standard NNs often produce overconfident and poorly calibrated predictions under data heterogeneity [Zhang et al., 2016, 2021a]. BNN, on the other hand, offers a probabilistic framework to model uncertainty and has shown promising results in continual learning and robustness [Immer et al., 2021]. Figure 1.3 illustrates how Bayesian networks produce smoother decision boundaries, reflecting improved uncertainty awareness compared to standard neural networks.

In this contribution, we propose a personalized Bayesian Federated Learning (BayesPFL) framework that integrates VI into PFL to quantify model uncertainty at both local and

global models explicitly. To achieve personalization, each client updates its local VI parameters by reusing the global distribution from the server and balancing the Kullback–Leibler (KL) divergence between the local posterior distribution and the server variational parameters. This strategy improves the upper bounds on this KL divergence compared to traditional distributed BNNs [Zhang et al., 2022b, Chen et al., 2023].

1.4.5 Rethinking variational inference in similarity spaces to personalized federated learning under malicious scenarios

FL shows challenges due to data heterogeneity, usually each client often follows a non-IID distribution [Zhang et al., 2022b]. PFL addresses this by adapting the global model to client-specific data; however, discriminating between legitimate outlier updates (caused by heterogeneity) and malicious updates (introduced by adversaries) remains an open problem [Li et al., 2021d]. In this chapter, we enhance our Bayesian Personalized FL (BayesPFL) framework and apply it to federated environments with malicious clients. Our goal is to integrate Bayesian inference, uncertainty quantification, and personalized learning to enhance model reliability and robustness in the presence of heterogeneous data distributions and adversarial scenarios.

Therefore, we extend our previous framework by rethinking VI in our auxiliary representation space (S-space). Our goal is (i) to quantify uncertainty to enhance personalization, and (ii) to leverage this uncertainty to separate benign heterogeneity from adversarial behavior. The resulting formulation yields theoretical guarantees that connect an optimal variational latent space with improved robustness: uncertainty helps to identify and mitigate malicious updates without discarding informative but atypical client updates.

From a theoretical perspective, we derive bounds for reparameterized gradient estimators and stability conditions for proximal VI with exponential parameterization, and use these insights in federated settings via client-drift and server-momentum analyzes.

Chapter 2

Towards Robust Federated system using Marginal Likelihood to counter label poisoning attacks

Network traffic monitoring is an essential task to understand network behavior and component status. In that context, FL has emerged as a promising approach for network traffic monitoring-based defense systems to provide a distributed and scalable way to train a global machine learning model without requiring data sharing among clients. However, most existing FL methods assume that a federated environment has only honest clients; for instance, in supervised learning, they assume that all labels were assigned truthfully. To address such issue, this study introduces a novel FL framework for a robust defense against network attacks, focusing on label-poisoning attacks within a supervised learning framework. Our approach distinguishes itself from existing literature by detecting network attacks and identifying and mitigating the influence of malicious clients intent on undermining the federated model's integrity. Specifically, our framework incorporates a novel aggregation approach that leverages marginal likelihood to effectively weigh contributions from honest and malicious clients in the presence of label-poisoning attacks. By quantifying the data's adherence to the model using marginal likelihood, we enhance the framework's ability to detect and mitigate the influence of malicious clients. While our model can detect various network attacks, this chapter concentrates on DDoS attacks as a proof of concept. We evaluate the effectiveness of our proposal through three real-world DDoS datasets and show that it outperforms existing techniques.

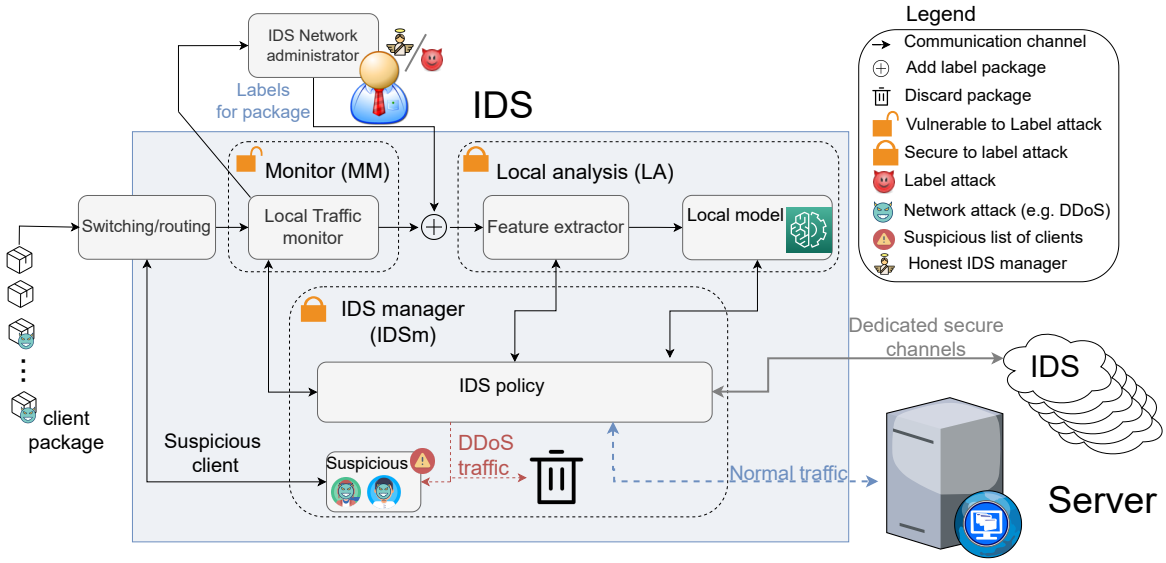


Figure 2.1: Our IDS architecture design adopts the FL system for network attack detection.

2.1 Our Proposal

This chapter proposes a new framework to identify network attacks in FL systems prone to attacks of label poisoning. The framework proposes a multi-step approach for defending against network attacks within a FL context. The framework involves three main phases: (i) Self-Supervised Learning (SSL), where an autoencoder model is employed to learn a label-agnostic representation of input data; (ii) Fine-tuning model, where the encoder's learned representation is utilized for a DDoS classification task, and a new classifier is trained based on these representations, while the encoder's weights are kept frozen; and (iii) Marginal likelihood to quantify data adherence to the local model, enabling the detection of malicious nodes attempting to attack the model.

The resulting global model, resilient to poisoning label attacks, is then aggregated at the server for monitoring and shared with clients, thus creating a robust defense against network attacks while maintaining privacy in an FL setting. This framework's novelty lies in its focus on federated training defense, addressing network and label attacks, and its adaptability to various scenarios beyond FL.

2.1.1 Framework Rationale

Bansal et al. [2020] proposed an upper bound on the generalization gap for classi-

fication in neural networks models that can be quantified in three parts: *Robustness gap*, *Rationality gap* and *Memorization gap*. Thus, typically for neural networks, the robustness and rationality gaps are small [Bansal et al., 2020]. A narrow robustness gap implies that adding a limited amount of wrong labels causes minor degradation in performance. In contrast, the small rationality gap suggests that getting the wrong label is not better than getting no label at all. However, modern classifiers are heavily over-parameterized [Zhang et al., 2021c]. Thus, these models easily fit a random labeling of the training data and, consequently, present a high Memorization gap, as seen in Zhang et al. [2017].

To get around this problem, Bansal et al. [2020] presented evidence that the memorization gap is negligible if the simple classifier has low complexity, regardless of the complexity of the representation. The memorization gap tends to zero if the simple classifier is under-parameterized.

We hypothesize that our approach mitigates the high-memorization gap problem and, consequently, the label-poisoning attack in FL environments. Although our self-supervised model can be highly parameterized, the label-poising attack does not affect the model's performance. We further train an under-parameterized classifier that tends to minimize the memorization gap. Ultimately, we estimate the quantification of data adherence associated with the under-parameterized classifier to detect label-poisoning attacks.

2.1.2 System Architecture design

Figure 2.1 depicts our system architecture, where the Intrusion Detection Systems (IDS) connects with the outside IoT system (a.k.a. clients) through a communication module (switching/routing module) used to handle the ingress/egress traffic.

The monitor module (MM) focuses on monitoring and analyzing packets received by the IDS from the federated clients. Its primary function is to generate traffic statistics reports (e.g., traffic protocols, package flow volume, and source/destination ports). MM periodically delivers a summary of the information to the IDS Manager. In addition, this module also forwards ingress traffic to the IDS network administrator, who uses this information (along with prior knowledge of the network's operation) to label traffic (e.g., identify it as DDoS). This report and the labels are forwarded to the feature extractor module to make the network attack training/prediction task.

Given the federation's potential vulnerability to attacks, the MM module is considered a risk area for manipulation by malicious network administrators. Such administrators could exploit this access to deliberately mislabel data (a label-poisoning attack),

compromising the system's integrity.

The local analysis module (LA) monitors the health of packages analyzed by MM. This module extracts features that can help the network attack classification task. With these features (along with a percentage of the data labeled by the IDS network administrator), LA uses the local model (Section 2.1.3) to perform the training (and later prediction) of the client packages. We collaboratively trained this local model using the FL paradigm (Section 2.1.5).

The IDS manager (IDSm) has the policy module that communicates with the two previous modules and coordinates the IDS workflow, i.e., based on the MM and LA, the appropriate dispatches manager policy to the malicious/benign package. Thus, with the information provided by the two previous modules, IDSm can send the packets to the server or discard the suspicious packets in case of an attack and update the list of clients suspected of being malicious.

Communication between IDSm and other IDS components happens via secure channels/protocols supported by the networks (e.g., the OpenFlow protocol). Furthermore, LA and IDSm modules are safe (protected from access or tampering) and reliable (they perform operations as expected).

2.1.3 Our framework

Our framework is based on three modules, as depicted in Figure 2.2. The first module performs a pre-training with a self-supervised learning (SSL) task; the second fits a classifier (e.g., single neural network layer) on the representation found by the SSL model and the labels; and the third estimates a quantifier for model selection agnostic to the training set. In the following, we detail each module.

2.1.3.1 Pretext task

Unsupervised learning (UL) refers to learning methods that do not need human-annotated labels. Among the UL approaches, a compelling case is when the model generates a pseudo-label for each input data, called self-supervised learning. For example, in autoencoder architecture, the pseudo-label equals input data. SSL models can be divided into two parts: (i) the encoder model, responsible for mapping the input data into an

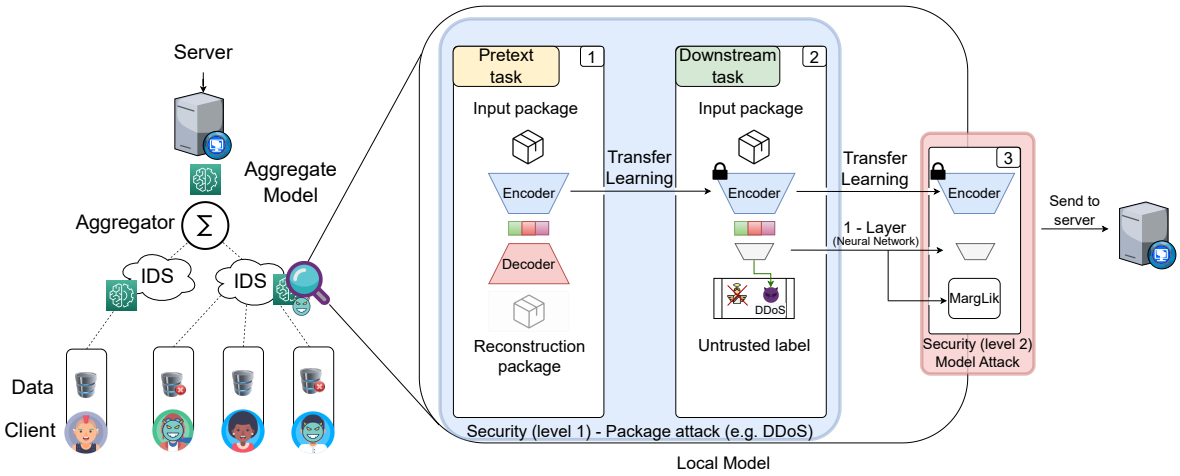


Figure 2.2: Our proposal to detect network attack in FL environment. The blue square represents the first level of security, while the red square represents the second security level. At the first level, we use the SSL model in two steps: Initially, we train the pre-task to obtain the weights of the encoder function (in this chapter, we use an autoencoder). In our method, we only train the encoder function in this step. Secondly, we use the representation obtained by the encoder function to train the network attack classifier (e.g., DDoS). In this step, the encoder function weight is frozen (represented with a lock in this figure). Finally, we estimate the data's adherence to the model and infer if the client is malicious or honest.

embedded space (named *pretext task*), and (ii) the projection head, trained to perform the inference using embedded space (named *downstream task*). In this chapter, we use an autoencoder as the SSL model.

Formally, let the set $\mathcal{X} = \{(\boldsymbol{x}_i)\}_{i=1}^{v}$, with $\boldsymbol{x}_i \in \mathbb{R}^m$, be defined in an m-dimensional feature space with v elements. The set $\mathcal{X}$ was called *original feature space*, and the neural network (a.k.a. encoder) function $e(.\,, \boldsymbol{\theta}) : \mathcal{X} \rightarrow \mathcal{Z}$ was parameterized by a weight set $\boldsymbol{\theta}$, where $\mathcal{Z}$ is the *latent feature space*. Thus, the set $\mathcal{Z} = \{(\boldsymbol{z}_i)\}_{i=1}^{v}$ is the latent feature space, where $\boldsymbol{z}_i = e(\boldsymbol{x}_i, \boldsymbol{\theta}) \in \mathbb{R}^n$ and $m > n$ (*latent feature space* dimension).

Similarly, the decoder (a.k.a. projection head) function can be defined as the inverse encoder function $e^{-1}(.\,, \boldsymbol{\theta}') : \mathcal{Z} \rightarrow \mathcal{X}$ where $\boldsymbol{\theta}'$ is a set of weights for the decoder. So, the optimal weights for the autoencoder can be estimated by minimizing the mean squared error function $\frac{1}{v}\sum_{i=1}^{v} \|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_2^2$, where $\boldsymbol{x}'_i = e^{-1}(e(\boldsymbol{x}_i\,, \boldsymbol{\theta})\,, \boldsymbol{\theta}')$ is the pseudo-label associate to $\boldsymbol{x}_i$.

Initially, we train our local model for self-supervised learning *pretext tasks*, i.e., we do not use any labels (only pseudo-label) in this training step. In this step, we are interested in obtaining a new label-agnostic representation of the data. At the end of this training, the model $e(.\,, \boldsymbol{\theta})$ maps the input data to a new representation space (*latent feature space*). As shown in Section 2.1.2, the malicious client cannot change the model training step, so it is impossible to change the trained model's weights. This attack consists of changing only the local data labels.

For reconstruction tasks, the autoencoder encodes the data to extract its most relevant features and projects them in $\mathcal{Z}$. Also, note that this step is performed only on the client (i.e., local model).

2.1.3.2 Downstream task

Since *pretext task* methods enable learning general-purpose representations, many approaches in the literature use this pre-trained model and fine-tuning (with human-annotated labels) to provide high performance and efficient data learning of *downstream tasks*.

After the *pretext task* training, we use the encoder module to get a new representation of the local model input. This new representation is fixed during the rest of the local training, i.e., after the *pretext task*, our model freezes the encoder's weight set $\boldsymbol{\theta}$. With the new representation, we discard the projection head module (the decoder) and replace it with a single-layer neural network to perform a network attack classification task (*downstream task*). In this step, the encoder function weight is frozen, i.e., we used the encoder weights estimated by self-supervised learning settings.

Finally, we use a fully connected layer $f(z_i, \boldsymbol{\Theta}) = [\boldsymbol{W} z_i + \boldsymbol{b}]_+$, where the operator $[.]_+ = \max(0, .)$ is the ReLU activation function, $\boldsymbol{W} \in \mathbb{R}^{c \times n}$ is the weight matrix, and $\boldsymbol{b} \in \mathbb{R}^c$ is the bias vector. For the rest of the chapter, we will assume that $\boldsymbol{\Theta} = \{\boldsymbol{W}, \boldsymbol{b}\}$. Thus, we construct a neural network classifier designed for network attack detection, exemplified by using $\Pr(y_i|x_i, \boldsymbol{\Theta}) = \texttt{Categorical}(y_i|\texttt{softmax}(f(z_i, \boldsymbol{\Theta})))$ in a binary problem classification framework (attack or non-attack), where $f(.,, \boldsymbol{\Theta})$ maps the *latent feature space* inputs and parameters $\boldsymbol{\Theta}$ to output logits. Specifically, for each $\boldsymbol{z}_i \in \mathcal{Z}$, there is an associated network attack label $y_i \in \mathcal{Y}$, focusing on DDoS as a primary example within the broader spectrum of network attacks. Therefore, our model's weights are $\boldsymbol{\theta}, \boldsymbol{\Theta}$.

In this way, we map the encoding found by the encoder into a discrete probability vector that quantifies the chance that the analyzed input data is due to a network attack (e.g., DDoS). In this step, we only use the label associated with the model input to train the local model.

One of the distinguishing characteristics of our approach is to consider that in addition to the network attack (system attack), the federation is susceptible to a model attack (label attack). Still, in the second step, our framework considers that the label used in the DDoS classifier training is unreliable, i.e., we consider that this label can be tampered with to harm the federation (as seen in Section 2.1.2).

2.1.3.3 Laplace approximation

The last step is quantifying data adherence to the local model. This crucial phase detects malicious nodes attempting to attack the federation's model.

We consider a Bayesian neural network $f(., \boldsymbol{\Theta})$ with prior $\Pr(\boldsymbol{\Theta})$ and likelihood $\Pr(\mathcal{D}|\boldsymbol{\Theta}) = \prod_{i=1}^{v} \Pr(y_i|z_i, \boldsymbol{\Theta})$. We assume that the data examples are independent and identically distributed from $\Pr(\mathcal{D}|\boldsymbol{\Theta})$, i.e., observations y are independent given inputs z (*latent feature* vector). Hence, the posterior is

$$\Pr(\boldsymbol{\Theta}|\mathcal{D}) = \frac{\Pr(\mathcal{D}|\boldsymbol{\Theta})\Pr(\boldsymbol{\Theta})}{\int \Pr(\mathcal{D}|\boldsymbol{\Theta})\Pr(\boldsymbol{\Theta})d\boldsymbol{\Theta}},$$

where $\Pr(\mathcal{D}) = \int \Pr(\mathcal{D}|\boldsymbol{\Theta})\Pr(\boldsymbol{\Theta})d\boldsymbol{\Theta}$ is the marginal likelihood intractable for common BNNs, which is we use the Laplace approximation for approximating the posterior distribution with a Gaussian centered at the maximum posterior estimate for fast approximate Bayesian inference. Thus, Laplace approximation uses second-order Taylor expansion as

$$\log p(\boldsymbol{\Theta}, \mathcal{D}) \approx \log p(\boldsymbol{\Theta}^*, \mathcal{D}) - g_{\boldsymbol{\Theta}^*}^{\top}(\boldsymbol{\Theta} - \boldsymbol{\Theta}^*) - \frac{(\boldsymbol{\Theta} - \boldsymbol{\Theta}^*)^{\top} H_{\boldsymbol{\Theta}^*}(\boldsymbol{\Theta} - \boldsymbol{\Theta}^*)}{2},$$

where $g_{\boldsymbol{\Theta}^*} = -\nabla_{\boldsymbol{\Theta}} \log \Pr(\boldsymbol{\Theta}, \mathcal{D})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}^*}$ (vanishes by assumption that $\boldsymbol{\Theta}$ is a maximum a posterior estimation $\boldsymbol{\Theta}^*$) and $H_{\boldsymbol{\Theta}^*} = -\nabla^2_{\boldsymbol{\Theta}\boldsymbol{\Theta}} \log \Pr(\boldsymbol{\Theta}, \mathcal{D})|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}^*}$ are the gradients and Hessian of the negative log joint distribution at $\boldsymbol{\Theta}^*$, respectively. Finally, the quantifier known as the **Laplace marginal likelihood**[1] approximation $\int \Pr(\mathcal{D}|\boldsymbol{\Theta})\Pr(\boldsymbol{\Theta})d\boldsymbol{\Theta} \approx \int \exp(\log \Pr(\mathcal{D}, \boldsymbol{\Theta}))\, d\boldsymbol{\Theta}$ is

$$\begin{aligned} \Pr(\mathcal{D}) &= \int \exp(\log \Pr(\boldsymbol{\Theta}, \mathcal{D}))d\boldsymbol{\Theta} \\ &\approx \int \exp\left(\log \Pr(\boldsymbol{\Theta}^*, \mathcal{D}) - \tfrac{(\boldsymbol{\Theta}-\boldsymbol{\Theta}^*)^{\top} H_{\boldsymbol{\Theta}^*}(\boldsymbol{\Theta}-\boldsymbol{\Theta}^*)}{2}\right) d\boldsymbol{\Theta}. \end{aligned}$$

As we can see in literature, the result

$$\Pr(\mathcal{D}) = \Pr(\Theta^*, \mathcal{D})(2\pi)^{\frac{p}{2}} \|H_{\Theta^*}\|^{-\frac{1}{2}} \tag{2.1}$$

follows from completing the square and solving a Gaussian integral [Immer et al., 2021].

To optimize the network parameters $\boldsymbol{\Theta}^*$, we perform regular neural network training on the maximum a posterior (MAP) objective using stochastic optimizers like, e.g., SGD, to estimate $\Pr(\boldsymbol{\Theta}^*, \mathcal{D})$ [Kristiadi et al., 2020, Immer et al., 2021]. In this chapter, we use in last neural network layer [Kristiadi et al., 2020] $f(.\,, \boldsymbol{\Theta})$ efficient implementations of H_{Θ^*} for Laplace neural network approximation proposed in Daxberger et al. [2021].

We observed that the Marginal Likelihood $\Pr(\mathcal{D})$ is a reasonable candidate for performing the task because it is estimated solely on the training data [MacKay, 1992].

[1] We will refer to the marginal likelihood for the rest of the chapter

It provides a measure of the model's fit (after the training phase) to the data without compromising the privacy of the individual client's data.

Finally, users send local models to the server at the end of these three training steps. Next, the server aggregates the local models, obtaining an aggregated global model to monitor network attacks. The server has a model capable of detecting network attack packets (Section 2.1.2), preserving the privacy restrictions established by FL environments. In addition, the server has information about the Marginal Likelihood $\Pr(\mathcal{D})$ client data, making the aggregate global model robust to the poisoning label attack. Moreover, the server sends the aggregated model to the clients, who can again carry out the training proposed by our framework.

2.1.4 Occam razor and marginal likehood

The marginal likelihood automatically encapsulates a notion of Occam's Razor. To illustrate this, we estimate Laplace approximation to find a Gaussian approximation to a probability density defined over a set of continuous variables.

We can consider the log of the Laplace Marginal Likelihood (LML) in Equation 2.1 as

$$\log p(\mathcal{D}) \propto \log p(\mathcal{D}, \boldsymbol{\Theta}^*) + \underbrace{\frac{M}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{H}_{\boldsymbol{\Theta}^*}|}_{\text{Occam factor}}. \tag{2.2}$$

The relationship between Occam's Razor and Laplace Approximation is based in the theory of Bayesian inference. Laplace Approximation is a technique that allows us to approximate a probability distribution with a Gaussian distribution around the maximum a posteriori point, which is often interpreted as the most likely solution to a given modeling problem.

Occam's Razor, on the other hand, is a philosophical principle that states that if there are several possible explanations for a given set of observations, the simplest explanation is the most likely. In Bayesian inference theory, this translates into the fact that when making inferences about a model, we should prefer simpler and less complex models unless there is clear evidence.

When we consider Equation 2.2 as our loss function for training the neural network, we realize that maximizing the fit of the model's marginal likelihood corresponds to increasing the value $\log p(\mathcal{D}, \boldsymbol{\Theta}^*)$ and minimizing the complexity term $\log|\mathbf{H}_{\boldsymbol{\Theta}^*}|$. The complexity term depends on the log determinant of the Laplace posterior covariance. Therefore, if $\log|\mathbf{H}_{\boldsymbol{\Theta}^*}|$ is large, the model strongly correlates with the training data [Im-

mer et al., 2021]. So, maximizing the Laplace $\log p(\mathcal{D})$ requires maximizing the data fit while minimizing the training sample correlation.

The Laplace Approximation implements Occam's Razor as we make the simplest possible assumption about the posterior distribution. Furthermore, the Laplace Approximation has been used in many applications, including training machine learning models in federated environments where data privacy is critical. In these scenarios, it is essential to have a model that can generalize well from a small dataset. The Laplace Approximation can help achieve this goal by providing a way to regularize the model and avoid overfitting.

2.1.5 FL Attack model: Poisoning Label Attack

In this chapter, we propose a modified version of the FedAvg approach to estimate the *federated* global model Θ_F as

$$\Theta_F \leftarrow \Theta_F + \eta \sum_{i=1}^{N_{sel}} \left[\mathbb{1}[t_u \geq \Pr(\mathcal{D}_i) \geq t_l] \Pr(\mathcal{D}_i)(\Theta^*_{\mathcal{D}_i} - \Theta_F)\right], \tag{2.3}$$

where $\Theta^*_{\mathcal{D}_i}$ is a maximum a posterior estimate of client i; N_{sel} is the amount of client selected in the training round; η is the learning rate; $\mathbb{1}$ is the indicator function, i.e., $\mathbb{1}[.] = 1$ if $[.]$ is true and 0 otherwise; t_u and t_l are thresholds, and $\Pr(\mathcal{D}_i)$ is the Marginal Likelihood[2]. This chapter uses the Laplace Marginal Likelihood as in eq (2.1).

In a **federated poisoning attack**, a malicious user sends a local model to the server designed to degrade the aggregate model to some arbitrary goal defined by the attacker (e.g., maximize the classification error). Typically, we consider the extreme scenario where the malicious user knows the model used by the federation and has a subset of labeled data.

More specifically, **federated label-poisoning attack** considers attacks by malicious users who train local models with manipulated labels to degrade the aggregate model. In this case, the attack is on the aggregate model, regardless of the application. These assumptions help assess the model's robustness to extreme attacks. This analysis can be beneficial in applications that need certain levels of assurance about the FL environment's performance.

[2]For the sake of notation clarity, we did not represent the local batches of the FL process in Equation.(2.3), although the real process occurs every a certain number of local batches.

2.2 Experiment Design

In this section, we can summarize our findings in the following research questions:

RQ 2.1. *How can aggregation metrics, such as Marginal Likelihood, be leveraged to gain insights into the behavior of individual clients and the entire network? Can this metric effectively aid in the identification and diagnosis of sources of model vulnerability?*

RQ 2.2. *To what extent does the proposed Marginal Likelihood aggregation approach demonstrate the ability to generalize DDoS detection without malicious users in a FL environment?*

RQ 2.3. *How does the performance of the proposed Marginal Likelihood aggregation method compare to other established robust aggregation techniques explicitly designed to counteract the influence of malicious clients?*

RQ 2.4. *How does the performance of the proposed method evolve when faced with varying proportions of labeled and unlabeled data during the training process?*

In the following section, we discuss the results based on each research question.

2.2.1 Dataset and Neural Network architecture

This section provides an overview of several DDoS datasets used in this chapter. Distributed Denial of Service (DDoS) attacks have emerged as a significant challenge, demanding comprehensive research to understand and mitigate their impact. By employing these datasets, researchers can assess the efficacy of existing defense mechanisms and devise innovative solutions that adapt to the dynamic nature of DDoS attacks. The datasets encapsulate distinctive characteristics, including attack types, attack sources, traffic volumes, and target demographics. These attributes are necessary for a comprehensive approach to studying DDoS attacks.

CSE-CIC-IDS2018 [Sharafaldin et al., 2018] is a dataset that contains benign packets and usually DDoS attacks. It resembles real-world data and includes the network traffic results in PCAP format, labeled flows containing the timestamp, source, destination IPs, source and destination ports, protocols, and attacks. Finally, in the feature extraction process from the raw data, the authors extracted 80 traffic features (e.g., Mean Packet Length and Variance of Request/response time difference).

NF-UNSW-NB15-V2 [Sarhan et al., 2022] dataset, developed by the University of New South Wales (UNSW), Australia, comprises a diverse collection of real-world network traffic data designed for evaluating intrusion detection and prevention mechanisms. UNSW-NB15 covers ten distinct attack categories, and It includes 49 network-based features extracted from payload data and header information, enabling in-depth analysis of network interactions. In addition, Each instance is labeled as either standard or malicious traffic, with 175,341 instances of attacks and 2,201,684 instances of regular traffic; the dataset offers a substantial volume of data for robust evaluations.

NF-ToN-IoT-V2 [Sarhan et al., 2022] dataset is a comprehensive collection derived from the ToN-IoT pcap files, transformed into NetFlow records. With 16,940,496 data flows, it encapsulates diverse IoT network behaviors categorized into specific classes. In this chapter, we consider only benign and DDoS packages. Notably, after preprocessing, 32.64% of flows represent attacks, while 67.36% are benign.

Since the dataset comprises tabular data, we use an autoencoder as a self-supervised model. We initialize all weights of the autoencoder layers from a zero-mean normal distribution and biases as outcomes of a normal distribution with mean 0.5 and standard deviation 10^{-2}, following [Koch et al., 2015]. We separate the dataset into training, validation, and test, with proportions of 80%, 10%, and 10%, respectively.

We set the encoder dimensions to m–2048–512 (the decoder is a symmetrical neural network). For the experiments, the projection head module is 256-n, where m is the number of features of the input data and n is the number of classes. All layers are fully connected, and we adopt the Rectified Linear Unit as the activation function.

In addition, we use the mini-batch Stochastic Gradient Descent (SGD), with a learning rate of 10^{-2}. We set up all baselines with the hyperparameters recommended in their original proposals.

2.2.2 Experimental Scenario

We use the framework Flower to develop solutions and applications in FL. We perform a non-iid data distribution among the users in this experiment. We first split the dataset into training and testing subsets in our experimental setup. We randomly distribute the train data among individual clients non-uniformly (quantity-based label imbalance) [Li et al., 2020b, 2022b]. These clients independently train their models using local data while the central server aggregates their updates. After training, the aggregated

model is evaluated using the separate testing subset. This process allows us to assess the model's performance while adhering to data privacy and decentralized learning principles inherent in FL scenarios.

We employ a server and 50 clients to evaluate our model, and we train our method with an NVIDIA Quadro RTX 6000 GPU (24 GB) for a total of 400 epochs (server). For each training round, the server selects five clients to train the local model, i.e., each model is trained using only the local data. Finally, we aggregate the models on the central server, which forwards the aggregated model to the clients.

For simplicity, we consider uncorrelated noise for the label-poisoning attack (all labels are equally likely to be corrupted). We use ρ as the system noise level (fraction of noisy clients). Each malicious client corrupts $\gamma\%$ of its labels by flipping them. In our experiments, inspired by Hendrycks et al. [2018], we used $\gamma = 40\%$. Also, note that for DDoS attack scenarios, we typically consider a *large percentage* (ρ) of malicious users for a stress scenario [Dao et al., 2022].

2.3 Experimental Results

In this section, we present the results of our proposal. Specifically, we analyze the Marginal Likelihood quantified in an FL environment with malicious users (Section 2.3.1); we conduct a performance evaluation comparison in two scenarios: (i) considering all datasets are labeled (Section 2.3.2) and (ii) a more realistic scenario considering only small labeled dataset (Section 2.3.3).

2.3.1 Marginal Likelihood Analysis

We start the model evaluation by qualitatively assessing the performance of the Marginal Likelihood quantifier used in this chapter. In identifying malicious users in FL, marginal likelihood can be a quantifier for model selection agnostic to the training set. On the other hand, neural networks are heavily over-parameterized [Bansal et al., 2020]. In this chapter, we hypothesize that reducing the neural network representation and consequently reducing *memorization gap* (with self-supervised learning) mitigate label-poisoning attacks in FL settings. Hence, we use marginal likelihood to select models for each client to compare how well the models fit the observed data. Consequently, when used

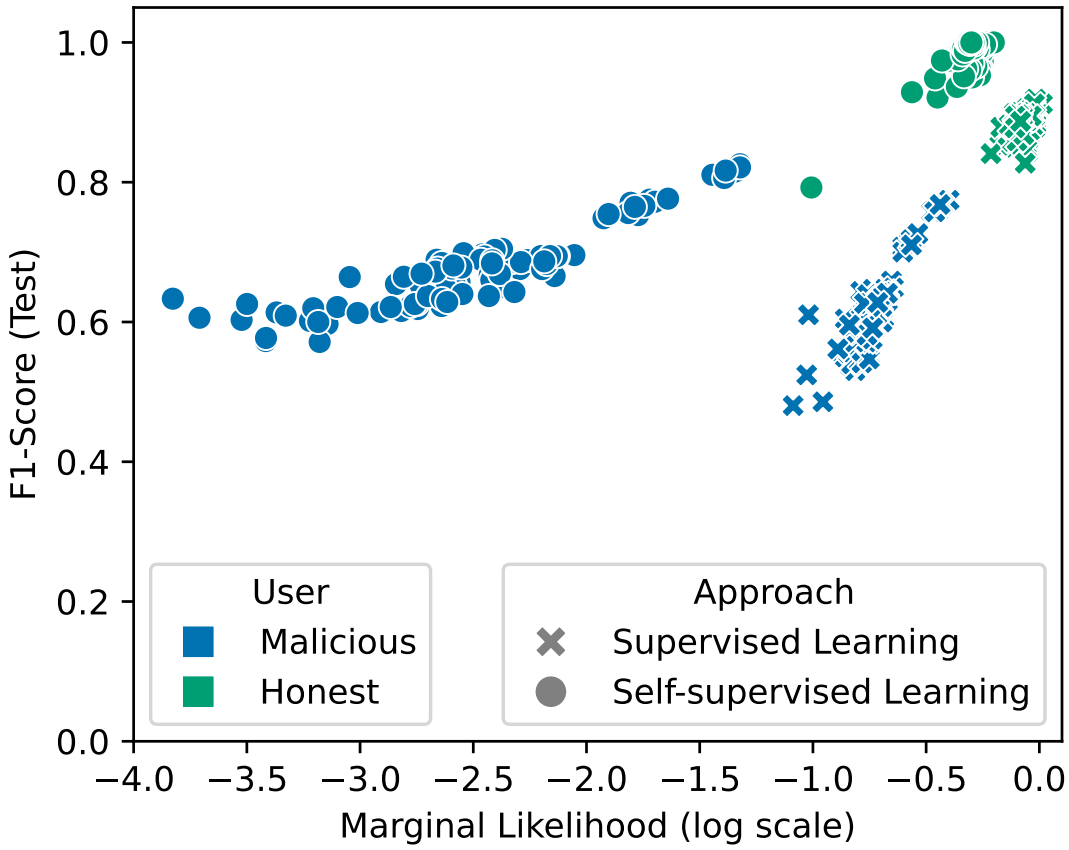
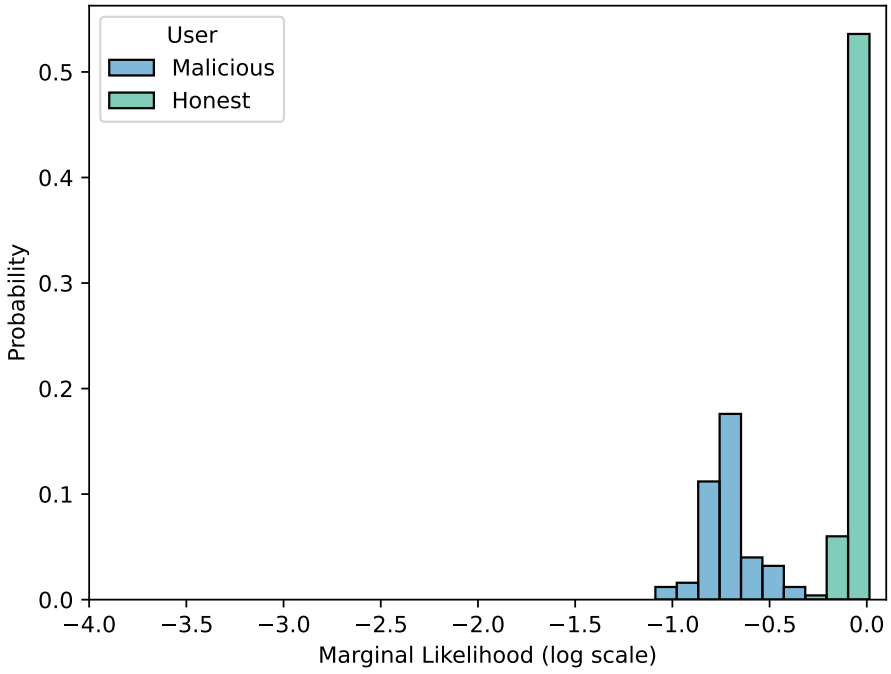


Figure 2.3: In practice, we do not have malicious/honest client annotated data and thus cannot train a classifier. To address this, the proposed model aggregates clients' models weighted by the marginal-likelihood quantifier. We observe that the self-supervised approach assigns much lower weights to malicious clients than the supervised approach.

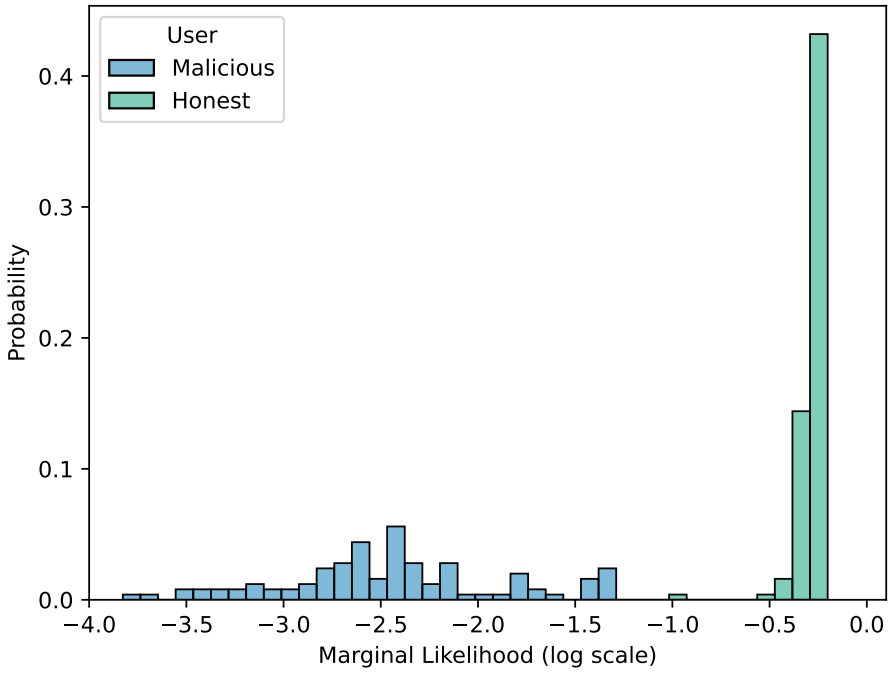
with self-supervised learning approaches, marginal likelihood will be more discriminative against corrupt models.

In this experiment, we split **NF-UNSW-NB15-V2** dataset as described in Section 2.2.2. This evaluation uses only the testing data, and we estimate the Marginal Likelihood quantifier (Equation. (2.1)) using the training data. We use this setting to evaluate how well marginal likelihood describes the model performance in a FL scenario (**RQ 2.1**), and in consequence, we can not access the test data to estimate marginal likelihood. Therefore, we compare the performance of self-supervised and supervised approaches and analyze the distribution of marginal likelihood values for each learning approach.

Figure 2.3 illustrates differences in the *memorization gap* effect of the Self-supervised Learning and the Supervised Learning approaches in a scenario where $\rho = 20\%$ of users send $\gamma = 40\%$ of noisy labels. The points represent a model update of a client of the federation. In this experiment, we calculated the F1-Score using the test dataset, while we



(a) Supervised Learning



(b) Self-supervised Learning

Figure 2.4: Marginal Likelihood distribution (negative log scale) of honest and malicious clients for (a) supervised learning local model and (b) self-supervised learning local model.

estimated the Marginal Likelihood using only the local users' dataset. Observe that the F1-Score of the Self-supervised learning is higher for a given Marginal Likelihood value. We also observe that with a lower log Marginal Likelihood value, the Self-supervised Learning approach tends to imply a higher F1-Score. It means that, typically, even when the data is less adherent to the model, Self-supervised Learning tends to outperform Supervised Learning [Gidaris et al., 2018, Zhang et al., 2021c]. The self-supervised learning approach yields a lower adherence of the data to the model, but it is more robust regarding the label-poisoning attack. The main point here is that Self-supervised learning is prone to the label-poisoning attack only in the last layer (classifier), which is a model with lower capacity (less prone to the *memorization gap*). At the same time, Supervised Learning is prone to the label-poisoning attack in the whole network (encoder and last layer), i.e., a higher capacity model, and consequently, overfitting malicious data.

As seen in Bansal et al. [2020] and Zhang et al. [2021c], the authors show this overfitting effect, i.e., when training a neural network with incorrect labels. Specifically, we observed the counterintuitive behavior in neural networks to be subjected to such conditions. Networks perform well on the training data set, reaching an accuracy metric close to 100%. However, we observed a deterioration in performance when exposed to a test dataset, resulting in substantially degraded accuracy metrics (below 30% for accuracy metric). The phenomenon underscores the need for a cautious interpretation of training accuracy to indicate model effectiveness. The misleading training accuracy in label-attack settings does not translate to performance robustness [Zhang et al., 2021c]. Hence, we can not use the F1-Score for the training data.

As we can see (also shown by other works [Daxberger et al., 2021, Immer et al., 2021]), a property of the Marginal Likelihood is that given a model, the update with a lower (higher) Marginal Likelihood tends to lead to a lower (higher) F1-Score. This property evidences the positive correlation between the F1-Score and the Marginal Likelihood. This means that the marginal likelihood is a quantifier that can be used to compare the quality of different updates in a given model by using only the training dataset, i.e., it is agnostic to the test/validation dataset. At the same time, the F1-Score requires a test dataset to be computed (not available due to privacy concerns).

Finally, we investigate the separability between two types of clients: (i) honest and (ii) malicious, based on the Marginal Likelihood quantifier. Figure 2.4(a)-(b) show the histogram of the data presented in Figure 2.4. We observe that for the supervised learning model, the distribution of the Marginal Likelihood quantifier has low separability between client types (classes present near values). On the other hand, considering the case of the SSL model, Figure 2.4(b) shows better separability between client types.

2.3.2 DDoS detection based Federated Learning

In this section, we present the results of our FL experiment focused on assessing the vulnerability of three datasets to label-poisoning attacks. We present an overview of the comparative approaches that serve as reference model for our research in FL. In Section 2.1.2, we introduce our model architecture. In our experiments, we consider that malicious user approaches can not modify the model training (only label attack), and the server can not collect data to evaluate the local models in the aggregation step, as seen in Li et al. [2021d].

Specifically, we evaluated the following proposals to demonstrate the effectiveness of our proposed method:

- Median [Yin et al., 2018] (**Median**) is a method that uses the median of local models to obtain a more accurate global model. This robust aggregation is a common technique in FL due to its effectiveness and simplicity of implementation.
- Krum-median [Blanchard et al., 2017] (**Krum**) is a robust aggregation rule that uses a Euclidean distance approach to select similar model updates. This method calculates the sum of squared distances between local model updates and then selects the model update with the lowest sum to update the parameters of the global model.
- Geometric median [Pillutla et al., 2022] (**Geometric**) is the point minimizing the sum of distances to the input points. Thus, the geometric median is a robust estimator in the presence of outliers and contamination.
- Trimmed mean [Fang et al., 2020] (**Trimmed**) removes the points above and below a given region (extreme values) and then calculates the mean of the remaining data. Due to this data removal, the trimmed mean helps eliminate the outliers that could unfairly affect the arithmetic average.
- FedEqual [Chen et al., 2021] (**FedEqual**) equalizes the weights of local model updates in FL, allowing most benign models to counterbalance malicious attackers' power and avoid excluding local models.

In our study, we conducted a comprehensive analysis of FL in a poisoning attack environment. For quantitative assessment, we compared the performance of each proposal against the Oracle method, which involves training a regular neural network with the softmax cross-entropy loss function. The Oracle method benefits from access to all non-corrupt data and follows a centralized approach without poisoning samples.

Table 2.1: Comparative evaluation for different ratios of noisy clients (ρ) considering all data training labeled. F1 scores are calculated for the attacked model and the Oracle model. We calculate degradation error as one minus the ratio of the F1-Scores. Higher values indicate higher model degradation compared to the Oracle model.

Dataset	Proposal	$d_{\text{error}} = 1 - F1/F1_{\text{Oracle}}$ (↓)				
		$\rho = 0$	$\rho = 0.2$	$\rho = 0.4$	$\rho = 0.6$	$\rho = 0.98$
CSE-CIC-IDS2018	Median	2.8%	13.7%	30.6%	42.4%	77.3%
	Krum	3.7%	8.1%	29.4%	58.3%	65.6%
	Geometric	3.8%	7.9%	38.8%	49.0%	70.9%
	Trimmed	3.1%	9.8%	48.1%	57.0%	68.6%
	FedEqual	**2.2%**	6.2%	20.9%	40.9%	73.7%
	Supervised learning	2.3%	5.9%	14.8%	31.7%	46.9%
	Our proposal	2.5%	**3.1%**	**7.8%**	**10.1%**	**15.3%**
	Oracle (F1-Score)	0.991	0.989	0.973	0.970	0.964
NF-UNSW-NB15-V2	Median	2.8%	10.3%	19.7%	47.7%	64.9%
	Krum	2.3%	6.1%	13.2%	36.6%	61.5%
	Geometric	1.9%	4.4%	11.3%	32.0%	59.9%
	Trimmed	2.1%	3.6%	25.6%	41.0%	62.8%
	FedEqual	3.3%	3.8%	14.8%	43.1%	69.2%
	Supervised learning	**0.9%**	3.0%	9.4%	28.9%	39.3%
	Our proposal	1.4%	**1.7%**	**5.8%**	**14.5%**	**22.7%**
	Oracle (F1-Score)	0.937	0.926	0.911	0.902	0.889
NF-ToN-IoT-V2	Median	3.8%	6.9%	23.4%	58.7%	70.8%
	Krum	5.0%	7.4%	18.1%	50.4%	71.9%
	Geometric	3.9%	12.2%	23.4%	47.1%	68.7%
	Trimmed	3.0%	5.6%	26.5%	46.5%	65.5%
	FedEqual	4.6%	6.3%	16.1%	38.6%	72.7%
	Supervised learning	**0.8%**	3.1%	6.6%	28.8%	40.3%
	Our proposal	1.1%	**2.5%**	**3.2%**	**12.0%**	**20.6%**
	Oracle (F1-Score)	0.968	0.954	0.945	0.935	0.930

To evaluate the impact of a poisoning model attack, we utilized the F1-Score of the attacked model, comparing it with the F1-Score of the Oracle model. From this comparison, we calculated the degradation error, denoted as $d_{\text{error}} = 1 - F1/F1_{\text{Oracle}}$, where $F1$ represents the F1-Score of the attacked model, and $F1_{\text{Oracle}}$ represents the F1-Score of the Oracle model. This metric estimates the model degradation (higher means more degraded). Additionally, we conducted an ablation study by contrasting the performance of the federated Marginal Likelihood aggregation (described in Equation. (2.3)) with a traditional supervised learning model and our proposed method that employs Semi-Supervised Learning (SSL). Table 2.1 shows a comparative evaluation between our method and some FL aggregation (between local model's weights) methods found in the literature.

In the absence of malicious clients ($\rho = 0$), our proposed approach demonstrates competitive performance compared to the analyzed methodologies (**RQ 2.2**). Specifically, we observe a marginal difference of 1.7% between the best and worst performing techniques

for the **CSE-CIC-IDS2018** dataset.

Furthermore, we noticed that, for this dataset, FedEqual exhibits superior performance compared to all other techniques. Conversely, the Marginal Likelihood aggregation using a supervised learning model and our approach achieve the second and third-best performance, respectively (Oracle achieved 0.991 in F1-Score). This observation can be attributed to the fact that the three mentioned techniques retain all models during aggregation, unlike conventional methods in homogeneous environments that may exclude outlier models based on suspicion of malicious influence. In contrast, the privacy-preserving nature of FL introduces a heterogeneous environment where end devices possess varying data classes and quantities, thereby complicating the distinction between malicious and benign outliers.

Turning to the other two datasets, our proposal achieves the second-best result, while the Supervised learning approach attains the highest performance. This result shows the viability of employing marginal likelihood as an aggregation mechanism for local models, even in scenarios without malicious entities. Furthermore, our approach demonstrates commendable competitiveness in such benign contexts.

In our FL experiment, we investigate the impact of malicious clients on various aggregation methods. As the fraction of malicious clients increases, we observe a significant degradation in results for all methods except our proposed method (**RQ 2.3**). For 40% of corrupted clients ($\rho = 0.4$), all baselines have a considerable degradation, resulting in a degradation error value of more than 14%, 9% and 6% for **CSE-CIC-IDS2018**, **NF-UNSW-NB15-V2** and **NF-ToN-IoT-V2**. In comparison, our proposed method maintains satisfactory performance, achieving an d_{error} of 7.8%, 5.8%, and 3.2% while still being able to complete the task. This drop in performance can be attributed to the prevalence of malicious users, which is not typically accounted for in existing outlier mitigation approaches. Notably, methods like **Krum**, which group similar models and remove outliers, exhibit considerable degradation for $\rho > 0.2$ due to the increased prevalence of malicious users in the FL environment.

Finally, for the case where we have 98% of corrupted clients ($\rho = 0.98$), all baselines have steeply degraded results, indicating that the aggregate model leads to a low F1-Score for the analyzed data. In addition, it is essential to note that we have only one honest client in this configuration of $\rho = 0.98$, and the Oracle method achieved 0.964, 0.889, and 0.930 in F1-Score for the three datasets. Moreover, our proposal is the only one that kept the ability to perform the task. For example, for **CSE-CIC-IDS2018** dataset, we observed an improvement in the result in 67.37% and 76.68% compared with the second and third best results, respectively, in this setting.

Table 2.2: Comparative evaluation for different ratios of noisy clients (ρ) considering 5% of data training labeled. F1 scores are calculated for the attacked model and the Oracle model. We calculate degradation error as one minus the ratio of the F1-Scores. Higher values indicate higher model degradation compared to the Oracle model.

Dataset	Proposal	$d_{\text{error}} = 1 - F1/F1_{\text{Oracle}}$ (↓)				
		$\rho = 0$	$\rho = 0.2$	$\rho = 0.4$	$\rho = 0.6$	$\rho = 0.98$
CSE-CIC-IDS2018	Median	6.7%	16.7%	36.6%	48.3%	66.3%
	Krum	8.9%	16.0%	33.2%	50.1%	72.2%
	Geometric	7.8%	10.8%	30.3%	46.9%	68.9%
	Trimmed	9.0%	11.4%	35.8%	57.7%	70.6%
	FedEqual	6.3%	10.0%	27.1%	52.5%	69.2%
	Supervised learning	**5.5%**	9.9%	19.8%	28.1%	49.4%
	Our proposal	6.5%	**8.9%**	**9.7%**	**10.3%**	**12.9%**
	Oracle (F1-Score)	0.982	0.967	0.955	0.941	0.927
NF-UNSW-NB15-V2	Median	6.2%	14.3%	21.3%	49.2%	60.1%
	Krum	7.1%	15.4%	25.7%	42.3%	58.2%
	Geometric	6.7%	10.6%	20.0%	36.4%	48.7%
	Trimmed	5.9%	12.1%	28.1%	59.2%	63.8%
	FedEqual	6.9%	9.4%	19.7%	31.6%	56.4%
	Supervised learning	**5.0%**	7.5%	17.8%	24.2%	34.8%
	Our proposal	5.7%	**6.9%**	**10.2%**	**16.3%**	**23.6%**
	Oracle (F1-Score)	0.921	0.910	0.904	0.891	0.877
NF-ToN-IoT-V2	Median	4.1%	9.9%	29.4%	58.8%	68.7%
	Krum	**1.8%**	8.5%	19.9%	40.3%	59.4%
	Geometric	4.2%	11.0%	30.4%	41.1%	62.8%
	Trimmed	5.4%	9.9%	29.7%	49.3%	67.1%
	FedEqual	4.8%	9.0%	19.5%	41.2%	64.8%
	Supervised learning	2.1%	5.6%	10.4%	26.3%	41.1%
	Our proposal	4.0%	**4.6%**	**8.5%**	**21.9%**	**29.2%**
	Oracle (F1-Score)	0.957	0.944	0.930	0.921	0.912

2.3.3 Realistic DDoS detection based Federated Learning

To perform a more realistic experiment, we assume that the FOG network administrator cannot label all training data in the system. Data collection and annotation are usually expensive and tedious. Consequently, we seldom have a large volume of labeled data to train supervised approaches. In this new experiment, we consider the more common case where the IDS network administrator has only enough resources to label part of the data but has access to a large volume of unlabeled data. Therefore, the IDS network administrator can label only 5% of the training data (**RQ 2.4**). Furthermore, the administrator possesses a substantial amount of unlabeled data.

In our experiment, we randomly sampled 5% from each client training dataset to use as labeled data, and the remaining training data was considered unlabeled data to

train the self-supervised approach.

The results, as presented in Table 2.2, showcase the effectiveness of the different FL algorithms in enhancing the performance of DDoS detection systems. Furthermore, the Oracle model's F1 score provides a reference point for evaluating the algorithms' effectiveness.

For **CSE-CIC-IDS2018** dataset with $\rho = 0$, we observed that the supervised learning (Marginal Likelihood aggregation) approach achieved the best result (slightest degradation error) for the dataset analyzed, thus surpassing all other analyzed algorithms. For reference, the oracle achieved an F1-Score of 0.982.

However, in the presence of malicious clients, we see that the approaches' performance is degraded compared to our proposal. For the scenario with 20% of malicious clients ($\rho = 0.2$), our approach outperforms all other methods used in this experiment, displaying only 8.9% for degradation error (for reference, oracle's F1-Score was 0.967). This advantage is even more marked when we analyze more extreme scenarios with a higher proportion of malicious users. Finally, for the scenario where $\rho = 0.98$ we observe that our approach gets 12.9% for degradation error (oracle achieved 0.927 in F1-Score), resulting in a difference of 73.88% for **Supervised learning** using Marginal Likelihood aggregation (second-best result).

Analyzing the Table 2.2, **NF-UNSW-NB15-V2** dataset considering only 5% label data training for the conventional scenario ($\rho = 0$), we found that our proposal had the second-best performance compared to the literature proposals, thus showing that our approach is effective for DDoS detection even without malicious users.

However, we observe that the degradation error is greater than the experiment considering all data labeled (reported in Table 2.1). This effect is evident for values of $\rho \leq 0.4$. Probably for $\rho = 0$, this effect occurs because, with less labeled data, the neural network tends to have more difficulty classifying the data correctly. Besides, as the amount of malicious users grows (e.g., $\rho = 0.2, 0.4$) in the federation, the lack of information on unlabeled data, as well as attacks on the model affects the performance of the model, degrading its result significantly.

Even with this degradation related to the lack of labels, we observed that our proposal manages to overcome all other approaches in a scenario where we consider malicious users ($\rho > 0$). In summary, our FL approach demonstrates resilience to the presence of malicious clients, maintaining stable performance even as their proportion increases. For example, we observe that for $\rho = 0.4$, our proposal reduces degradation error of 42.69% and 48.22% when compared to the second (Supervised learning) and third (FedEqual) best approach for **NF-UNSW-NB15-V2** dataset.

Finally, in the experiment using the **NF-ToN-IoT-V2** dataset, we verified that for the case without malicious users ($\rho = 0$), our proposal has the third best result, being surpassed by the Krum approach (best result) and Supervised learning (second-

best result). However, our proposal outperforms all analyzed approaches in the scenario with malicious users.

Specifically, our approach achieves a degradation error of 4.6%, 8.5%, 21.9%, and 29.2% for the respective proportions of malicious clients. In comparison, other aggregation techniques exhibit higher degradation errors, with **Median**, **Krum**, **Geometric**, **Trimmed**, and **FedEqual** ranging from 9.9% to 68.7%, 8.5% to 59.4%, 11.0% to 62.8%, 9.9% to 64.8%, and 5.6% to 41.1%, respectively. Furthermore, the supervised learning proposal shows higher degradation errors than our method, ranging from 5.6% to 41.1%.

The experimental results highlight the effectiveness of our proposed method in handling label-poisoning attacks, consistently outperforming other aggregation techniques.

2.4 Related Work

The security issues in machine learning systems, and consequently FL, have been extensively studied [Hao et al., 2020, Wang et al., 2020, Su and Qu, 2022]. This section provides an overview of the existing literature on (supervised and unsupervised) DDoS detection techniques in FL environments.

Li et al. [2021c] propose a novel federated intrusion detection system for DDoS attacks. First, the authors propose using the prototypical vector to quantify the correlation between the representation spaces estimated locally (through neural networks) and the global model. Finally, the authors use the average weighted by the inverse of the euclidian distance between the local and global prototype vectors to aggregate local models. The authors use the CICDDoS2019 dataset to evaluate the proposal, obtaining 97% accuracy for five federation clients.

Tian et al. [2021] propose a lightweight residual neural network for the DDoS classification task and achieved 99.20% accuracy for the CICDDoS2019 dataset. Similarly, Lv et al. [2022] propose a convolution neural network to counter DDoS attacks in FL environments. The accuracy of DDoS attack detection (CICDDoS-2019 dataset) and multiclass classification is 99% and 90%, respectively. Finally, Dimolianis et al. [2022] use Multi-layer Perceptrons to detect malicious packet signatures in DDoS attack scenarios.

Zhang et al. [2021d] analyze the problem of DDoS detection, considering the scenario of non-iid data in a federated environment. Thus, the authors propose FLDDoS, which uses the hierarchical aggregation algorithm based on K-Means and a resampling data method based on SMOTEENN. Considering these methods, the authors obtained 92.62% and 99.62% for the F1-Score metric for the CICIDS and NLSKDD datasets, respectively. Chen et al. [2022] propose an approach based on majority resampling (random

under-sampling) and minority class synthetic sample generation (SMOTE) to estimate the contribution of each client, followed by the aggregation of the federated models. In addition, Ali et al. [2023] propose a new weighted FL model based on an artificial neural network to detect DDoS attacks in SDN Control Plane in IoT Network with non-iid client data.

In a multi-task scenario, Zhao et al. [2019] propose a FL approach using a multi-task deep neural network for network anomaly detection, VPN (Tor) traffic recognition, and traffic classification simultaneously. Experimental results show that the proposed method outperforms the baseline methods concerning detection and classification performance, making it a promising solution for data privacy and scarcity concerns in network security.

Image-based approaches are gaining popularity among the new set of systems vulnerability detection due to their ease of use and infrastructure for synthetic images [Barros et al., 2022a]. Thus, Toldinas et al. [2022] transform Network flow feature records into images and classify these images in an FL scenario. The F1 score values of Global Models testing from 93.78% to 96.86% were obtained in two experiments with the BOUN DDoS dataset.

Architectures based on fog computing can obstruct the malicious traffic generated by the DDoS attack from the user to the server. In this scenario, Fog functions as a filtering layer for the generated traffic, thus increasing the federation's security. Li et al. [2022a] revisit DDoS attacks and propose an iterative model averaging protocol to make the attack more expensive than possible profits. Thus, the authors combine FL and fog computing for the DDoS classification problems, using Iterative Model Averaging (IMA)-based gated recurrent unit.

Dao et al. [2022] propose FOGshield to detect and prevent DDoS attacks in fog-based heterogeneous IoT systems. A cloud-based orchestrator (i.e., FL model) and multiple fogging endpoint defenders to improve the attack detection performance. Finally, using self-organizing maps, the authors exploit their local traffic to filter abnormal flows in fog. Similarly, Neto et al. [2022] propose a multi-fog IoT environment using FL. The experiments performed in a simulated environment achieved 84.2% accuracy on CICDDoS2019.

Although unsupervised techniques show promising results [Dao et al., 2022, Li et al., 2022a], using supervised learning algorithms can significantly improve the performance of model detection [Van Engelen and Hoos, 2020], as these models are trained to recognize specific patterns in labeled data and can more efficiently learn to distinguish legitimate from malicious traffic.

Detecting DDoS attacks is challenging due to the variety of attack vectors and their high variability. Therefore, DDoS detection models must be highly accurate and reliable. Machine learning techniques that can be trained with reduced labeled data samples are

Table 2.3: Summary of approaches shown in this related work section

Year	Paper	Supervised Learning	Malicious Client	Small LB available	Robust label poisoning
2019	Zhao et al. [2019]	✓	✗	✗	✗
2021	Li et al. [2021c]	✓	✗	✗	✗
	Lv et al. [2022]	✓	✗	✗	✗
	Tian et al. [2021]	✓	✗	✗	✗
	Zhang et al. [2021d]	✓	✗	✗	✗
2022	Chen et al. [2022]	✓	✗	✗	✗
	Dao et al. [2022]	✗	✗	✗	✓
	Dimolianis et al. [2022]	✓	✗	✗	✗
	Li et al. [2022a]	✗	✗	✗	✓
	Neto et al. [2022]	✓	✗	✗	✗
	Su and Qu [2022]	✓	✗	✗	✗
	Yin et al. [2022]	✓	✓	✗	✓
2023	Ali et al. [2023]	✓	✗	✗	✗
	Liu et al. [2023b]	✓	✗	✓	✗
	Our approach	✓	✓	✓	✓

crucial to enable effective DDoS detection even with limited datasets. These techniques reduce the time and cost of manual labeling and allow models to be trained with data distributed across multiple devices without compromising data privacy.

Liu et al. [2023b] propose a bidirectional LSTM model for DDoS attack detection in this context. To verify the effectiveness of the FL framework, authors evaluate clients who only have access to a small portion (5%) of the total dataset and compare the proposed performance with traditional centralized training methods.

Finally, Yin et al. [2022] propose a trusted multi-domain DDoS detection method based on FL, where a reputation evaluation method based on blockchain is the main novelty. The proposed method divides the types of DDoS attacks into different sub-attacks and designs FL datasets to protect the data privacy of each domain. The experimental results show that the accuracy of most categories of the multi-domain DDoS detection method can reach more than 95%.

2.4.1 Discussion

Most of the methods cited indicate the feasibility of identifying DDoS packages in FL environments. Table 2.3 summarizes FL approaches related to the discussed topic in the chapter. The table is organized into columns describing different key aspects of the approaches: if the proposal was trained in a supervised learning setting, if the federation had malicious clients, if the model was trained with small data samples, and if

the proposal is robust against label poisoning attacks. Cells are marked with a "✓" or a "✗" accordingly.

A supervised approach in FL models improved accuracy and reduced data required to train the model. However, it also introduces new security concerns that need to be addressed, e.g., label-poisoning attacks can compromise the model's integrity and lead to incorrect predictions [Barros and Ramos, 2022].

In this way, we observe that some approaches in the literature propose supervised models for DDoS detection [Zhao et al., 2019, Zhang et al., 2021d, Li et al., 2021c, Tian et al., 2021, Su and Qu, 2022, Chen et al., 2022, Dimolianis et al., 2022, Neto et al., 2022, Lv et al., 2022, Ali et al., 2023, Liu et al., 2023b], but do not take into account this new attack surface generated by supervised models. To mitigate this attack surface, it is essential to implement appropriate security measures, such as data validation and encryption, to detect and prevent attacks from malicious clients [Nguyen et al., 2021]. By doing so, we can ensure our models' integrity and maintain their predictions' accuracy.

Although blockchain can help select trustworthy participants in a federated environment and prevent label poisoning attacks [Yin et al., 2022], some limitations and challenges must be considered. First, the use of blockchain can increase the complexity and cost of the system since the technology is not yet widely adopted and requires additional resources for implementation and maintenance [Issa et al., 2023].

On the other hand, collecting a large amount of labeled data is challenging due to time constraints and financial or hardware resources. In these cases, using a small sample of labeled data may be a viable solution for model training. These approaches can enable the model to be trained more efficiently, resulting in better DDoS detection performance. However, in this scenario, we observe in the literature that only the work of Liu et al. [2023b] deals with a low amount of labeled data. However, they do not tackle model attacks in the federation.

Thus, our method considers the federation's susceptibility to malicious clients that aim to poison the federated model. In this scenario, FL models tend to have their performance degraded without security consideration since most models discussed in this section assume that the federation clients are honest. In addition to considering the network attack (system attack), we propose an approach that also considers the presence of malicious clients that tend to affect federation training (model attack). Finally, as detailed in Section 2.1.2, we propose an architecture design for robust data labeling.

2.5 Final Remarks

This chapter proposes a novel distributed framework to detect network attacks. Our proposal presents evidence that, in addition to detecting DDoS attacks, it is robust to model attacks, thus presenting two levels of security. We have specifically designed our approach to resist label-poisoning attacks. We plan to create countermeasures to resist other model attacks for future work. Our framework achieves better performance when compared to state-of-the-art techniques. This conclusion is evidenced mainly in scenarios with a large fraction of malicious users ($\rho > 0.2$). Finally, it is worth emphasizing that the SSL model freezes the encoder weights, leading to a smaller capacity model. As seen in this chapter, quantifiers that consider the trade-off between the model's capacity and generalization error prove to be relevant for FL model training. Therefore, investigating how to quantify Occam's Razor effect in a distributed heterogeneous environment (e.g., FL) is an exciting open research direction.

Chapter 3

Ad-Hoc uncertainty quantification via metric learning to mitigate malicious clients replacement in federated learning

This chapter, which was published at the IEEE Global Communications Conference [Barros and Ramos, 2022], employs an ad-hoc approach that does not utilize Bayesian neural networks. The methodologies and findings from this study are further extended in next Chapters.

In this chapter, we propose a new similarity function for FL applications to tackle the model poisoning vulnerability. Our method uses a new security aggregation proposal based on the quantification of the heterogeneity of the data. We adopt our previous model as a backbone for our FL approach. We presented some theoretical results for this model in Barros et al. [2022b] and leverage these properties to design the data heterogeneity quantifier herein proposed. We evaluate our proposal in a FL scenario without any attack, where we achieve a performance of 87.68% and 80.74% on F1-Score. These results outperformed the vanilla neural network model without our auxiliary space by 52.84% and 58.88% in a classification model on a real-world dataset. Finally, in the experiment considering model poisoning, our approach reached an F1-score of 81.79% and 73.92% for the two real datasets analyzed in the FL experiment, outperforming other methods by 8.60% and 5.54%, respectively.

3.1 Our proposal

3.1.1 Data representation

*This **data representation** section builds upon partial results from my master's dissertation, which explored related concepts in a centralized environment. The methodologies and findings discussed here are extensions of that initial work, adapted and further developed for the FL context. For more details on Deep Metric Learning and our previous proposal, SMELL, see the Appendix A*

We hypothesized that the *latent feature space* could be improved the feature representation with an auxiliary space. Unlike the literature, we used our previous approach (Appendix A) to quantify similarity using an auxiliary space called *S-Space* [Barros et al., 2022a], as we can see in Figure 3.1 and we will detail it now.

We defined a function $f^S : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{S}$ that maps a pair of elements from the *original feature space* into a new auxiliary space called *similarity space* (or *S-Space*), where if $\boldsymbol{x}_i$ has the same label as $\boldsymbol{x}_j$, then $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a similar pair. On the other hand, we consider $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to be a dissimilar pair if $\boldsymbol{x}_i$ has a different label than $\boldsymbol{x}_j$.

We define the mapping function f^S to a pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ by the following pairwise operation

$$\begin{aligned} \boldsymbol{s}_{ij} &= f^S(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &= |f_\Theta(\boldsymbol{x}_i) - f_\Theta(\boldsymbol{x}_j)| \\ &= |\boldsymbol{z}_i - \boldsymbol{z}_j| \\ &= (|z_i^1 - z_j^1|, |z_i^2 - z_j^2|, \cdots, |z_i^d - z_j^d|), \end{aligned} \tag{3.1}$$

where $\boldsymbol{s}_{ij}$ has the same dimension as $\boldsymbol{z}_i$ and z_i^n is the n-th *feature* of the i-th sample in a *latent space representation* $\mathcal{Z}$. Finally, we use the absolute operation to preserve symmetry, i.e., $\boldsymbol{s}_{ij} = \boldsymbol{s}_{ji}$.

In *S-Space*, we have some markers to calculate the similarity between input pairs. The set of similarity markers is defined as $\mathcal{M}^+$, and similarly, markers in the set $\mathcal{M}^-$ quantify dissimilarity. So the set of all markers is $\mathcal{M} = \mathcal{M}^+ \cup \mathcal{M}^-$, where $\mathcal{M}^+ \cap \mathcal{M}^- = \emptyset$.

Therefore, for $\boldsymbol{s}_{ij}$ associated to the pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the closer the vector $\boldsymbol{s}_{ij}$ to a marker $\boldsymbol{m}^+ \in \mathcal{M}^+$, the greater the similarity between the pairs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Analogously, the distance to $\boldsymbol{m}^- \in \mathcal{M}^-$ measures the dissimilarity. We can use the d function to measure the similarity between $\boldsymbol{s}_{ij}$ and the marker $\boldsymbol{m}_w \in \mathcal{M}$, as $q_{ij}^w = d(\boldsymbol{m}_w, \boldsymbol{s}_{ij})$, where q_{ij}^w is the similarity/dissimilarity of $\boldsymbol{s}_{ij}$ and the marker m_w. We define $q_{ij}^+ = \sum_p q_{ij}^p$ for all $\boldsymbol{m}_p \in \mathcal{M}^+$ and $q_{ij}^- = \sum_n q_{ij}^n$ for all $\boldsymbol{m}_n \in \mathcal{M}^-$. In this proposal, following [Maaten and Hinton, 2008], we use Cauchy kernel as

$$d(\boldsymbol{m}_w, \boldsymbol{s}_{ij}) := q_{ij}^w = \frac{(1 + \|\boldsymbol{s}_{ij} - \boldsymbol{m}_w\|_2^2)^{-1}}{\sum_{m_{m'} \in \mathcal{M}} (1 + \|\boldsymbol{s}_{ij} - \boldsymbol{m}_{m'}\|_2^2)^{-1}}. \tag{3.2}$$

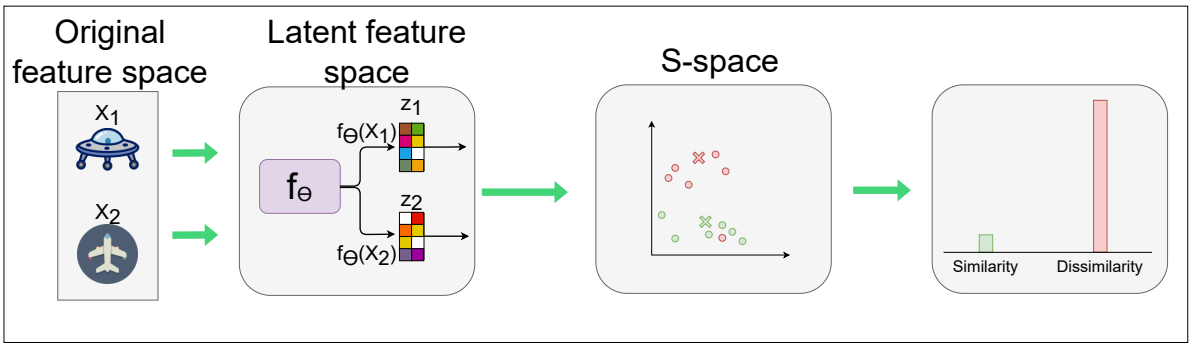


Figure 3.1: Schematic of similarity extraction tailored S-space.

By definition, we consider that q_{ij}^{+} is a probability that the pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ has the same label and q_{ij}^{-} is a probability that the pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ will be dissimilar (different label). Also, by construction, there are no common elements in the sets $\mathcal{M}^{+}$ and $\mathcal{M}^{-}$, i.e., we have $q_{ij}^{+} + q_{ij}^{-} = 1$.

Let $\mathcal{Q} = \{q_{ij}\}$, the model output, be the set that contains the pairs $q_{ij} = (q_{ij}^{+}, q_{ij}^{-})$ corresponding to the probability of the elements of a pairwise input $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ be similar or dissimilar, respectively. The loss function j can be defined as

$$J(\mathcal{X}) = \sum_{i \in \mathcal{X}} \sum_{j \in \mathcal{X}} H_c(\boldsymbol{u}_{ij}, \boldsymbol{q}_{ij}), \tag{3.3}$$

where H_c is the binary cross-entropy loss and $\boldsymbol{u}_{ij} \in \mathcal{U}$ is defined as $\boldsymbol{u}_{ij} = (1, 0)$ if i has same label as j and $\boldsymbol{u}_{ij} = (0, 1)$, otherwise[1].

So, our proposal adds the time complexity $O(dw)$, where d is the latent feature space dimension (typically $d < 1024$, and w is the number of markers (typically $w < 8$). Furthermore, $O(dw) \ll O(N)$, where $O(N)$ is the complexity of the neural network training (encoder function).

3.1.2 Aggregation

The S-space as firstly proposed in our previous work [Barros et al., 2022b] (see the Appendix A). Due to the construction of the S-space, we can obtain some theoretical proprieties.

Definition 3.1. *(Optimal Latent Space) Let $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ be a pair in original representation space and a latent representation function $f_\Theta : \mathcal{X} \to \mathcal{Z}$. The transformation f_Θ generates an optimal latent space $\mathcal{Z}$ when the expected value user Neural network f_Θ is*

[1]Marker positions and neural-network weights are jointly optimized using stochastic gradient descent (SGD) via backpropagation (c.f. Appendix A.3.3).

$\mathbb{E}_{i,j}[\|\boldsymbol{s}_{ij}\|_2] = 0 \implies \ell(\boldsymbol{x}_i) = \ell(\boldsymbol{x}_j)$, *where function* $\ell : \mathcal{X} \to \mathcal{Y}$ *maps an unlabeled example* $\boldsymbol{x}_i$ *into their respective label* y_i.

Our proposal can group points of the same class into clusters. It is worth noting that we defined the optimal space as a conditional instead of a biconditional statement. From this definition, we can observe that our model may create several different clusters of the same class (see more details in Section 5.4).

Proposition 3.1. *In S-space, given* k *positive markers in the set* $\mathcal{M}^+$, $n-k$ *negative markers in* $\mathcal{M}^-$ *and* $k = n-k$, *the latent space found by our approach, i.e., the estimation of the parameters* Θ *of* f_Θ, *generates an optimal latent space if* $\exists\ \boldsymbol{m}_i \in \mathcal{M}^+$ *so that* $\|\boldsymbol{m}_i\|_2^2 < \|\boldsymbol{m}_j\|_2^2$ *for any* $\boldsymbol{m}_j \in \mathcal{M}^-$.

Proof. Given $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, our proposal measures the similarity between the entries through the Cauchy kernel given by q_{ij}^+, so that for optimal weights $\boldsymbol{\Theta}^*$ and $\mathcal{M}^*$ it follows that $q_{ij}^+ \to 1 \iff \ell(\boldsymbol{x}_i) = \ell(\boldsymbol{x}_j)$.

Since f_Θ generates an optimal space, we then have $\mathbb{E}[\|\boldsymbol{s}_{ij}\|_2] = 0 \implies \ell(\boldsymbol{x}_i) = \ell(\boldsymbol{x}_j)$, so, it follows that for a optimal latent space, we must have (Eq. 3.2)

$$\begin{aligned} q_{ij}^+ &= \frac{\sum_{k\in\mathcal{M}^+}(1+\|\boldsymbol{m}_k\|_2^2)^{-1}}{\sum_{k\in\mathcal{M}^+}(1+\|\boldsymbol{m}_k\|_2^2)^{-1} + \sum_{s\in\mathcal{M}^-}(1+\|\boldsymbol{m}_s\|_2^2)^{-1}} \\ &= \frac{1}{1+\frac{\sum_{s\in\mathcal{M}^-}(1+\|\boldsymbol{m}_s\|_2^2)^{-1}}{\sum_{k\in\mathcal{M}^+}(1+\|\boldsymbol{m}_k\|_2^2)^{-1}}}. \end{aligned}$$

Hence, if we want $\ell(\boldsymbol{x}_i) = \ell(\boldsymbol{x}_j)$, we should ideally have q_{ij}^+ tends to 1. It follows that $\sum_{s\in\mathcal{M}^-}(1+\|\boldsymbol{m}_s\|_2^2)^{-1} < \sum_{k\in\mathcal{M}^+}(1+\|\boldsymbol{m}_k\|_2^2)^{-1}$. Therefore, let m^+ be the element with the smallest module in the set $\mathcal{M}^+$; we then have $\sum_{k\in\mathcal{M}^+}(1+\|\boldsymbol{m}_k\|_2^2)^{-1} < k(1+\|\boldsymbol{m}^+\|_2^2)^{-1}$. Analogously, we can consider $\boldsymbol{m}^-$ as the vector with the largest module in the set $\mathcal{M}^-$, so, $\sum_{k\in\mathcal{M}^-}(1+\|\boldsymbol{m}_k\|_2^2)^{-1} > (n-k)(1+\|\boldsymbol{m}^-\|_2^2)^{-1}$.

We can then conclude that $k(1+\|\boldsymbol{m}^+\|_2^2)^{-1} > (n-k)(1+\|\boldsymbol{m}^-\|_2^2)^{-1}$, and therefore, $(n-k)(1+\|\boldsymbol{m}^+\|_2^2) < k(1+\|\boldsymbol{m}^-\|_2^2)$. Furthermore, adding the restriction that our propose has the same count of positive and negative markers ($k = n-k$), we have $1+\|\boldsymbol{m}^+\|_2^2 < 1+\|\boldsymbol{m}^-\|_2^2$. □

Thus, by the **Proposition** 3.1, given the positive marker with the smallest module $\|\boldsymbol{m}^+\|_2$, we get information about the separability of the *latent feature space*. So, in this chapter, we propose a novel function to quantify the separability of *latent feature space* (called model inference ability) as

Definition 3.2. *We define the inference ability* K *of a model based on S-space* $\mathcal{S}$ *as being*

$$K(\mathcal{S}) = 1 - \frac{\|\boldsymbol{m}^+\|_2}{\|\boldsymbol{m}^-\|_2},$$

where $\|\boldsymbol{m}^+\|_2 \le \|\boldsymbol{m}\|_2$ *for all* $\boldsymbol{m} \in \mathcal{M}^+$ *and* $\|\boldsymbol{m}^-\|_2 \le \|\boldsymbol{m}\|_2$ *for all* $\boldsymbol{m} \in \mathcal{M}^-$.

Therefore, let $\mathcal{S}_i$ be the similarity space found by the FL user model $T_{\mathcal{X}_i}$ associated with the user u_i, we can calculate the inference capacity as being $K(\mathcal{S}_i)$. Given an FL application, we can build the aggregate model as$\Theta_{Fed} \leftarrow \Theta_{Fed} + \eta \sum_{i=1}^{N_{sel}} \left[j_i(\Theta^*_{\mathcal{X}_i} - \Theta_{Fed}) \right]$, where $j_i = K(\mathcal{S}_i) / \sum_{s=1}^{N_{sel}} K(\mathcal{S}_s)$.

3.2 Methodology

3.2.1 Attack Model

A Θ^*_{Mal} model is said to be malicious when it deliberately fulfills its intention to harm a machine learning system. The authors in [Bagdasaryan et al., 2020] define a malicious model that aims to replace global model Θ_{Fed} to introduce error (e.g., bad functionality) in the new model Θ_{Poi}. Thus, given a FL round with $\{T_i\}_{i=1}^{N_{sel}-1}$ local non-malicious model and a malicious model T_{Mal}, the aggregation can be defined in this attack as

$$\Theta_{Poi} \leftarrow \Theta_{Fed} + \eta \left[\sum_{i=1}^{N_{sel}-1} \left[p_i(\Theta^*_{\mathcal{X}_i} - \Theta_{Fed}) \right] + p_{Mal}(\Theta^*_{Mal} - \Theta_{Fed}) \right],$$

where Θ_{Poi} is the new global federation model. Typically, $p_i = 1/N_{sel}$ or $p_i = n_i / \sum_{i=1}^{N_{sel}} n_i$, with $n_i = |\mathcal{X}_i|$ as can see in Li et al. [2020a].

Thus following [Bagdasaryan et al., 2020, Chen et al., 2021], when Θ_{Fed} converges, we have that $\sum_{i=1}^{N_{sel}-1} \left[p_i(\Theta^*_{\mathcal{X}_i} - \Theta_{Fed}) \right] \approx 0$. Thus, a malicious client model Θ^*_{Mal} can replace the global model Θ_{Fed} with a poisoning model Θ_{Poi} as

$$\Theta^*_{Mal} = \Theta_{Fed} + \frac{1}{\eta p_{Mal}} (\Theta_{Poi} - \Theta_{Fed}).$$

In this chapter, we use $\eta = 0.01$, as can see in Bagdasaryan et al. [2020]. In addition, we defined ρ as the proportion of malicious clients in the federation.

3.2.2 Dataset

To analyze the FL performance of our proposal, we used two datasets:

– **Malimg Dataset** was developed by the University of California's vision research laboratory[2]. This dataset contains 9339 samples from 25 malware families, obtained through experiments of mixtures of network and the Windows operating system malware. The Malimg dataset has a diversity of examples. Specifically, the number of samples from the *malware* families ranges from 80 to 2949.

– **MaleVis Dataset** is a corpus involving byte images of 26 (25+1) classes. Here, 1 class represents the goodware samples while the rest of the 25 classes correspond to different malware types. The MaleVis dataset involves a total of 14226 images. For the MaleViz dataset, malware examples range from 470 to 500. The Maleviz dataset contains multiple classes with 25 malware families and 1832 goodware examples.

Each binary code of *malware* is a vector of 8-bit unsigned integers organized in a two-dimensional matrix resized to 64 x 64 and visualized as a gray-scale image.

3.2.3 Model evaluation

We use the framework Flower [Beutel et al., 2020b] to develop solutions and applications in the context of FL. We separate the dataset into training (users), validation (server), and test (server), with proportions of 80%, 10%, and 10%, respectively. We performed a non-iid data distributed among the users (training) used in this experiment, i.e., we randomly distributed the *malware* code samples in a non-uniform way (quantity-based label imbalance), as can be seen in McMahan et al. [2017]. Thus, we used a server and 40 clients to evaluate our model, and we trained our method with an NVIDIA Quadro RTX 6000 GPU (24 GB) for a total of 125 epochs (server). For each training round, the server selects five clients to train the local model, i.e., each model is trained using only the local data. Finally, the models are aggregated on the central server, which forwards the aggregated model to the clients.

We use the K-nearest neighbors (KNN) classifier to evaluate our approach with three neighbors, according to Wang et al. [2019a]. The performance of the KNN classification can sometimes be significantly improved through supervised similarity functions. We calculated the experiment's accuracy and F1-Score.

[2] https://vision.ece.ucsb.edu/research/signal-processingmalware-analysis

3.2.4 Network architecture

We initialize the position of markers with Lloyd's algorithm [Lloyd, 1982]. For the encoder, according to Barros et al. [2022a], we set network dimensions to m-500-500-2000-d for all datasets, where m is the number of features of the input data, and d is the *latent space representation* dimension. For this chapter, we use $d = 64$. All layers of neural networks are fully connected, and we use ReLU as the activation function. In addition, we use Stochastic Descending Gradient (SGD) with momentum, with a learning rate of 0.01 and a momentum of 0.9.

The optimization model depends on some hyper-parameters (n, k), so we investigated which value of these variables could maximize the model's accuracy. We used the grid search technique for hyper-parameter optimization and found $k = 3$ similarity markers and $n - k = 3$ dissimilarity markers. We used these values throughout the chapter.

3.3 Results and Discussion

Initially, for the Malimg dataset, we evaluated the performance of the K quantifier proposed in this chapter. We evaluated this quantifier during the client training, as shown in Figure 3.2.

This experiment shows a negative correlation between the value of *loss* and the model inference ability K. The *Loss* function quantifies a measure of dissimilarity between the model result and the label, i.e., the lower the value of *loss*, the better the model performance for the analyzed data. However, the *loss* depends on the validation dataset, so if a change is made in the evaluation dataset, the *loss* value is modified. Note that this behavior is not observed in our proposal, as the quantifier is defined on the model's markers, so this quantifier is agnostic to the validation dataset used by our proposal.

In Table 3.1 and 3.2, we performed an ablation study in our proposal and, in this evaluation, we compare our method with three approaches: (i) We use a NN with a federated average aggregation approach; (ii) We use a NN and *markers* approach with federated average approach and (iii) we use our proposal (section 3.1.2). The server never had access to user data, only local models. We use the models trained for 100 epochs, and malicious clients attack the models in the remaining 25 rounds.

Initially, for none malicious client ($\rho = 0$), we have that NN without markers (i) has

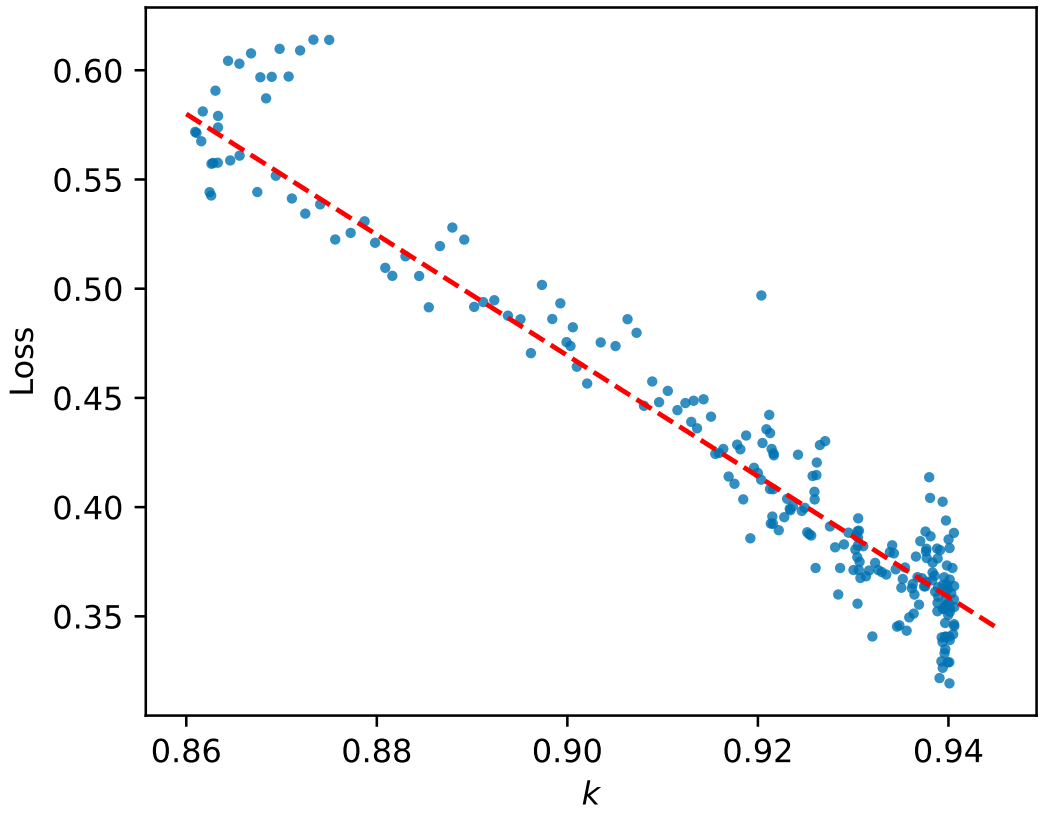


Figure 3.2: Loss versus inference ability K.

the worst performance for both datasets among all the analyzed approaches, with an F1-Score of 29.01% and 20.22%, surpassing by the two versions in ablation experiment. The NN and marker (ii) approach using FedAvg aggregation has the second-best performance for F1-Score with 84.79% for the Malimg dataset (Table 3.1) and 79.92% for the MaleViz dataset (Table 3.2). Finally, our aggregation proposal (iii) achieved the best F1-Score performance for Malimg and MaleViz datasets with 87.68% and 80.74%.

However, when we added the malicious clients, we noticed that the FedAvg aggregation algorithm had a degraded result. For $\rho = 0.125$ in the MaleViz dataset, we

Table 3.1: Results of Malimg dataset and best values are in bold.

Method	$\rho = 0$		$\rho = 0.125$		$\rho = 0.25$	
	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score
(i) NN w/o markers + FedAvg	40.09%	29.01%	–	–	–	–
NN w/ markers + Trimmed-mean [Fang et al., 2020]	89.83%	76.31%	80.99%	67.63%	79.98%	61.86%
NN w/ markers + FedEqual [Chen et al., 2021]	90.72%	83.27%	82.53%	74.26%	76.62%	73.19%
NN w/ markers + Median [Yin et al., 2018]	91.43%	83.61%	81.19%	71.51%	80.24%	65.38%
NN w/ markers + Krum-mean [Blanchard et al., 2017]	92.61%	82.39%	90.26%	76.83%	72.04%	67.49%
(ii) NN w/ markers + FedAvg	92.93%	84.79%	65.68%	57.99%	50.66%	42.07%
(iii) Our proposal	**93.15%**	**87.68%**	**92.56%**	**83.12%**	**89.53%**	**81.79%**
NN w/ marker (Centralized)	96.17%	90.94%	–	–	–	–

Table 3.2: Results of MaleViz dataset and best values are in bold.

Method	$\rho = 0$		$\rho = 0.125$		$\rho = 0.25$	
	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score
(i) NN w/o markers + FedAvg	21.62%	20.22%	–	–	–	–
NN w/ markers + Trimmed-mean [Fang et al., 2020]	79.71%	79.30%	70.65%	71.92%	67.06%	64.91%
NN w/ markers + FedEqual [Chen et al., 2021]	78.97%	77.90%	74.46%	73.42%	68.27%	68.38%
NN w/ markers + Median [Yin et al., 2018]	78.25%	76.94%	72.32%	72.06%	63.84%	62.21%
NN w/ markers + Krum-mean [Blanchard et al., 2017]	79.13%	78.45%	73.63%	73.24%	67.89%	67.75%
(ii) NN w/ markers + FedAvg	**80.50%**	79.92%	52.23%	51.72%	43.60%	38.71%
(iii) Our proposal	80.09%	**80.74%**	**79.95%**	**79.46%**	**74.78%**	**73.92%**
NN w/ marker (Centralized)	83.31%	82.79%	–	–	–	–

see that NN w/ marker approach (ii) gets a value of 52.23% and 51.72% for ACC and F1-Score metrics, respectively. Considering the same ρ value, our proposal gets 79.95% and 79.46% for the same metrics. Finally, for $\rho = 0.25$, we noticed that NN w/ marker (ii) approach obtains a 43.60% for the ACC metric, resulting in a difference equal to 31.18% when compared to our proposal (iii). This difference is more evident when we compare the F1-Score metric, where we observe a difference between the two approaches of 35.21%. A similar conclusion is observed in the Malimg dataset in Table 3.1.

In addition, we evaluated the performance of our proposal versus different model aggregation approaches found in the literature. We observed that our proposal performs best for the two metrics analyzed. For $\rho = 0.125$, NN w/ markers + Krum-mean aggregation obtained the second-best performance for the Malimg dataset, with ACC and F1-Score equal to 90.26% and 76.83%, respectively, in Malimg dataset. Our proposal achieves the best results compared to all other techniques, reaching 93.15% and 87.68%.

For $\rho = 0.25$, the second and third-best ACC results were obtained by NN w/ markers + Median and NN w/ markers + Trimmed-mean with 80.24% and 79.98%, a difference of 9.29% and 9.55% to our proposal, respectively. Similarly, for the F1-Score metric, our approach obtained 81.79%, the best overall result. The second and third best approaches obtained 73.19% and 67.49%, by NN w/ markers + FedEqual, and NN w/ markers + Krum-mean, respectively. A similar conclusion is observed in the MaleViz dataset in Table 3.2.

Finally, we compared our approach with the centralized proposal, i.e., we merged all local client data to train a single model and evaluated this model with the data server. This centralized approach performed very similarly to our proposal but did not consider users' privacy. Thus, our method effectively performs a classification task in a FL scenario, preserving clients' data privacy.

3.4 Related work

Several works propose an FL system to estimate the importance of each client's contribution without access to their local data [Mohri et al., 2019]. Typically, we obtained this estimate by evaluating the impact of each client on the performance of the aggregate FL model. The aggregation of local models can use clients' contribution amounts to distribute rewards and ensure fairness in FL.

Many defense methods found in the literature are based on identifying anomalous behavior arising from local models sent by customers. For the detection of anomalies, some works are based on robust aggregation functions to outliers such as median [Yin et al., 2018], mean with exclusion [Fang et al., 2020], geometric mean [Pillutla et al., 2022], and clustering of nearby gradients [Blanchard et al., 2017]. However, removing the outliers may prevent valuable information from being lost from the non-malicious local models. This type of exclusion in a client with heterogeneous data can generate overfitting in the global model, thus causing poor performance in a real-world situation [Chen et al., 2021].

Analyzing FL approaches that use metric learning techniques, Yu et al. [2020] proposed a generic framework for training clients that only know one class, called FedAwS. Based on the contrastive loss approach, the proposal builds a latent representation space that groups points of the same class into local groups, i.e., each client builds its own latent space. To build the aggregate model, FedAwS aggregates the local *latent spaces*, thus building a new representation with the global context of the data. Similarly, Park et al. [2021] proposes FedMetric, in which clients update local models with a loss function based on metric learning to minimize intraclass variance and simultaneously maximize interclass variance to prototype vectors generated by each client.

To the best of our knowledge, we have not found any work in the literature that proposes a metric learning technique for the FL (attack) application scenario. The works discussed in this section [Yu et al., 2020, Park et al., 2021] use the proposed loss functions for learnable metrics for classification problems, but they do not propose any learnable metrics. Therefore, to the best of our knowledge, this chapter proposes the first proposal of a learnable metric in an FL scenario.

3.5 Final Remarks

This chapter proposes a new similarity function based on deep metric learning for FL scenarios. Our proposal defines a quantifier to perform a security model aggregation tailored to S-space. Therefore, we obtained an interpretable quantifier about the data's heterogeneity through the markers' position in the similarity space. Furthermore, our proposal is agnostic to the validation dataset. We have conducted various experiments on the analyzed dataset, showing that our approach consistently outperforms other methods. Our model showed evidence of robustness to attacks, showing a gain of 35.21% against the standard neural network model and a difference of 8.60% for the F1-Score compared to aggregation proposals found in the literature. In future work, we intend to investigate the self-supervised learning scenario because these approaches tend to act as regularizers in neural networks, thus increasing the generalization of the proposal. This makes it perform well in zero-shot learning tasks and non-identically distributed problems, typically in FL applications.

Chapter 4

From Ad-Hoc quantification to reputation-based personalized federated learning under adversarial threats with a sedentary behavior Use Case

This chapter, which was published at [Barros et al., 2024a,b, 2025a], employs an ad-hoc approach to quantify uncertainty in FL. The methodologies and findings of this study are further extended in Chapter 5.

Detecting sedentary behavior is receiving increasing attention due to its significant health implications. However, distinguishing these low-intensity activities in federated learning scenarios is more complex than general human activity recognition. This complexity arises from heterogeneous feature distributions that can occur even for the same labeled activity, such as running and playing soccer, which may exhibit different sensor patterns despite both representing high-intensity activities [Zhu et al., 2023]. This chapter proposes a robust personalized FL approach for the classification of sedentary behavior under adversarial conditions. Our method leverages ordinal pattern descriptors [Cardoso-Pereira et al., 2022, Chagas et al., 2022] to extract meaningful symbolic representations from wearable sensor time series, then applies a meta-learning framework with Siamese Neural Networks to rapidly adapt across clients. Next, a reputation mechanism further protects the global model by penalizing malicious updates. Experiments on multiple public datasets show that our method achieves high F1-scores compared to baselines in the literature.

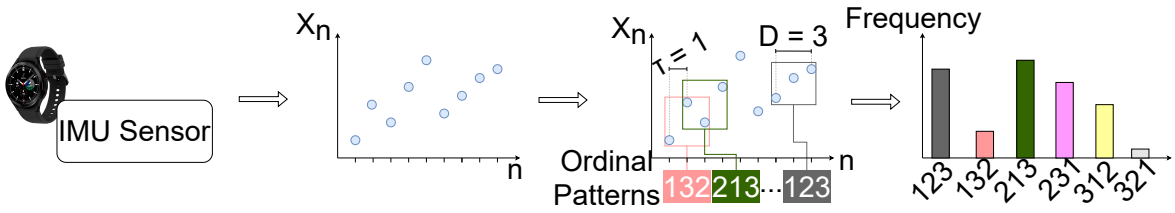


Figure 4.1: Illustrative workflow of ordinal pattern extraction for a TS measured by an IMU sensor.

4.1 Methodology

This chapter proposes a robust PFL approach for sedentary behavior classification in environments with malicious users. **First**, raw time-series data are transformed into a symbolic representation using Ordinal Patterns (OP) to capture temporal dynamics (as shown in Figure 4.1 and 4.2) and describe activity-driven changes in TS [Chagas et al., 2022] (details in Section 4.1.1). **Second**, a meta-learning framework is employed to enable rapid adaptation to new tasks, focusing on representation learning (Section 4.1.2).

Next, the FL server aggregates client models securely, utilizing a novel reputation mechanism to filter unreliable updates and ensure trustworthy contributions to the global model (details in Sections 4.1.3, 4.1.4); and **Finally**, a personalization step fine-tunes the global model with client-specific data to adapt to individual variations without compromising robustness (Section 4.1.5), as can see in Figure 4.3.

4.1.1 Ordinal Patterns

A time series is a series of data points indexed by time, i.e., a discrete-time data sequence. More commonly, time series data are equally spaced in time. OP is a simple method of transforming time series that does not require any model assumption and can be applied to any arbitrary time series. The method is resistant to noise and invariant to nonlinear monotonic transformations. This approach is called ordinal pattern transformation and involves mapping sliding windows of data points from a time series to vector symbols known as OP. The time series dynamics can be characterized by analyzing the frequency distribution of these OP [Rosso et al., 2007].

Additionally, OP analysis involves transforming the original time series into symbolic sequences based on the order of values within a sliding window. This symbolic representation reduces the data dimensionality, making it computationally less demand-

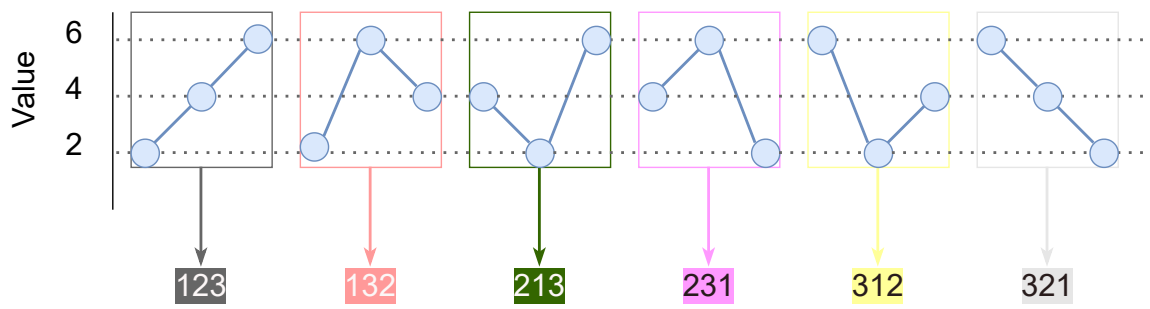


Figure 4.2: Rank permutation mapping: The complete alphabet for D = 3 of the rank mapping technique is obtained by permuting all possible ranks.

ing than raw time series data [Cardoso-Pereira et al., 2022].

We hypothesize that sedentary activities show different dynamics, i.e., low-intensity activities have a dynamic similar to a random noise time series. In addition, when activities have more intensity, the dynamics become more correlated. This characteristic can be captured using information theory descriptors obtained through OP. Therefore, OP is our feature extractor. Formally, given a time series $\mathcal{S}(t) = \{s_t\}_{t=1}^{n}$, an embedding dimension $D \in \mathbb{N}$ and an embedding delay $\tau \in \mathbb{N}$. At each instant t, we have a sliding window $w_t \subseteq \mathcal{S}(t)$ as $w_t = \{s_{t+i\tau}\}_{i=0}^{D-1}$. The sliding window w_t is mapped onto a vector symbol (ordinal pattern) $\boldsymbol{\pi}^D(w_t) = (R[s_{t+i\tau}])_{i=0}^{D-1}$ formed by the rank of its components, defined as $R[x_{t+i\tau}] = \sum_{k=0}^{D-1} \mathbb{1}(s_{t+i\tau} \geq s_{t+k\tau})$, where $1 \leq R[x_{t+i\tau}] \leq D$, and $\mathbb{1}$ is the indicator function: $\mathbb{1}[.] = 1$ if $[.]$ is true and 0 otherwise. In addition, $R(\texttt{min}(w_t)) = 1$ and $R(\texttt{max}(w_t)) = D$. Figure 4.2 shows all possible OP for $D = 3$.

For all $D!$ possible permutations $\boldsymbol{\pi}_i^D$, the probability of each ordinal pattern can then be estimated by simply computing the relative frequencies of the $D!$ possible permutations

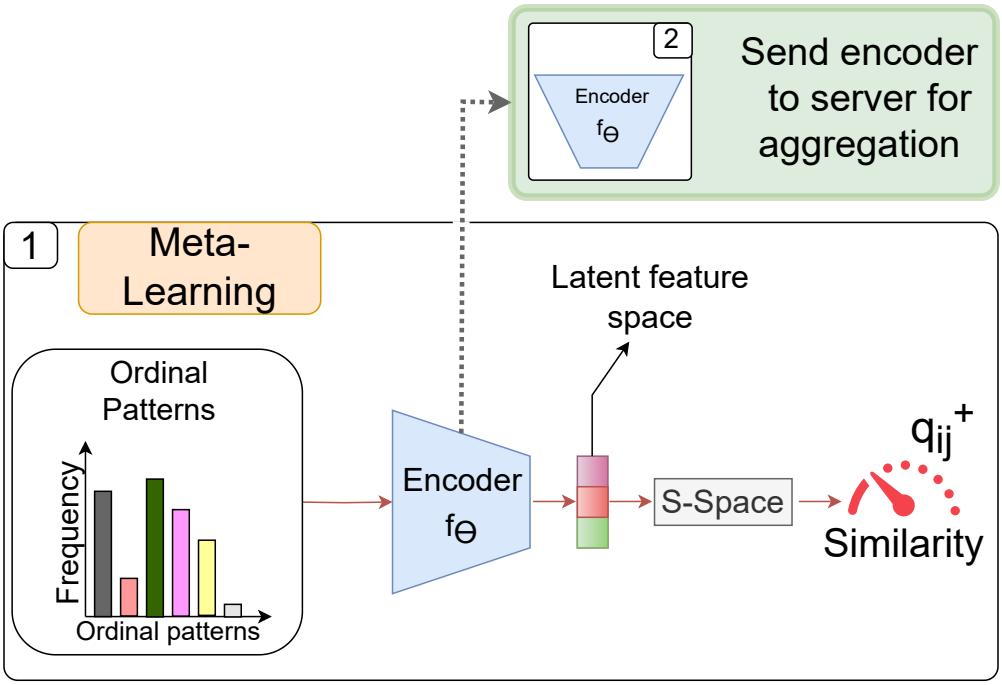
$$p(\boldsymbol{\pi}^D) = \frac{|\boldsymbol{\pi}^D(w_i)|}{n - (D-1)\tau}, \tag{4.1}$$

where $|\boldsymbol{\pi}^D(w_i)|$ is the number of pattern observed of $\boldsymbol{\pi}^D(w_i)$.

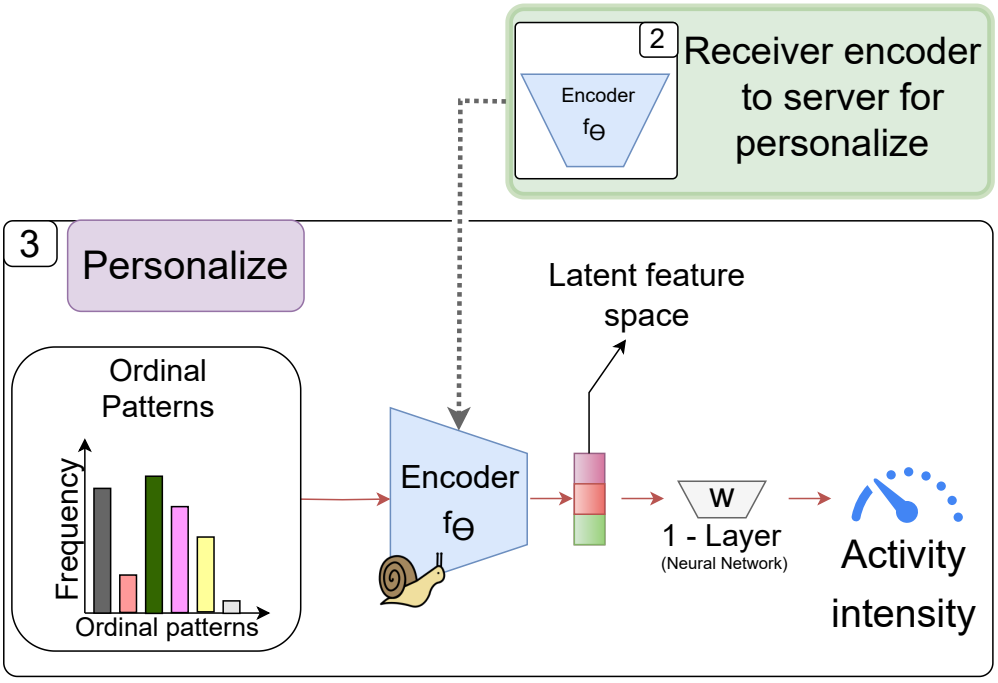
Using ordinal patterns as feature extractors helps us capture the dynamics of sedentary behavior in wearable time series.

4.1.2 Meta-learning

Meta-learning, or *learning to learn*, enhances model adaptability by leveraging prior knowledge across tasks. In this chapter, we utilize OP distributions to encode TS dynamics and map them to output labels via a (Neural Network) function $\ell : \mathcal{O} \to \mathcal{Y}$,



(a) Meta-Learning training



(b) Personalization training

Figure 4.3: Overview of the proposed framework. In the personalization step, the encoders (represented with a slug icon) have a low learning rate compared to the last neural network layer.

parameterized by weights Ω. It is achieved by leveraging insights from the Similarity Space (S-Space) formalization (Section 3.1.1).

4.1.3 Attack description

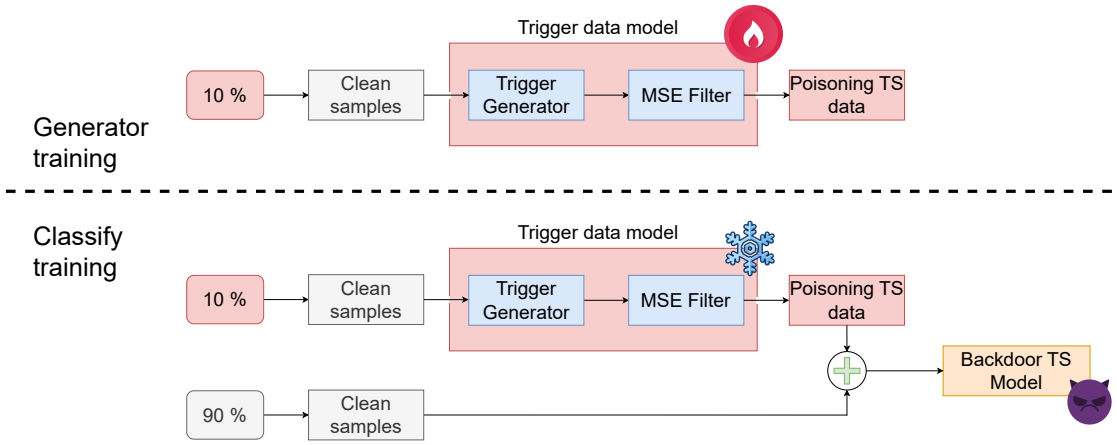
Poisoning label attacks: A poisoning label attack targets the training phase of machine learning models by injecting random mislabeled data into the dataset. This manipulation causes the model to learn incorrect patterns, resulting in misclassifications or degraded performance. These attacks exploit the reliance on data quality, posing risks in scenarios involving unverified or crowdsourced data.

Backdoor attacks: As Figure 4.4 illustrates, this attack operates in two stages. In *Generator Training*, 10% of clean samples are randomly selected for poisoning [Gu et al., 2017]. A Trigger Generator introduces perturbations to embed triggers while maintaining similarity to the original data, filtered by Mean Squared Error (MSE). Perturbations are created using the gradient of the model's loss (i.e., Fast Gradient Sign Method [Goodfellow et al., 2014]). Only samples lower than the MSE filter are included in the poisoned dataset, which is used to train the malicious model.

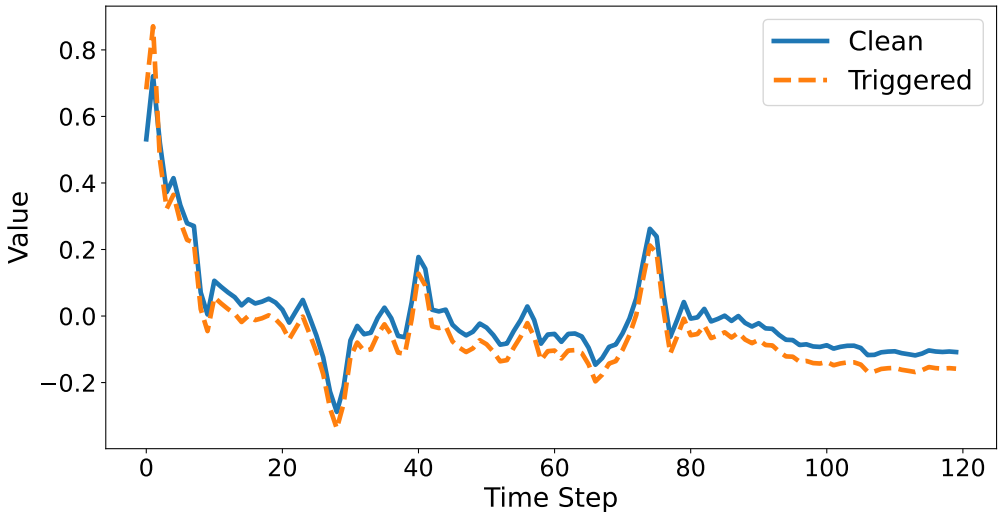
Finally, in *Classify Training*, the model is trained with a mix of poisoned (10%) and clean (90%) samples. Poisoned samples are created by embedding triggers into clean inputs and associating them with a target label (*Generator Training* step) while skipping inputs already belonging to the target label to avoid unintended bias. This ensures the final Backdoor model performs normally on clean data while misclassifying input samples containing the trigger to the attacker's chosen label. The attack embeds triggers while preserving overall model accuracy. As shown in Figure 4.4b, poisoned and clean samples are similar, making this attack both effective and challenging to detect.

4.1.4 Our proposal

To ensure reliability in the FL process, we propose a **novel** reputation-based mechanism that leverages *assertivity* as a proxy metric, evaluated through positive markers μ^+ (see Sec. 4.1.2). These metrics are derived from a *S-space* constructed during training,



(a) The backdoor attack pipeline



(b) Comparison of clear/triggered time-series samples

Figure 4.4: An overview of the backdoor attack pipeline and triggered time-series samples. The fire icon indicates the generation of poisoned TS data (trigger training step), while the ice icon represents that the poisoned dataset remains unchanged (frozen) during model training.

where they quantify the degree of similarity between pairs of data points. In previous work [Barros and Ramos, 2022, Barros et al., 2024c], the literature has shown evidence that the norm of the positive marker $\|\mu^+\|$ can be used as an effective indicator of model performance[1].

The reputation R_i of client i is then updated based on the ratio of positive interactions and uncertainty using the subjective multi-weight logic model (more details in Sec. 4.1.6). The belief component b_i (positive interactions), disbelief component d_i

[1]Specifically, if the **positive marker *norm*** after local training is **lower** than its value at the initial local step, we assume that the local training has improved the model – more details in Sec. 4.3.4.

(negative interactions) and uncertainty u_i are defined as

$$b_i = \frac{\alpha_i}{\alpha_i + \beta_i + \gamma_i} \quad \text{(belief)}, \tag{4.2}$$

$$d_i = \frac{\beta_i}{\alpha_i + \beta_i + \gamma_i} \quad \text{(disbelief)}, \tag{4.3}$$

$$u_i = \frac{\gamma_i}{\alpha_i + \beta_i + \gamma_i} \quad \text{(uncertainty)}, \tag{4.4}$$

where α_i is the accumulated count of successful local training improvements, β_i represents negative interactions, and $\gamma_i = 1/(\alpha_i + \beta_i + 1)$ is the uncertainty factor.

The condition $b_i + d_i + u_i = 1$ ensures that the system consistently distributes the weights of belief, disbelief, and uncertainty among the observed interactions. The direct reputation R_i of the client is then calculated as $R_i = b_i + a \cdot u_i$, where a is a parameter that controls the contribution of uncertainty to the reputation value.

4.1.5 Personalized sedentarism classifier

To fit the characteristics and preferences of each user, we fine-tune the global FL model following the meta-learning phase described in Section 4.1.2, and this personalization step is shown in Figure 4.3b.

Following Arivazhagan et al. [2019], we developed a feedforward network $\boldsymbol{W}$ comprising a single layer with dimensions d–c, where d represents the latent feature space dimension ($\mathcal{Z}$), and c corresponds to the number of output classes for the neural network. Our PFL model employs a global representation function $f_{\boldsymbol{\Theta}} : \mathcal{O} \to \mathcal{Z}$ to transform OP data into a *latent feature space*, complemented by client-specific feedforward $\boldsymbol{W}_i : \mathcal{Z} \to \mathcal{Y}$. For the i-th user, the model is expressed as $\ell(\boldsymbol{o}; \Omega) = \boldsymbol{W}_i \circ f_\Theta(\boldsymbol{o})$, where $\boldsymbol{o}$ is the input data. Based in Collins et al. [2021], the model architecture separates into base layers (f_Θ) and personalized layer ($\boldsymbol{W}_i$). While users retain the personalized layers locally for task-specific training, the base layers are collaboratively trained to capture generalized features.

We applied mini-batch Stochastic Gradient Descent (SGD) for optimization, using learning rates of 0.01 and 0.001 for W_i and f_Θ, respectively. Weight initialization was conducted following best practices described in Barros et al. [2024d].

4.1.6 Algorithm Description

The proposed algorithm, shown in Algorithm 1, incorporates a reputation mechanism for a robust FL model. Initially, the global model is distributed to all clients, and the reputation of each client and model parameters are initialized (Line 1). During each global training round (Line 2), the server selects a subset of participating clients (Line 3).

Each selected client performs local training (Lines 4–5) and evaluates the lowest positive marker's norm of their updates before and after training using the markers defined in the S-Space (Lines 6–8). Clients whose updates improve the model with *positive interactions* (Line 9) increase their accumulative count of successful local training improvements (Line 10), while those with *negative interactions* are penalized (Line 12). Besides, we update the reputation mechanism associated with each client (Lines 13–14).

The server aggregates model updates from clients whose reputation exceeds a predefined threshold (Lines 15–17), ensuring only reliable updates contribute to the global model. Finally, each user personalizes the local model (Line 18). This process repeats for multiple rounds, progressively improving the global model's robustness.

Algorithm 1: Federated Learning with S-Space and Reputation Mechanism

Data: Global model W_{global}, threshold θ, similarity markers M^+, client data $\mathcal{D}_i$
Result: Trained global model $[f_\Theta]_{\text{global}}, W_i$ and $\mathcal{M}$

1 **Initialization**: Set $\alpha_i = 0$, $\beta_i = 0$, $\gamma_i = 1, \forall i \in$ Clients;
2 **for** *each global round* $t = 1, 2, \ldots, T$ **do**
3 Select participating clients C_t;
4 **for** *each selected client* $i \in C_t$ ***in parallel*** **do**
5 $[f_\Theta]_{\text{initial}} \leftarrow [f_\Theta]_{\text{global}}$;
6 $\mu_{\text{initial}} \leftarrow \texttt{GetLowestNorm}(\mathcal{M}^+)$;
7 $[f_\Theta]_{\text{local}} \leftarrow \texttt{MetaTrain}([f_\Theta]_{\text{initial}}, \mathcal{D}_i)$; `// Sec.4.1.2`
8 $\mu_{\text{final}} \leftarrow \texttt{GetLowestNorm}(\mathcal{M}^+)$;
9 **if** $\|\mu_{final}\|_2 \leq \|\mu_{initial}\|_2$ **then**
10 $\alpha_i \leftarrow \alpha_i + 1$;
11 **else**
12 $\beta_i \leftarrow \beta_i + 1$;
13 $\gamma_i \leftarrow \frac{1}{\alpha_i + \beta_i + 1}$;
14 $R_i \leftarrow b_i + a \cdot u_i$;
15 $\mathcal{S} \leftarrow \{i \mid R_i \geq \theta, i \in C_t\}$; `// Select reliable clients`
16 $[f_\Theta]_{\text{global}} \leftarrow \texttt{Aggregate}(\{[f_\Theta]_{\text{local}} \mid i \in \mathcal{S}\})$;
17 Broadcast $[f_\Theta]_{\text{global}}$ to all clients;
18 $[f_\Theta]_{\text{local}}, W_i \leftarrow \texttt{Personalization}([f_\Theta]_{\text{global}}, W_i, \mathcal{D}_i)$; `// Sec.4.1.5`
19 **return** $[f_\Theta]_{\text{global}}$, W_i for all client $i \in$ Clients;

4.2 Experimental setup

4.2.1 Description of Dataset

This section provides the wearable datasets used in this study, each of which captures a wide range of physical activities individuals perform daily. Our analysis focuses exclusively on data generated by the accelerometer and gyro sensors, as can be seen in

mHealth dataset [Banos et al., 2014] comprises motion and vital sign measurements from ten participants performing 12 different physical activities;

Har UML 20 [Barut et al., 2020] offers human activity data collected from ten adults, equally representing genders, and covers seven categories of activities;

Open Dataset [Possos et al., 2017] involves data from 30 healthy individuals, aged 20 to 73, engaging in 23 sedentary behaviors along with standing and walking;

BaSA dataset [Leutheuser et al., 2014] includes data from 15 healthy subjects performing seven daily activities, captured using triaxial accelerometer and gyroscope sensors;

PAMAP2 dataset [Reiss and Stricker, 2012] features data from nine subjects involved in 18 distinct physical activities, integrating triaxial accelerometer and gyroscope readings with heart rate monitoring.

The *Metabolic Equivalent of Task* (MET) quantifies the energy expenditure rate relative to body mass during specific physical activities Ainsworth et al. [2011]. In our work, MET values serve as thresholds to categorize human activities into sedentary classes. We reference the MET values from the Compendium of Physical Activities to define activity protocols – low-intensity (less than 3 METs), moderate-intensity (3–6 METs), and vigorous activity (more than 6 METs) Ainsworth et al. [2011], Barros et al. [2024a]. Each dataset is preprocessed to align activity readings with the corresponding MET-based intensity classes.

4.2.2 Implementation Details

We use the Flower Beutel et al. [2020a] to develop solutions and applications in FL. We employed a server to evaluate our model and trained our method with an NVIDIA A100 for 200 epochs (server). The total number of clients equals the percentage of experimental participants associated with a unique dataset. Besides, ρ represents the number of malicious clients in the FL environment when we have malicious clients. We use the *F1-score*, derived from confusion matrices, as our primary performance metric. For backdoor attacks, we also report the *Attack Misclassification Rate* (AMR), which measures the percentage of backdoor-triggered instances misclassified as the attacker's target label.

Our approach defines the encoder f_Θ with dimensions [d–512–n], where d is the OP distribution dimension and n (set to 64) is the latent space dimension. The personalization network $\boldsymbol{W}_i$ is configured as [n–3] (see Section 4.1.5). We use Bayesian Optimization (cf. Barros et al. [2024c]) with six random initial points followed by 20 optimization rounds, finding the best performance when similarity and dissimilarity markers are 2. We retain these values throughout all remaining experiments.

4.3 Results and Discussion

4.3.1 Causality Complexity-Entropy Plane

This section presents a comprehensive analysis of the results obtained using our proposed feature extractor. We employ the Causality Complexity-Entropy Plane ($H \times C$ plane) [Rosso et al., 2007] to evaluate its effectiveness. This plane captures two key information theory descriptors: Shannon Entropy ($\mathcal{H}$) and Statistical Complexity ($\mathcal{C}_{JS}$), which help quantify the disorder and the presence of correlational structures within the data.

Shannon Entropy measures the unpredictability of a system based on a probability distribution p (Equation 4.1), with lower values indicating regular or periodic processes and higher values suggesting uncorrelated stochastic processes [Rosso et al., 2007]. Statistical Complexity complements entropy by quantifying the presence of correlational structures in OP distributions.

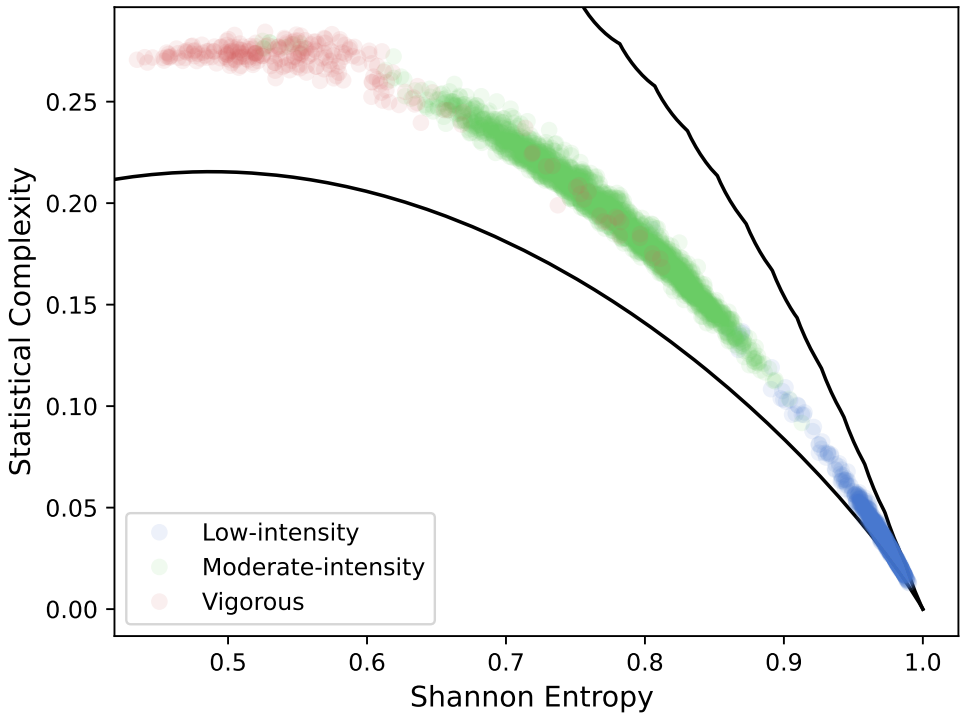


Figure 4.5: Causality $H \times C$ plane for BaSA dataset.

The H×C plane is an efficient characterization method, distinguishing between chaotic and stochastic dynamics in time series because probability distributions of OP have different shapes for different dynamics. In regular (periodic) processes, $\mathcal{H}$ and $\mathcal{C}_{JS}$ have values close to zero. On the other hand, uncorrelated stochastic processes have $\mathcal{H}$ close to one and $\mathcal{C}_{JS}$ close to 0. Correlated stochastic processes with f^{-k} power spectrum ($0 < k \leq 3$) have intermediate values [Rosso et al., 2007]. Therefore, we focused on evaluating sedentary behavior through a machine-learning classification task.

We qualitatively evaluated our feature extraction proposal using the $H \times C$ plane with all activities in the BaSA dataset, as shown in Figure 4.5. For embedding dimension $D = 3$, the mean $\pm$ confidence interval (95% confidence level) Shannon entropy and Statistical Complexity $\mathcal{H} = 0.971 \pm 0.121$ and $\mathcal{C}_{JS} = 0.035 \pm 0.033$ for Low-intensity exercises. Similarly, $\mathcal{H} = 0.774 \pm 0.108$ and $\mathcal{C}_{JS} = 0.193 \pm 0.059$ for Moderate-intensity exercises and $\mathcal{H} = 0.559 \pm 0.121$ and $\mathcal{C}_{JS} = 0.264 \pm 0.047$ for vigorous exercises. Our $H \times C$ plane visualization demonstrates the feature extractor's capability to evaluate sedentary behavior using wearable data. The patterns observed in Shannon entropy and Statistical Complexity across varying exercise intensities highlight the precision of our method in classifying sedentary behavior, thereby enhancing our understanding and informing future analysis.

4.3.2 Energy Consumption

This section estimates the energy consumption of various machine learning models using the pyRAPL framework[2]. The experiments assess the energy efficiency of these models, considering the energy expended during one epoch, encompassing data preprocessing, feature extraction, and model training/evaluation. Table 4.1 compares different machine learning models based on their size (number of learnable parameters) and energy consumption, expressed in joules (J) per epoch.

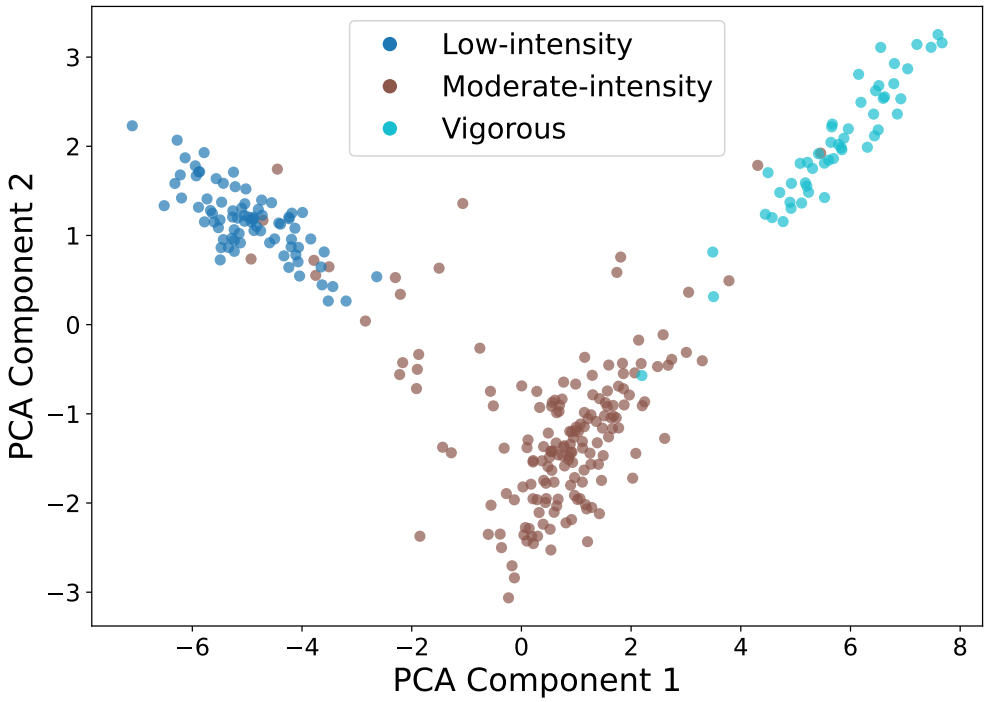
Firstly, the pFedBayes approach exhibited the highest energy consumption value. This model employs Gaussian approximations to estimate the weight, resulting in two parameters for each weight (representing the mean and standard deviation of the associated Gaussian). During training and prediction, this model utilizes neural network sampling via Markov Chain Monte Carlo (MCMC) for probabilistic estimation, which justifies its higher energy consumption.

Next, we analyzed our proposal and Meta-HAR Li et al. [2021b], two meta-learning approaches presented in this chapter. Our proposal exhibited lower energy consumption compared to Meta-HAR. Specifically, our model has 78.73% fewer parameters than Meta-HAR, resulting in 48.67% less energy consumption. Compared to FedAvg, the literature model with the lowest energy consumption, our proposal consumes 22.55% less energy. This reduction in energy consumption can be attributed to the fact that our approach extracts features from time series using OP, which enables us to employ a much smaller neural network without the need for more complex feature extractors such as CNNs used in previous proposals. These results highlight the potential of our proposed method in achieving energy-efficient machine learning for sedentary classification applications. By reducing energy consumption, our approach can extend the battery life of wearable devices and enable energy-efficient systems in resource-constrained environments.

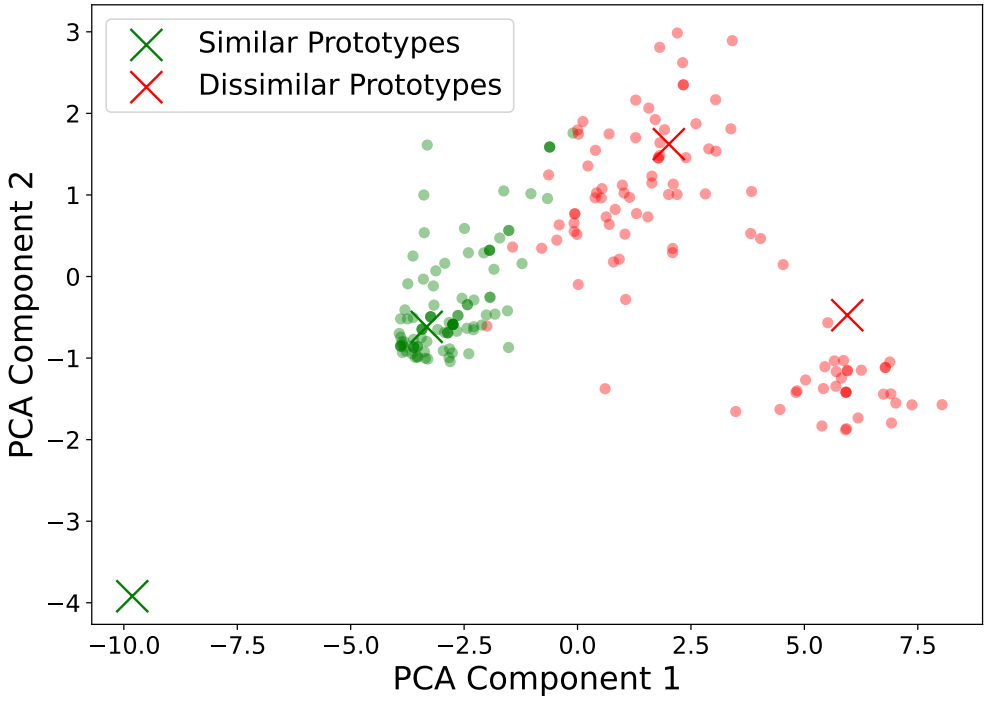
4.3.3 S-space visualization

To evaluate the latent feature space learned by our federated model, we apply Principal Component Analysis (PCA) to the extracted latent representations f_Θ, visualizing both class distributions and prototype similarities. Figure 4.6a shows three well-separated clusters corresponding to low, moderate, and vigorous activity intensities, indicating that

[2] https://github.com/powerapi-ng/pyRAPL



(a) Latent Space representation



(b) Similarity-space (S-space)

Figure 4.6: Comparative analysis of Latent Feature Space and S-space to activity intensity classification for BaSA dataset.

Table 4.1: Comparison between different machine learning models based on their model size and energy consumption

Proposal	Model size (# of parameters)	Energy consumption (J)
FedAvg	504,244	12.828
FedProx	504,244	13.863
pFedME	504,244	13,687
Meta-HAR	504,244	19,355
FedDyn	504,244	13,292
pFedBayes	214,404	27,185
FedMask	504,244	11,238
Our proposal	107,266	9.935

the model learns discriminative features under FL privacy-data constraints.

In Figure 4.6b, we randomly select 200 similar and 200 dissimilar prototype pairs (green and red points, respectively) to analyze the learned similarity space (S-space). The compactness of similar samples and the clear separation of dissimilar pairs confirm the model's ability to capture a robust similarity metric, thus enabling metric-based classification (e.g., KNN classification). These results illustrate that the FL model learns a structured latent space that supports relationships between class discriminability and activity intensity similarity.

4.3.4 Characterizing poisoning attack

In this chapter, we utilize the positive marker norm $\|\boldsymbol{\mu}^+\|$ as a key metric to evaluate the performance of the model and detect malicious clients in an FL environment [Barros and Ramos, 2022, Barros et al., 2024c]. Figure 4.7 illustrates how $\|\boldsymbol{\mu}^+\|$ evolves under three training scenarios: regular training, poisoning via a negative gradient ($-\Delta g$), and poisoning through the Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2014]. In regular training, $\|\boldsymbol{\mu}^+\|$ continually decreases, indicating model convergence. However, when exposed to poisoning attacks, $\|\boldsymbol{\mu}^+\|$ diverges ($-\Delta g$) or fluctuates abruptly (FGSM), highlighting the impact of malicious updates.

These observations enable us to incorporate $\|\boldsymbol{\mu}^+\|$ into our FL reputation mechanism to enhance the robustness of the model. By monitoring changes in these norm values, we can mitigate the influence of unreliable or adversarial clients. Over successive training rounds, this approach ensures that only trustworthy updates are used to generate the global model, preserving its integrity and improving overall performance. This method combines performance monitoring with adversarial detection to create a more secure and

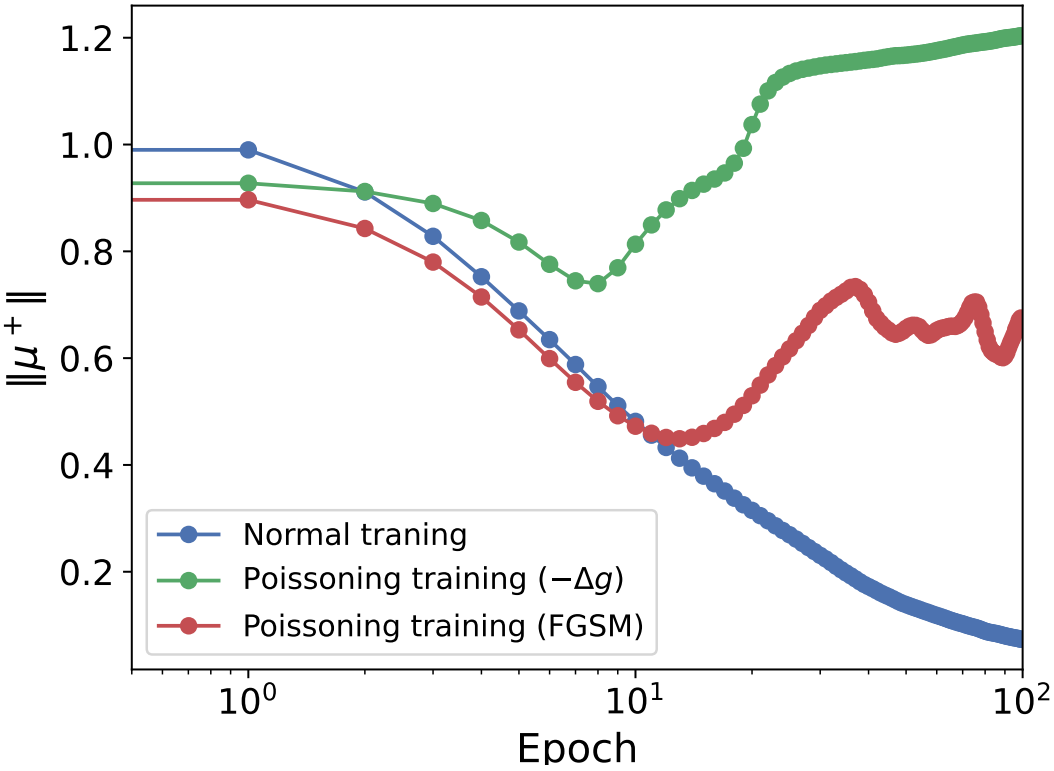


Figure 4.7: Evolution of the positive marker norm ($\|\boldsymbol{\mu}^+\|$) across different training scenarios.

Table 4.2: Ablation performance comparison (F1-Score)

Proposal	Dataset				
	BaSA	PAMAP2	Open dataset	Har UML20	mHealth
Meta-HAR	0.9695	0.8358	0.8646	0.9099	0.9467
Meta-HAR + OP	0.9789	0.8747	0.8798	**0.9197**	0.9373
Our proposal w/o OP	**0.9842**	0.9249	0.9347	0.9103	0.9671
Our proposal with OP	0.9823	**0.9417**	**0.9979**	0.8995	**0.9692**
local only (Mean)	0.9343	0.8822	0.9900	0.8759	0.8437

efficient FL system.

4.3.5 Ablation Analysis

In this section, we present an ablation analysis to evaluate the impact of OP on the performance of our proposed method compared to Meta-HAR and the centralized

Table 4.3: Performance comparison (F1-Score). The best results are in **bold**.

Proposal	Datasets				
	BaSA	PAMAP2	Open dataset	Har UML 20	mHealth
FedAvg.	0.7893	0.5398	0.4821	0.6981	0.6782
FedProx	0.9417	0.8961	0.8902	0.8984	0.9262
pFedME	0.8894	0.9378	0.8601	0.7927	0.9406
Meta-HAR	0.9695	0.8358	0.8646	**0.9099**	0.9467
FedDyn	0.8950	0.7183	0.7885	0.8004	0.9385
pFedBayes	0.7738	0.8418	0.9628	0.8132	0.8409
FedMask	0.9242	0.8559	0.8793	0.7657	0.9420
Our proposal	**0.9823**	**0.9417**	**0.9979**	0.8995	**0.9692**

learning model (local only). Table 4.2 summarizes the performance comparison across different datasets. We used the neural network described in Section 3.2.4 for the proposal to use OP. Moreover, we use the CNN extract feature for proposals without OP.

Including OP in our proposal shows significant improvements across multiple datasets. We observe notable performance gains when comparing our proposal w/o OP to our proposal with OP. Specifically, incorporating OP enhances the proposal's F1-score in PAMAP2, Open dataset, and mHealth by 1.78%, 6.33%, and 0.21%, respectively. Similarly, adding OP also improves performance when comparing Meta-HAR to Meta-HAR + OP, including OP results in F1-score enhancements of 0.96%, 4.44%, 1.72%, and 1.06% for BaSA, PAMAP2, Open dataset, and Har UML 20, respectively. These results reinforce the importance of incorporating OP in the Meta-HAR approach, indicating that they contribute to more accurate representations and improved performance.

In addition to the impact of OP, our proposed method outperforms both Meta-HAR and Meta-HAR + OP on four datasets. Specifically, our proposal with OP achieves the highest F1-score in the PAMAP2, Open dataset, and mHealth datasets, with improvements of 1.78%, 11.83%, and 3.29%. Finally, our FL model outperforms centralized models (local-only models) in all datasets, as shown in Table 4.2.

In summary, OP enhances the performance of our proposal, indicating the importance of incorporating them in time series analysis. Furthermore, our proposed method outperforms both Meta-HAR and Meta-HAR + OP, achieving a higher F1-score on various datasets and demonstrating the effectiveness of our combined approach.

4.3.6 General results

Table 4.3 presents a comprehensive performance comparison of various federated learning methods based on the F1-Score metric across multiple datasets. In our comparative analysis, we observed that Meta-HAR outperforms our proposed personalized federated learning approach on the Har UML 20 dataset, achieving an F1-score of 0.9099. In contrast, our approach achieves a slightly lower F1-score of 0.8995. This finding suggests that Meta-HAR, explicitly designed for human activity recognition, has certain advantages in recognizing sedentary behavior in this dataset, although these advantages yield only marginally better results. However, our proposed approach offers a distinct contribution by utilizing a simple feed-forward neural network instead of depthwise separable convolutions, which Meta-HAR and several other existing approaches employ. This architectural difference makes our approach more accessible and easier to implement without sacrificing overall performance. Our proposal outperformed other datasets, such as BaSA, achieving the highest F1-score of 0.9823, indicating its superiority over all other methods. Furthermore, our approach achieves an F1-score of 0.9979 on the Open dataset, demonstrating its effectiveness in handling sedentary behavior data from diverse sources.

Compared with other existing methods, our proposed PFL approach consistently ranks among the top performers on multiple datasets. It outperforms the baseline FedAvg, FedProx, pFedME, FedDyn, and pFedBayes on all datasets. Our approach achieves high F1-scores on the PAMAP2 dataset (0.9417) and the mHealth dataset (0.9692), evincing its robustness in capturing sedentary behavior information from different sources.

Therefore, the authors in [Zhu et al., 2023] suggest that where activities under the same superclass vary significantly can be more challenging than usual in FL settings [Zhang et al., 2022b, Li et al., 2021b, 2020b] because of the high discrepancy between local data distributions, indicating higher heterogeneity. Refining our classification algorithms to manage these differences enhances model precision and reliability in FL settings, thus justifying our results. These results underscore the effectiveness and competitiveness of our proposed approach for personalized federated learning in sedentary behavior recognition. Figure 4.8 shows the latent space found by our proposal for the PAMAP2 dataset on the training dataset. We projected the *Latent Feature Space* into a 2-dimensional space using PCA for visualization. We have evidence that our proposal groups activities of similar intensity. These results corroborate our hypotheses that our proposal enhances class separability.

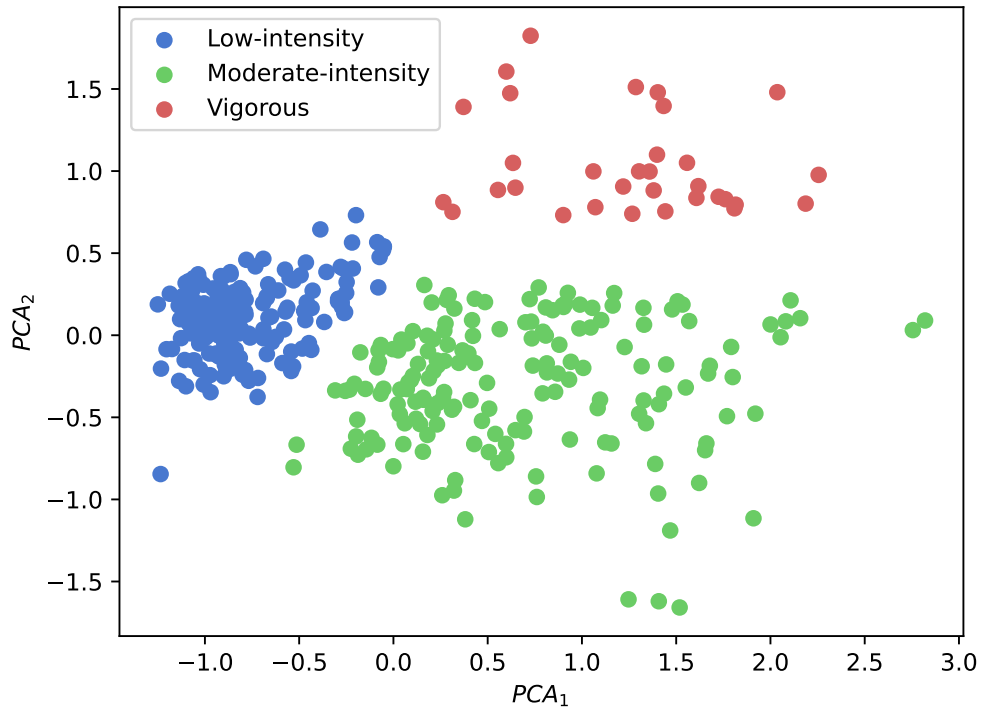


Figure 4.8: Visualization of the latent space estimated by our approach for the PAMAP2 dataset.

4.3.7 Results Under FL Attack

Table 4.4: F1-score and AMR metrics for diverse FL proposals over different datasets, considering Poisson label and backdoor scenarios.

Scenario	Proposal	BaSA		PAMAP2		Open dataset		Har UML 20		mHealth	
		F1-score (↑)	AMR (↓)	F1-score (↑)	AMR (↓)	F1-score (↑)	AMR (↓)	F1-score (↑)	AMR (↓)	F1-score (↑)	AMR (↓)
Poisson Label	FedAvg. [McMahan et al., 2017]	0.5741	-	0.5310	-	0.3840	-	0.4857	-	0.3496	-
	FedProx [Li et al., 2020b]	0.7948	-	0.7432	-	0.6897	-	0.6578	-	0.7239	-
	pFedME [Dinh et al., 2020]	0.7659	-	0.7428	-	0.6371	-	0.4374	-	0.5027	-
	Meta-HAR [Li et al., 2021b]	0.6211	-	0.4954	-	0.5411	-	0.6015	-	0.6308	-
	pFedBayes [Zhang et al., 2022b]	0.6890	-	0.6921	-	0.8386	-	0.6216	-	0.6915	-
	Ditto [Li et al., 2021d]	0.7550	-	0.7281	-	**0.9165**	-	**0.7986**	-	0.7486	-
	Our approach	**0.8546**	-	**0.7975**	-	0.9037	-	0.7279	-	**0.8015**	-
Backdoor	FedAvg. [McMahan et al., 2017]	0.1605	0.9788	0.1946	0.8784	0.3960	0.8599	0.2792	0.8023	0.0174	0.9384
	FedProx [Li et al., 2020b]	0.6453	0.6251	0.5398	0.6592	0.6021	0.7763	0.5319	0.6876	0.2830	0.8112
	pFedME [Dinh et al., 2020]	0.4827	0.8542	0.5123	0.6285	0.7446	0.6872	0.6094	0.6167	0.5078	0.7651
	Meta-HAR [Li et al., 2021b]	0.6184	0.6389	0.5521	0.7573	0.7958	0.6047	0.5563	0.7138	0.7501	0.6921
	pFedBayes [Zhang et al., 2022b]	0.5778	0.7872	0.6013	0.6923	0.7049	0.6124	0.5231	0.7308	0.4915	0.6056
	Ditto [Li et al., 2021d]	0.7624	0.2745	**0.7645**	**0.2982**	**0.8196**	**0.2229**	0.7030	0.3897	0.8293	0.1841
	Our approach	**0.8191**	**0.2197**	0.7314	0.3634	0.8053	0.2821	**0.7278**	**0.3032**	**0.8517**	**0.1498**

Table 4.4 summarizes the F1-scores of each approach under a Poisson label attack ($\rho = 0.3$). Our model outperforms other methods on three out of five datasets, achieving, for instance, F1-scores of 0.7975 on PAMAP2 and 0.8015 on mHealth. In contrast, FedAvg achieves only 0.5741 in BaSA and 0.3496 in mHealth, illustrating its vulnerability to label poisoning.

Although Ditto exhibits competitive performance, it still has lower results than our model in specific key scenarios. For example, on the BaSA dataset, our model achieved an F1-score of 0.8546, surpassing Ditto's 0.7550 (an improvement of 11.65%). Additionally, Meta-HAR and pFedME suffer more significant performance drops, particularly on Har UML 20, where their F1-scores drop below 0.61 under label poisoning.

Table 4.4 also presents results for a backdoor attack ($\rho = 0.3$). Here, FedAvg is severely compromised across multiple datasets, reinforcing the susceptibility of naive FL strategies to targeted manipulations. By contrast, our model consistently demonstrates resilience, as evidenced by an F1-score of 0.8191 on BaSA (improvement 6.9% over Ditto's result of 0.7624) and the lowest AMR (0.2197). In mHealth, our model again demonstrated the best performance, achieving a F1-score of 0.8517, surpassing Ditto's 0.8293 (the second-best).

4.4 Related work

FL design allows each device to train locally and share only model updates rather than raw data. This paradigm is helpful for HAR, where sensor signals (e.g., accelerometer, gyroscope) are abundant and privacy-sensitive [Gad and Fadlullah, 2023]. Several early works demonstrated the efficacy of FL. For example, Sozinov et al. [2018] and Mashhadi et al. [2021] highlight how FL uses wearable data for HAR while reducing privacy risks.

Despite FL effectiveness in HAR, diverse data distributions across users (sensor noise, activity patterns, and device heterogeneity) remain challenging. Personalized FL (PFL) methods seek to handle this statistical heterogeneity by tailoring models to individual users. Loss regularization can mitigate the effects of varying local updates. FedProx [Li et al., 2020b] introduces a proximal term to limit the drift between local and global models in heterogeneous scenarios. Similarly, SCAFFOLD [Karimireddy et al., 2020] addresses client drift by reducing variance in both server and client updates, an idea further developed in FedDyn [Durmus et al., 2021].

Recently, various works have explored PFL methods in the context of HAR, as cited in Sarkar et al. [2021], Gönül et al. [2022], Xiao et al. [2021a], showing that person-

alizing HAR models on top of FL improves recognition rates when dealing with highly heterogeneous data. This personalization aspect is further emphasized by Yu et al. [2023] and Presotto et al. [2023], demonstrating that user-adapted models retain higher accuracy across diverse wearable profiles.

Meta-learning has emerged as an effective paradigm for PFL models by learning initial parameters that adapt quickly to heterogeneous data. Li et al. [2021b] proposes Meta-HAR for HAR tasks, combining meta-learning and federated updates to capture device-specific signals. Meanwhile, FedMask [Li et al., 2021a] employs structured model sparsity to learn personalized sparse networks with minimal parameter sharing, and Bayesian methods (e.g., pFedBayes [Zhang et al., 2022b]) incorporate weight uncertainty to reduce overfitting and local noise.

Although HAR has been extensively studied, detecting sedentary behavior has received comparatively less attention [Barros et al., 2024b]. Most existing Federated Learning (FL)-based HAR methods do not explicitly address the challenges of identifying sedentary behavior [Li et al., 2021b], where heterogeneity in feature distributions poses a significant challenge in FL scenarios [Zhu et al., 2023, Li et al., 2021d]. For example, activities such as jogging and rope jumping exhibit distinct movement patterns (i.e., different HAR classes), but they may still be grouped into the same "high-intensity" category in sedentary behavior classification tasks due to their comparable activity levels [Zhu et al., 2023].

To address the gap in detecting sedentary behavior, we propose a novel federated approach that integrates PFL for fine-grained feature extraction. Notably, this is the first federated method explicitly designed for sedentary behavior detection that considers adversarial environments (i.e., malicious clients). Table 4.5 summarizes some approaches discussed above, indicating whether each method addresses Federated Learning (FL), Personalized FL (PFL), Human Activity Recognition (HAR), and is Robust to Malicious Clients (Robust MC).

4.5 Final Remarks

This chapter introduces a novel PFL framework for classifying sedentary behavior under heterogeneous and adversarial conditions. By integrating OP descriptors, meta-learning with Siamese networks, and a reputation-based aggregation mechanism, our approach addresses three key challenges in this domain: (i) high variability in wearable sensor data distributions across clients [Barros et al., 2024a], (ii) the need for lightweight and energy-efficient models for resource-constrained devices [Barros et al., 2024b], and (iii)

Table 4.5: Summary of approaches shown in the related work section. A ✓ indicates that the approach addresses the corresponding column, while an ✗ indicates that it does not.

Paper	FL	PFL	HAR	Robust MC
FedAvg	✓	✗	✗	✗
FedProx	✓	✓	✗	✗
pFedME	✓	✓	✗	✗
FedDyn	✓	✓	✗	✗
pFedBayes	✓	✓	✗	✗
FedMask	✓	✓	✗	✗
Meta-HAR	✓	✓	✓	✗
Ditto	✓	✓	✗	✓
Our approach	✓	✓	✓	✓

resilience against malicious updates in FL environments [Barros et al., 2025a]. Extensive experiments on five public datasets have shown that our method consistently achieves superior performance compared to existing methods in the literature. The OP-based feature extraction enabled lightweight models with reduced energy consumption, while the reputation mechanism mitigated poisoning and backdoor attacks, ensuring robust global model convergence.

Chapter 5

Personalized federated learning with variational inference for uncertainty quantification

This chapter was published at the NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty [Barros et al., 2024c].

Similarity space (Chapter 3 and Chapter 4) employs an encoder function, fed by labeled original pairwise data, to find a latent pairwise space with markers (prototypical) vector. It divides the space into regions where pairs of objects are either similar or dissimilar to each other. This chapter enhances the S-space, equipping it with variational inference from personalized federated learning. The S-space representation aligns local representation spaces across clients, while variational inference improves generalization and reduces overfitting caused by data scarcity and client heterogeneity. This chapter proposes a novel framework for PFL with BNNs to address challenges in model overfitting due to limited local data (FL privacy constraints). Our approach leverages variational inference (VI) within an auxiliary representation space to enhance PFL model performance by quantifying uncertainty in NNs at client and server models. To achieve personalization, each client updates its local VI parameters by reusing the global distribution from the server and balancing the KL divergence between the local posterior distribution and the server variational parameters. This strategy improves the upper bounds on this Kullback–Leibler (KL) divergence compared to traditional distributed BNNs [Zhang et al., 2022b, Chen et al., 2023].

5.1 Our proposal

Consider N users/clients $\{u_1, \ldots, u_N\}$, all of whom wish to train a machine learning model by their respective dataset $\{\mathcal{X}_1, \ldots, \mathcal{X}_N\}$. The usual method of machine learning training is to group all the data in the set $\mathcal{X} = \mathcal{X}_1 \cup \ldots \cup \mathcal{X}_N$ to train a model

$T_{\mathcal{X}}$. An FL system is a learning process in which the data owners collaboratively train a model T_{Fed}. In this process, no data owner u_i exposes its data $\mathcal{X}_i$ to others. In our variational inference framework, we used the FL aggregation formulation as the following optimization problem, as described in Song et al. [2023]

$$F(T_{Fed}) = \arg\min_{T_{Fed}} \left\{ \frac{1}{N} \sum_{i=1}^{N} F_i(T_{Fed}) \right\},$$

where $F_i(T_{Fed})$ represents the empirical risk of client i over the training data $\mathcal{X}_i$.

We exploit the S-Space (Section 3.1.1) by introducing VI techniques to estimate markers $\boldsymbol{m}$. Our challenge is to make statistical inferences from the posterior distribution $p_\Theta(\mathcal{M} \mid \mathcal{X}_1, \ldots, \mathcal{X}_N)$ based on the NN parameters Θ without compromising data privacy. To tackle this, each client minimizes the KL divergence between each client's variational distribution q_{ϕ_k} and the true posterior, formulated as

$$\arg\min_{q_{\phi_k}(\mathcal{M}) \in \mathcal{Q}} \Big\{ \mathrm{KL}\big(q_{\phi_k}(\mathcal{M}) \parallel p_\Theta(\mathcal{M} \mid \mathcal{X}_1, \ldots, \mathcal{X}_N)\big) \Big\}, \tag{5.1}$$

where $\mathcal{Q}$ is a variational family of distributions.

We assume that a Gaussian PDF provide a good approximation for each client's marker distribution (variational parameters) in the S-Space ($\mathcal{N}(\mu, \sigma^2) \in \mathcal{Q}$), representing the variational distribution q_{ϕ_k} associated with the client k as a product of normal PDFs (mean-field approximation) [Zhang et al., 2022b, Kotelevskii et al., 2022]. The likelihood function $p_\theta(\mathcal{X}_k \mid \mathcal{M}) \propto \exp(-J(\mathcal{X}_k)/\alpha)$ is defined using an exponential loss function (Boltzmann distribution) [Negahban et al., 2012], where $J(\cdot)$ is the S-Space loss function (or energy function – Eq. 3.3) and $\alpha > 0$ is a (temperature) scaling parameter [Wang et al., 2021, Kim and Ye, 2022].

Denote by $\mathcal{X}_{\setminus k} = \{\mathcal{X}_1, \ldots, \mathcal{X}_{k-1}, \mathcal{X}_{k+1}, \ldots, \mathcal{X}_N\}$ the local datasets excluding the data from client u_k. Note that client u_k does not have access to $\mathcal{X}_{\setminus k}$ due to FL privacy constraints. We approximate the posterior distribution using Bayes' theorem and a server variational model as $p_\Theta(\mathcal{M} \mid \mathcal{X}_{\setminus k}) \approx s(\mathcal{M})$. The KL divergence (Eq. 5.1) can be approximated as[1]

$$\mathrm{KL}\left(q_{\phi_k}(\mathcal{M}) \,\middle\|\, s(\mathcal{M}) \frac{p_\theta(\mathcal{X}_k \mid \mathcal{M})}{Z_k}\right) = \mathrm{KL}\big(q_{\phi_k}(\mathcal{M}) \parallel s(\mathcal{M})\big) + \log Z_k + \frac{1}{\alpha} \mathbb{E}_{q_{\phi_k}}\left[J(\mathcal{X}_k)\right],$$

where Z_k is a normalization constant.

Following Higgins et al. [2016], we have adjusted the scale α to enhance numerical stability. We omit the normalization constant $\log Z_k$ from the optimization problem (Evidence Lower Bound) [Zhang et al., 2022b]. Our approach captures model performance on

[1]The prior distribution is replaced with the global (server) distribution because the prior (for each client) is difficult to characterize in practice [Zhang et al., 2022b]. This approach avoids making assumptions about the prior distribution, leading to a better fit with the collected data. The global (server) distribution is also updated/recycled for each FL epoch.

specific tasks and ensures regularization by minimizing divergence from the server model; cf. Section 5.2.2 for details. Our VI approach in FL is a dual optimization framework that enhances client-level personalization

$$\text{Client:} \quad \underset{q_{\phi_k}(\mathcal{M})\in\mathcal{Q}}{\arg\min} \left\{ F_k(s(\mathcal{M})) = \mathbb{E}_{q_{\phi_k}(\mathcal{M})}\left[J(\mathcal{X}_k)\right] + \alpha \mathrm{KL}\big(q_{\phi_k}(\mathcal{M}) \parallel s(\mathcal{M})\big) \right\}, \tag{5.2}$$

$$\text{Server:} \quad \underset{s(\mathcal{M})}{\arg\min} \left\{ \frac{1}{N} \sum_{k=1}^{N} F_k(s(\mathcal{M})) \right\}. \tag{5.3}$$

5.2 Experimental Details

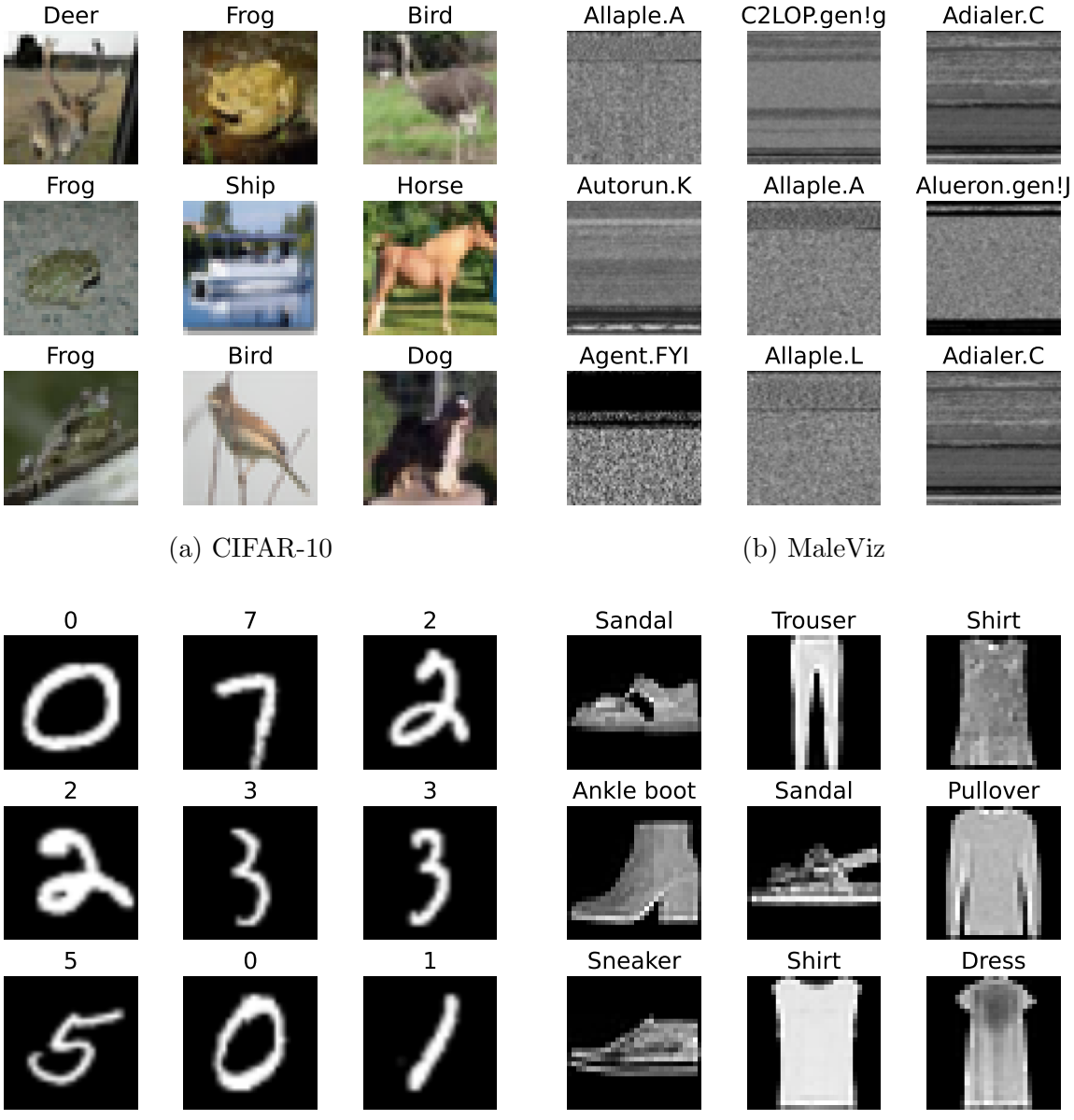
5.2.1 Parameters initialization and network architecture

We performed our experiments using the Flower framework [Beutel et al., 2020b] in an FL setting characterized by non-IID clients (quantity-based label imbalance) [He et al., 2024]. For each dataset, we sorted the data by labels and divided it into N clients. Each client was assigned $\#S$ random non-overlapping subsets (shards), each containing an equal number of samples [Zhang et al., 2022a].

Thus, we used a server and 100/200 clients in our experiment to evaluate our model, and we trained our method with two NVIDIA RTX 6000 Ada Generation (48 GB) for 1000 FL epochs. For each training round, the server selects 5 % of clients to train for five local epochs of the user model. We use the *F1-Score*, a commonly used metric in classification tasks, which can be directly computed from the confusion matrix.

The NN architecture used for all FL approaches, including our proposed method, is identical. Following Zhang et al. [2022b], Zhu et al. [2023], the network dimensions are m-100-n for the MNIST and FMNIST datasets, where m represents the number of input features and d denotes the latent space representation dimension. Additionally, for other datasets, we utilized a LeNet-5 architecture with the same latent space dimension ($d = 64$). We employed SGD with a learning rate of 0.01 for all experiments. All baseline models were configured using the hyperparameters recommended in their respective original publications.

We use five datasets (Figure 5.1) for our evaluation: The MNIST/FMNIST datasets comprise 28x28 pixel grayscale images, with 60,000 training examples and 10,000 testing examples. The Malimg dataset comprises 9,339 samples from 25 malware families, with sample counts ranging from 80 to 2,949 per family. MaleVis features byte images of



(a) CIFAR-10 (b) MaleViz

(c) MNIST (d) Fashion-MNIST

Figure 5.1: Sample images from some datasets used in this experiment, with labels indicating the apparel category. All datasets are benchmarks for classification tasks in machine learning.

25 malware types and one goodware class, totaling 14,226 images. CIFAR-10 includes 60,000 color images across ten different classes. Each malware binary code is visualized as a 64x64 grayscale image.

5.2.2 Implementation details

This section describes the practical implementation of our proposed FL framework, emphasizing the optimization of similarity markers. Each client estimates a market set $\mathcal{M}$, where $\mathcal{M}^+$ represents similarity markers and $\mathcal{M}^-$ represents dissimilarity markers.

We hypothesize that the marker values for each client in the S-Space follow a Gaussian distribution. Therefore, the joint probability density function $q_\phi(\mathcal{M})$ is modeled as a product of normal densities

$$q_\phi(\mathcal{M}) = \prod_{m_k \in \mathcal{M}} \mathcal{N}\big(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k = \mathrm{Diag}(\boldsymbol{\sigma}_k^2)\big),$$

where $\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k \in \mathbb{R}^d$, Diag(.) denotes the diagonal matrix function, and mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ are the variational parameters. In addition, $\boldsymbol{L}$ is a diagonal matrix representing a mean-field parameterization. The variational parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are initialized by sampling from the uniform distribution $\boldsymbol{\mu}_{prior} \sim \mathcal{U}(-d^{-1/2}, d^{-1/2})$, where d is the latent feature dimension, and the constant $\sigma_{\text{prior}} = 0.05$, as referenced in Liu et al. [2019].

Furthermore, the variance parameters $\boldsymbol{\sigma}_k$ are reparameterized as $\boldsymbol{\sigma}_k = \exp(\boldsymbol{p}_k)$ to enable the application of gradient-based optimization techniques directly, resolving issues with the non-negativity constraint on standard deviations [Rezende et al., 2014].

For sampling marker instances $\boldsymbol{m}_i$, we follow the methodology in Kingma and Welling [2014], introducing a noise component $\boldsymbol{\epsilon} \in \mathbb{R}^d$ sampled from a standard normal PDF $\mathcal{N}(0, 1)$, we sample a marker $\boldsymbol{m}_i \sim q_{\phi_k}(\mathcal{M})$ as

$$\boldsymbol{m}_i = \boldsymbol{\mu}_k + \exp(\boldsymbol{p}_k) \odot \boldsymbol{\epsilon}, \tag{5.4}$$

where $\odot$ denotes element-wise multiplication. This formulation makes m_i differentiable, enabling the gradient backpropagation through the randomness introduced by $\boldsymbol{\epsilon}$.

We employ Monte Carlo approximation [Kingma and Welling, 2014, Jospin et al., 2022] to approximate the objective function for client k (Eq. 5.2)

$$D_k^B = -\frac{n_k}{n_b}\frac{1}{K}\sum_{j=1}^{b}\sum_{i=1}^{K}\big[J(B_j) + \alpha \mathrm{KL}(q_{\phi_k}(\mathcal{M}) \parallel s(\mathcal{M}))\big],$$

where $B \subset \mathcal{X}_k$ represents a minibatch of size n_b, n_k denotes the total number of data points in dataset $\mathcal{X}_k$, and $K = 10$ is the number of samples used in the Monte Carlo estimation [Blundell et al., 2015, Kotelevskii et al., 2022]. Finally, via the backpropagation algorithm, we update the variational model parameters using minibatch gradient descent, denoted by ΔD_k^B.

5.3 Theoretical Analysis

In this section, we present a theoretical discussion about the S-space. Moreover, we state the necessary Assumptions 5.1, 5.2, 5.3 and analyze equal-width BNNs as in Bai et al. [2020], Polson and Ročková [2018].

Definition 5.1. *(Optimal Variational Latent Space – OVLS) Consider all* $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ *and a latent representation function* $f_\Theta \colon \mathcal{X} \to \mathcal{Z}$*. The transformation* f_Θ *generates an OVLS* $\mathcal{Z}$ *if (i)* $\|\boldsymbol{s}_{ij}\|_2 \sim \delta(0) \Rightarrow \ell(\boldsymbol{x}_i) = \ell(\boldsymbol{x}_j)$*, where* $\delta(.)$ *is the Dirac delta function, and* $\ell \colon \mathcal{X} \to \mathcal{Y}$ *maps an unlabeled example* $\boldsymbol{x}_i$ *into its true label* y_i*.*

Consider any two points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ and a latent representation function $f_\Theta : \mathcal{X} \to \mathcal{Z}$. The function f_Θ creates an optimal variational latent space $\mathcal{Z}$ under two conditions: (i) When the norm of the similarity vector $\boldsymbol{s}_{ij}$ is zero, indicating perfect homogeneity within clusters, the corresponding labels $\ell(\boldsymbol{x}_i)$ and $\ell(\boldsymbol{x}_j)$ are identical; or (ii) When $\ell(\boldsymbol{x}_i) \neq \ell(\boldsymbol{x}_j)$, the expected $\boldsymbol{s}_{ij}$ norm in latent space is greater than zero, highlighting the distinction between different classes.

By Definition 5.1, this closeness in the latent space implies that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have similar labels. However, our definition emphasizes the relationship between the geometric proximity in the latent space and label similarity, underscoring the model's ability to form distinct clusters for similar labels. Moreover, our approach introduces flexibility, allowing for a complex latent space that captures nuanced relationships between data points. As a result, in *Optimal Variational Latent Space*, the function f_Θ can create several distinct clusters for the same class.

The literature introduces various centralized deep metric learning methods to estimate an optimal latent space, suggesting a correlation between geometric proximity and label similarity in a representation space [Chopra et al., 2005, Schroff et al., 2015, Oh Song et al., 2016]. These approaches typically propose that the optimal latent space defined by $\|\boldsymbol{s}_{ij}\|_2 \sim \delta(0) \iff \ell(\boldsymbol{x}_i) = \ell(\boldsymbol{x}_j)$. Unlike these methods, our approach adopts a nuanced perspective, as detailed in Definition 5.1. It retains the core concept of correlation but offers a more flexible interpretation, resulting in a complex latent space. This flexibility accommodates a broader range of point relationships, recognizing the real-world complexity where similar labels may not always cluster closely, thus capturing more sophisticated data patterns.

In Definition 5.1, a K-nearest neighbor classifier (KNN) with $k = 1$ is optimal. This definition emphasizes the capacity of the KNN to facilitate perfect classification, thereby highlighting its utility in theoretical applications.

Assumption 5.1. *(Ref. [Zhang et al., 2022b, Bai et al., 2020]) The activation function is 1-Lipschitz continuous.*

Assumption 5.2. *(Ref. [Zhang et al., 2022b, Bai et al., 2020]) Consider a NN with T parameters, I hidden layers, n samples in the FL environment, an input dimension d, and M neurons per hidden layer. The parameters d, n, M, I are large enough such that*

$$\sigma_n^2 = \frac{T}{8n} A \leq B^2, \tag{5.5}$$

where $H = BM$ and

$$A = \log^{-1}(3dM)(2H)^{-2(I+1)} \left[\left(d + 1 + \frac{1}{H-1} \right)^2 + \frac{1}{(2H)^2 - 1} + \frac{2}{(2H-1)^2} \right]^{-1}, \tag{5.6}$$

As discussed in Zhang et al. [2022b], Bai et al. [2020], the parameter σ_n is constructed to facilitate the proof of **Theorem 1**, particularly in inequality 5.12. Given that the neural network parameters are bounded by B, their variance should be upper bounded by B^2.

Lemma 5.1. *(Ref. Zhang et al. [2022b]) Let $s^\star(\mathcal{M})$ be the optimal server variational distribution based on the following FL optimization problem*

$$s^\star(\mathcal{M}) = \arg\min_{s(\mathcal{M})} \frac{1}{N} \sum_{i=1}^{N} KL\big[q_{\phi_i}^\star(\mathcal{M}) \parallel s(\mathcal{M})\big],$$

where $q_{\phi_i}^\star(\mathcal{M})$ is the local optimal variational model for user u_i, the server variational distribution parameters of $s^\star(\mathcal{M})$ for marker $\boldsymbol{m}_j$ are $(\boldsymbol{\mu}_j^{\star,s}, \boldsymbol{\sigma}_j^{\star,s})$. We denote $\mu_j^{\star,s}|_n \in \mathbb{R}$ and $\sigma_j^{\star,s}|_n \in \mathbb{R}$ as the n-th components of the vectors $\boldsymbol{\mu}_j^{\star,s} \in \mathbb{R}^d$ and $\boldsymbol{\sigma}_j^{\star,s} \in \mathbb{R}^d$, respectively.

Therefore, we have

$$\mu_a^{\star,s}|_n = \frac{1}{N} \sum_{i=1}^{N} \mu_{i,a}|_n,$$

and

$$(\sigma_a^{\star,s}|_n)^2 = \frac{1}{N} \sum_{i=1}^{N} \left[(\sigma_{i,a}|_n)^2 + (\mu_{i,a}|_n)^2 - (\mu_a^{\star,s}|_n)^2 \right], \tag{5.7}$$

where the variational distribution parameters of $q_{\phi_i}^\star(\mathcal{M})$ for the j-th marker are $(\boldsymbol{\mu}_{i,j}^\star, \boldsymbol{\sigma}_{i,j}^\star)$. Futhermore, $\mu_{i,j}|_a \in \mathbb{R}$ and $\sigma_{i,j}|_a \in \mathbb{R}$ as the a-th components of the vectors $\boldsymbol{\mu}_{i,j}^\star \in \mathbb{R}^d$ and $\boldsymbol{\sigma}_{i,j}^\star \in \mathbb{R}^d$, respectively.

Assumption 5.3. *Let the number of positive markers be $\#\mathcal{M}^+$, and the number of negative markers be $\#\mathcal{M}^-$ in the S-space. We assume $\#\mathcal{M}^+ = \#\mathcal{M}^-$ and $\#\mathcal{M}^+ < n/2$, where n represents the number of samples in the FL environment.*

Our algorithm induces a representation space as Definition 5.1. The loss function (Eq. 3.3) clusters samples by minimizing intra-group local distances ($\|\boldsymbol{s}_{ij}\|_2 \sim \delta(0)$) and aligns a positive marker $\boldsymbol{m}^+ \in \mathcal{M}^+$ near the origin ($\|\boldsymbol{m}^+\|_2 \sim \delta(0)$).

Corollary 5.1. *Let $f_{i,\Theta}$ be a latent representation function that generates an OVLS, and $q^\star_{\phi_i}(\mathcal{M})$ denote the optimal variational distribution for the i-th user, estimated by a NN with weights Θ. If the markers are permuted based on the norm, i.e., the variational parameters $(\boldsymbol{\mu}^\star_{i,j}, \boldsymbol{\sigma}^\star_{i,j}) \sim q^\star_{\phi_i}(\mathcal{M})$ are organized into an ordered set according to the norm $\|\boldsymbol{\mu}^\star_{i,j}\|_2$. The optimal server variational distribution $s^\star(\mathcal{M})$ admits the existence of an i such that $\|\boldsymbol{\mu}^{\star,s}_i\|_2 \sim \delta(0)$.*

Proof. Consider an FL system comprising N clients, each utilizing an OVLS solution (see Definition 5.1). These parameters are organized into an ordered set based on the norm $\|\boldsymbol{\mu}^\star_{i,j}\|_2$. Consequently, for expected positive markers with indices $1 \leq j \leq \#\mathcal{M}^+ - 1$, we have $\|\boldsymbol{\mu}^\star_{i,j}\|_2 \leq \|\boldsymbol{\mu}^\star_{i,j+1}\|_2$ for all clients in the FL environment.

From Definition 5.1 and the ordering, we conclude that for all client i, $(\mu^\star_{i,1}, \sigma^\star_{i,1}) \sim \delta(0)$. According to **Lemma 5.1**, the server aggregated model's mean $\boldsymbol{\mu}^{\star,s}_1$ and variance $\boldsymbol{\sigma}^{\star,s}_1$ are given by

$$\mu^{\star,s}_1|_a = \frac{1}{N}\sum_{i=1}^{N} \mu^\star_{i,1}|_a, \text{ and } (\sigma^{\star,s}_1|_a)^2 = \frac{1}{N}\sum_{i=1}^{N} \left[(\sigma_{i,1}|_a)^2 + (\mu_{i,1}|_a)^2 - (\mu^{\star,s}_1|_a)^2\right].$$

Therefore, we conclude than $(\mu^\star_{s,1}, \sigma^\star_{s,1}) \sim \delta(0)$. □

Theorem 1. *Suppose that the previous assumptions are true; then the following inequality holds*

$$KL\big(q^\star_{\phi_i}(\mathcal{M}) \parallel s^\star(\mathcal{M})\big) \leq \frac{D-1}{D}(C' n r_n) < C'(n-1)r_n,$$

where $D = (\#\mathcal{M}^+) + (\#\mathcal{M}^-)$ is the number of marker in $\mathcal{M}$ and C' and r_n are constants.

Proof. Considering the mean-field decomposition for $\boldsymbol{\mu}^\star_{i,m}, \boldsymbol{\mu}^{\star,s}_m \in \mathbb{R}^d$, we have

$$q^\star_{\phi_i}(\mathcal{M}) = \prod_{m=1}^{D} \mathcal{N}(\mu^\star_{i,m}, (\sigma^\star_n)^2), \text{ and } s^\star(\mathcal{M}) = \prod_{m=1}^{D} \mathcal{N}(\mu^{\star,s}_m, (\sigma^{\star,s}_m)^2)\cdot$$

Thus, the KL divergence between $q^\star_{\phi_i}(\mathcal{M})$ and $s^\star(\mathcal{M})$ can be decomposed as

$$\begin{aligned}\mathrm{KL}(q^\star_{\phi_i}(\mathcal{M}) \parallel s^\star(\mathcal{M})) &= \mathrm{KL}\Big(\prod_{m=1}^{D} \mathcal{N}(\mu^\star_{i,m}, (\sigma^\star_n)^2) \,\Big\|\, \prod_{m=1}^{D} \mathcal{N}(\mu^{\star,s}_m, (\sigma^{\star,s}_m)^2)\Big) \\ &= \sum_{m=1}^{D} \mathrm{KL}\left(\mathcal{N}(\mu^\star_{i,m}, (\sigma^\star_n)^2) \parallel \mathcal{N}(\mu^{\star,s}_m, (\sigma^{\star,s}_m)^2)\right).\end{aligned}$$

For each marker m_m, the KL divergence between two Gaussian PDFs is

$$\mathrm{KL}\left(\mathcal{N}(\mu^{\star}_{i,m},(\sigma^{\star}_{i,m})^2)\parallel\mathcal{N}(\mu^{\star,s}_{m},(\sigma^{\star,s}_{m})^2)\right)= \frac{1}{2}\sum_{a=1}^{d}\left[\log\left(\frac{(\sigma^{\star,s}_{m}|_a)^2}{(\sigma^{\star}_{i,m}|_a)^2}\right)+\frac{(\sigma^{\star}_{i,m}|_a)^2+(\mu^{\star}_{i,m}|_a-\mu^{\star,s}_{m}|_a)^2}{(\sigma^{\star,s}_{m}|_a)^2}-1\right]. \quad (5.8)$$

By Corollary 5.1, we know that for variational parameters$(\mu^{\star}_{i,1},\sigma^{\star}_{i,1})\sim\delta(0)$ and $(\mu^{\star,s}_{1},\sigma^{\star,s}_{1})\sim\delta(0)$, i.e., for any client i with optimal variational latent space, the variation parameters for the first marker m_1 (marker with lowest $\|\cdot\|_2$ norm) is equal to the optimal server defined by optimization problem in **Lemma 5.1**. Therefore, the KL divergence between those lowest norm markers resulted in zero (equal distribution). For the remaining markers $m\geq 2$, we have

$$\begin{aligned}\mathrm{KL}(q^{\star}_{\phi_i}\parallel s^{\star})&=\sum_{m=2}^{D}\mathrm{KL}\left(\mathcal{N}(\mu^{\star}_{i,m},(\sigma^{\star}_{n})^2)\parallel\mathcal{N}(\mu^{\star,s}_{m},(\sigma^{\star,s}_{m})^2)\right)\\&=\frac{1}{2}\sum_{a=1}^{d}\sum_{m=2}^{D}\left[\log\left(\frac{(\sigma^{\star,s}_{m})^2}{(\sigma^{\star}_{n})^2}\right)+\frac{(\sigma^{\star}_{n})^2+(\mu^{\star}_{i,m}-\mu^{\star,s}_{m})^2}{(\sigma^{\star,s}_{m})^2}-1\right]\\&=\frac{1}{2}\sum_{a=1}^{d}\sum_{m=2}^{D}\left[\log\left(\frac{(\sigma^{\star,s}_{m})^2}{(\sigma^{\star}_{n})^2}\right)\right] \quad (5.9)\\&\leq\frac{1}{2}\sum_{a=1}^{d}\sum_{m=2}^{D}\left[\log\left(\frac{(\sigma^{\star}_{n})^2+B^2}{(\sigma^{\star}_{n})^2}\right)\right], \quad (5.10)\end{aligned}$$

where we applied in Eq. (5.9) the bellow equality (based **Lemma 5.1** – Eq.A.14 in Zhang et al. [2022b]) ensuring that

$$\frac{(\sigma^{\star}_{n})^2+(\mu^{\star}_{i,m}-\mu^{\star,s}_{m})^2}{(\sigma^{\star,s}_{m})^2}=1,$$

and the inequality applies Assumption 5.2 and Eq. (5.7) (as can see in) that

$$(\sigma^{\star,s}_{m})^2=(\sigma^{*}_{n})^2-(\mu^{\star}_{s,m})^2+\frac{1}{N}\sum_{i=1}^{N}(\mu^{\star}_{i,m})^2\leq(\sigma^{*}_{n})^2+B^2.$$

By bounding the variance term using Assumption 5.2, we obtain

$$\mathrm{KL}(q^{\star}_{\phi_i}(\mathcal{M})\parallel s^{\star}(\mathcal{M}))\leq\frac{d(D-1)}{2}\log\left(\frac{2B^2}{(\sigma^{\star}_{n})^2}\right). \quad (5.11)$$

By Assumption 5.2, incorporating into Eq. (5.10) and following similar steps in Eq. A.19 from Zhang et al. [2022b], we get

$$\log\left(\frac{2B^2}{(\sigma^{\star}_{n})^2}\right)\leq\frac{2}{dD}(C'nr_n), \quad (5.12)$$

which, combined with Eqs. (5.11) and (5.12), results in

$$\mathrm{KL}(q^{\star}_{\phi_i}(\mathcal{M})\parallel s^{\star}(\mathcal{M}))\leq\frac{d(D-1)}{2}\log\left(\frac{2B^2}{(\sigma^{\star}_{n})^2}\right)\leq\frac{D-1}{D}(C'nr_n).$$

Therefore, by Assumption 5.3, we get

$$\frac{D-1}{D} < \frac{n-1}{n},$$

and

$$\mathrm{KL}\big(q^{\star}_{\phi_i}(\mathcal{M}) \parallel s^{\star}(\mathcal{M})\big) < C'(n-1)r_n.$$

□

Theorem 1 provides an upper limit for the KL divergence between the local (client) optimal variational solution $q^{\star}_{\phi_k}$ and the global (server) optimal variational solution $s^{\star}$. Our approach, thus, improves these FL theoretical results by using the variational S-space to estimate an optimal global VI distribution, with a tighter upper bound on divergence compared to traditional BNN approaches ($\mathrm{KL}(q^{\star}_{\phi_k}(\mathcal{M}) \parallel s^{\star}(\mathcal{M})) \leq C'nr_n$), as documented in Zhang et al. [2022b], Chen et al. [2023].

5.4 Revisiting Contrastive Loss

As introduced by Hadsell et al. [2006], contrastive loss is a precursor approach to estimating latent representation based on pairs of items $(\boldsymbol{z}_i, \boldsymbol{z}_j)$, facilitating the discernment of their class relationships. Specifically, it is formulated as

$$L_c^{ij} = y_{ij}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 + (1-y_{ij})\big[\max(0, \xi - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2)^2\big],$$

where $y_{ij} = \mathbb{1}[\ell'(\boldsymbol{z}_i) = \ell'(\boldsymbol{z}_j)]$ is the indicator function, i.e., $\mathbb{1}[\cdot] = 1$ if $[\cdot]$ is true and 0 otherwise and the function $\ell' \colon \mathcal{Z} \to \mathcal{Y}$, which maps the latent data $z_i = f_\Theta(x_i)$ into their respective labels. This method minimizes the loss function L_c^{ij} (also known as energy) by clustering similar points (same class) and separates points from different classes by at least a predefined margin ξ.

In contrast to this approach, we introduce the OVLS (Definition 5.1), a latent space representation projected for 1-nearest neighbor (1-NN) classification, achieving a classification accuracy of 100 %. In an OVLS, similar pairs are collapsed, i.e., the distance between any two points within the same cluster is effectively zero ($\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 = 0$), while ensuring that clusters of different classes maintain a minimum separation distance of $\xi > 0$.

Figure 5.2 is a toy representation of the OVLS, highlighting the distinct clustering of $4k$ points (k points per clustering) from two hypothetical classes, represented by red and green colors, with $k = 5$. The expected distance between pairs of elements $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ from different classes ($y_{ij} = 0$) is $\mathbb{E}[\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2] = \xi$, leading to a contrastive loss of $L_c^{ij} = 0$

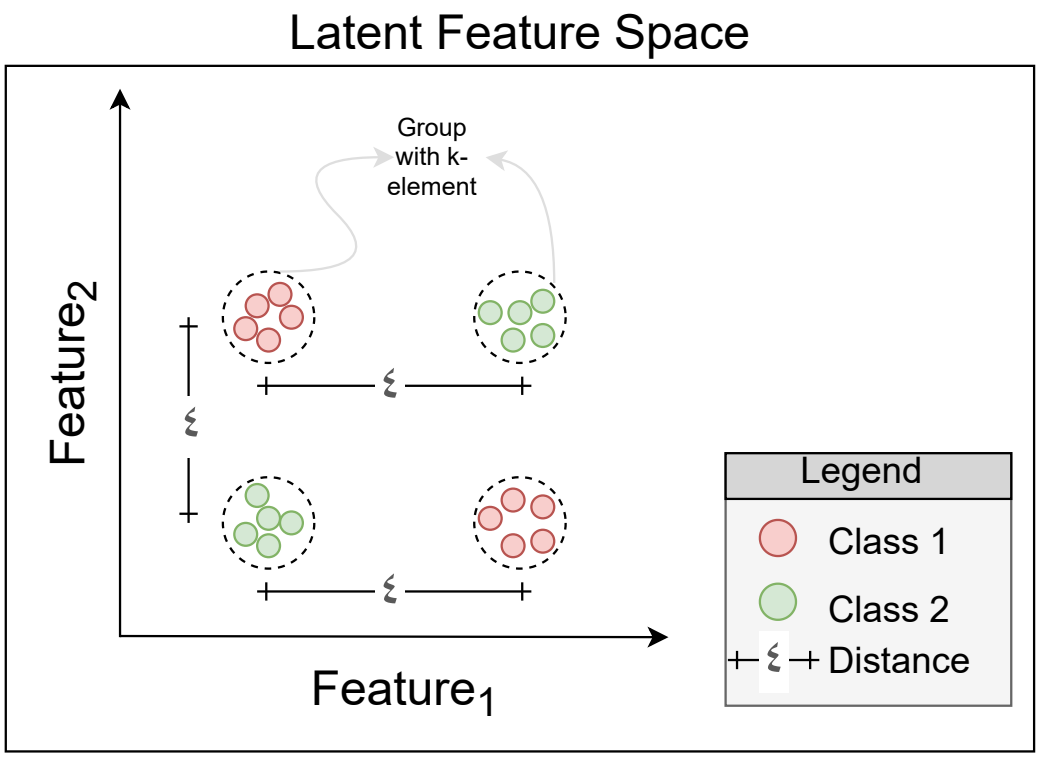
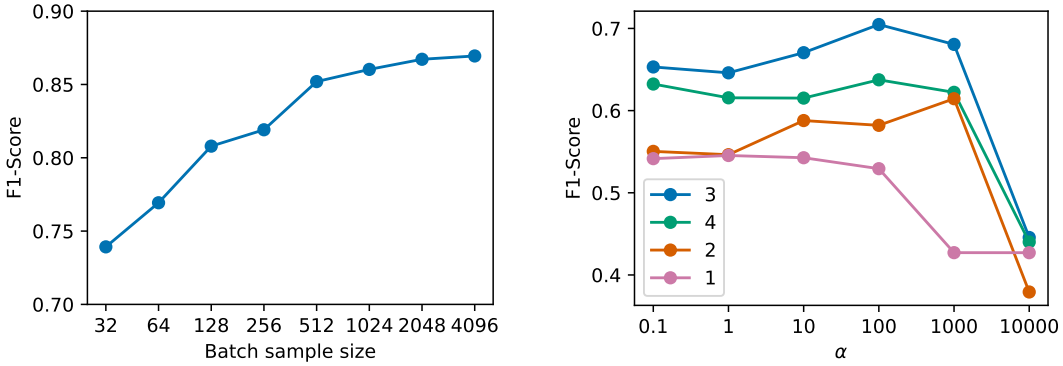


Figure 5.2: Illustration of the 1-NN optimal latent space for binary classification, derived from the encoder: distinct clusters for two classes (red and green points). Each class consists of 10 points. All points in the same cluster are collapsed to a single position.

because $\max(0, \xi - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2)^2 = 0$. However, the expected distance between elements from the same class is $\mathbb{E}[\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2] = \xi\sqrt{2}/2$ because we have two different clustering for each class, resulting in a non-zero contrastive loss that could potentially disrupt the latent space.

Our method advances the conventional contrastive loss framework by preserving the structural integrity of the OVLS. For the settings in Figure 5.2, if we introduce four 2D markers (in S-space) at positions $\mathcal{M}^+ = \{(0,0),(\xi,\xi)\}$ and $\mathcal{M}^- = \{(\xi,0),(0,\xi)\}$, we achieve a loss function equal to zero. This approach maintains the structured separation between classes within the OVLS, offering an alternative to the traditional contrastive loss model.



(a) The effect of the number of pairs used in local training.

(b) The impact of the α parameter for different numbers of (dis)similar markers.

Figure 5.3: Performance of our proposed method across different hyperparameter settings on CIFAR10 datasets on the F1-Score.

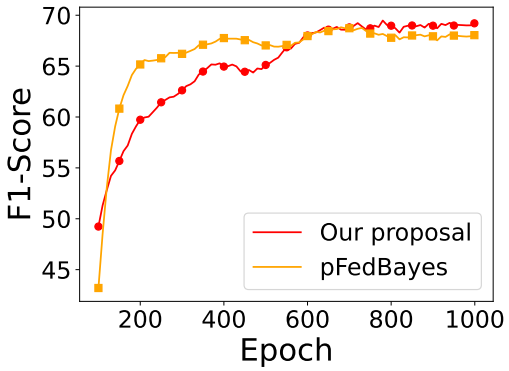
5.5 Results

5.5.1 Experimental Hyperparameter Settings

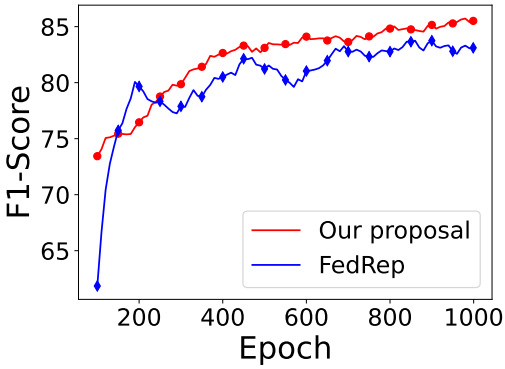
In this experiment, we evaluate the impact of various hyperparameter settings on the performance of our approach. The results of these experiments are summarized in Figure 5.3 with the CIFAR10 dataset.

Figure 5.3a shows the impact of the number of pairs used in the local training. As we increase the number of pairs, we gain more confidence in the pair distribution, which improves model performance. However, this also increases the training time. Although the performance suggests that using a larger number of pairs can improve model performance, the F1-Score exhibits marginal improvements beyond 2048 pairs. For this analysis, we adopted 2048 pairs per epoch in the training step to balance performance and training efficiency.

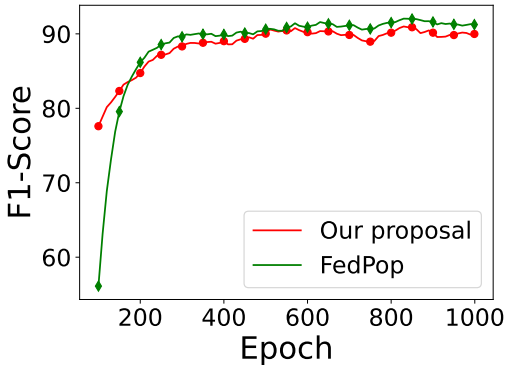
Figure 5.3b presents the effect of the α parameter, which controls the influence of KL divergence regularization between the local and global variational parameters (Eq. (5.2)). We also evaluate our approach with a different number of similar/dissimilar markers. As expected, our experiment indicates that the F1-Score is sensitive to α. For lower values of alpha ($0.1 \leq \alpha \leq 10$), our model remains stable. Furthermore, the number of markers affects our model's results, with more markers capturing the similarity structure more effectively. Therefore, our model achieves optimal performance with $\alpha = 100$ and three similar/dissimilar markers (six markers in total).



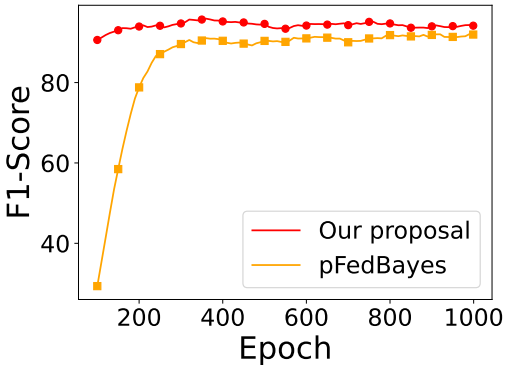
(a) Cifar10



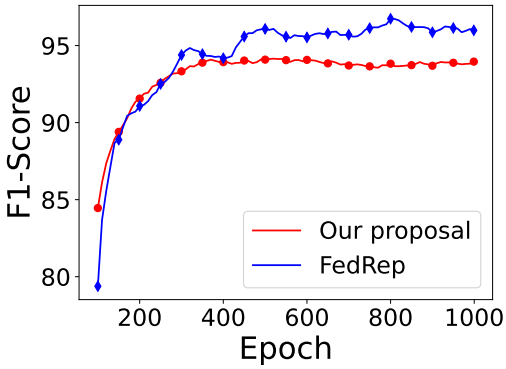
(b) FMNIST



(c) MaleViz



(d) Malimg



(e) MNIST

Figure 5.4: F1-Score results of our proposed method against the best approaches from the literature.

5.5.2 Quantitative Results

We compared our proposal to six PFL techniques from the literature [Arivazhagan et al., 2019, Li et al., 2021d, Zhang et al., 2022b, Kotelevskii et al., 2022, Collins et al., 2021, Tan et al., 2023b], as well as the standard FedAvg (Global Model – GM) and Local

Table 5.1: F1-Score of various PFL approaches across five datasets with ***100*** clients. The best results for each dataset are highlighted in **bold**, and the second-best results are underlined.

Proposal	Datasets									
	MNIST		FMNIST		MaleViz		Malimg		CIFAR10	
	#S = 4	#S = 5	#S = 4	#S = 5	#S = 4	#S = 5	#S = 4	#S = 5	#S = 4	#S = 5
Local	0.9113	0.9025	0.8238	0.8047	0.8562	0.8254	0.8191	0.8086	0.4048	0.3768
FedAvg. (GM)	0.8275	0.8934	0.7026	0.7338	0.7009	0.7182	0.8497	0.8509	0.4224	0.4459
FedRep	**0.9598**	0.9627	0.8351	**0.8616**	0.8439	0.8318	0.8874	0.9250	0.6891	0.7007
FedPer	0.9553	**0.9664**	0.8296	0.8458	0.8988	0.9015	0.8974	0.9112	0.6599	0.6726
FedPop	0.9180	0.9302	0.7734	0.8013	**0.9124**	0.9243	0.9033	0.9191	0.6211	0.6657
pFedSim	0.8973	0.9406	0.7270	0.7670	0.9093	0.9135	0.8899	0.9017	0.6643	0.6975
pFedBayes	0.8864	0.9143	0.8327	0.8452	0.8771	0.9025	0.9194	0.9236	0.6927	0.7098
DITTO	0.9041	0.9392	0.8035	0.8243	0.8498	0.8826	0.8978	0.9128	0.6247	0.6536
Our proposal	0.9386	0.9525	**0.8476**	0.8564	0.9005	**0.9324**	**0.9275**	**0.9369**	**0.7015**	**0.7202**

model without client communication (baseline). The results, summarized in Table 5.1, show that our approach outperformed FedAvg across all five datasets. For example, on the CIFAR10 dataset, our method achieved an F1-score improvement of 27.43 % with four shards per client.

Our method also shows improvements across other datasets. For instance, on CIFAR10, our approach achieved the highest F1 scores of 0.7015 and 0.7202 with four and five shards per client, respectively. On the Malimg dataset, it scored 0.9324 with five shards per client, surpassing FedRep, which achieved the second-best result of 0.9250. However, on the MNIST dataset, our proposal achieved the third-best results for both shard values.

Figure 5.4 compares the F1-scores between our proposal and the best methods from the literature across five datasets with four shards per client over 1000 FL epochs. In summary, our approach achieved the best performance in six out of ten FL settings and the second-best in two additional settings. These findings are consistent with results obtained using 200 clients, as detailed in Table 5.2.

5.5.3 Markers analysis

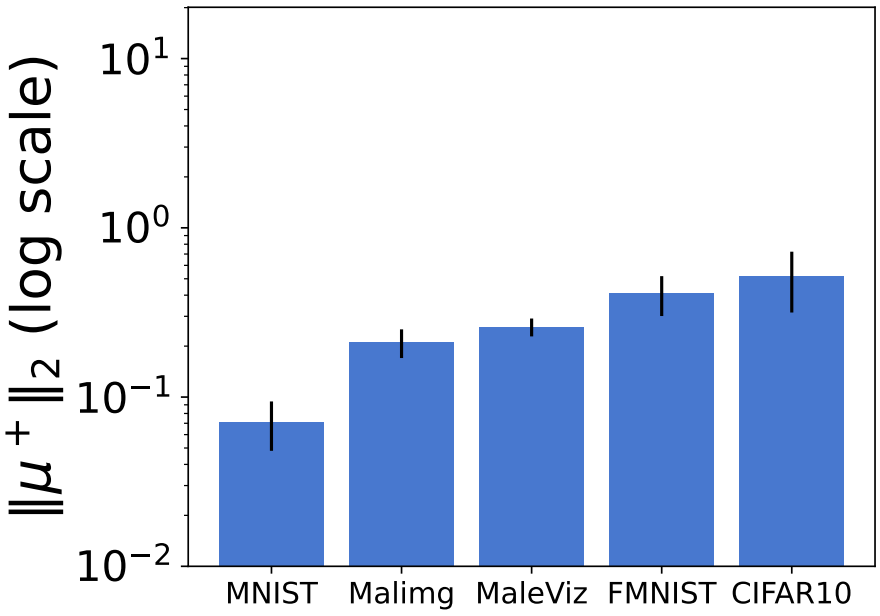
This section analyzes the norms of the positive and negative expected markers, denoted as $\|\mu^+\|_2$ and $\|\mu^-\|_2$, respectively, across five datasets. Figure 5.5 shows the positive expected markers norm $\|\mu^+\|_2$ (Figure 5.5a) and the negative expected markers norm $\|\mu^-\|_2$ (Figure 5.5b). For simplicity, each dataset utilizes one positive marker and one negative marker.

Table 5.2: F1-Score of various PFL approaches across five datasets with ***200*** clients. The best results for each dataset are highlighted in **bold**, and the second-best results are underlined.

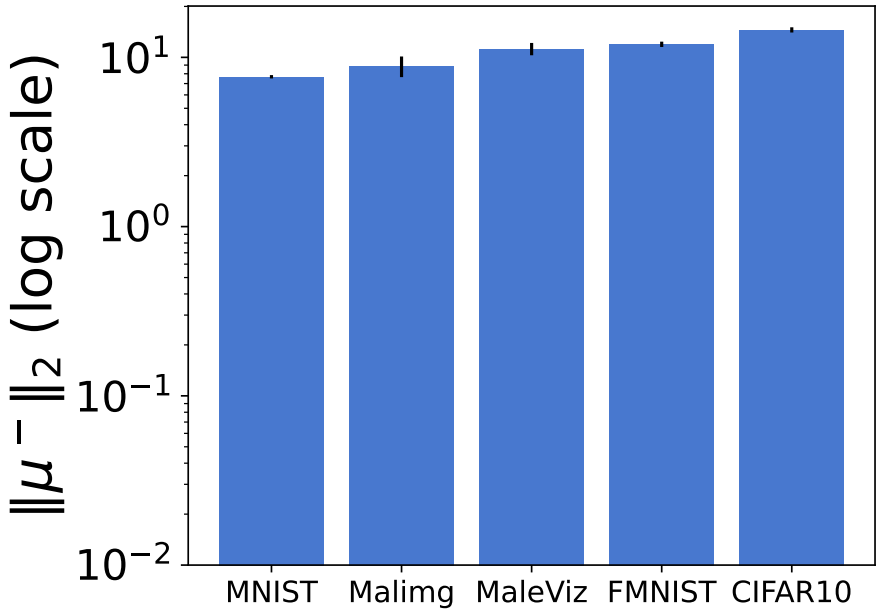
Proposal	Datasets									
	MNIST		FMNIST		MaleViz		Malimg		CIFAR10	
	$\#S=4$	$\#S=5$	$\#S=4$	$\#S=5$	$\#S=4$	$\#S=5$	$\#S=4$	$\#S=5$	$\#S=4$	$\#S=5$
Local	0.8983	0.8821	0.7911	0.7803	0.8154	0.7714	0.8050	0.7807	0.3840	0.3615
FedAvg. (GM)	0.8131	0.8425	0.6951	0.7166	0.6682	0.6790	0.8069	0.8307	0.4275	0.4393
FedRep	0.9456	0.9601	0.8090	0.8271	0.7918	0.8112	0.8507	0.8638	0.5588	0.6213
FedPer	**0.9461**	**0.9692**	0.8039	0.8197	0.8315	0.8686	0.8816	0.9181	0.6301	0.6601
FedPop	0.9072	0.9323	0.7593	0.7824	**0.8624**	0.8733	0.8677	0.8993	0.5839	0.6420
pFedSim	0.8901	0.9115	0.7168	0.7492	0.8333	0.8537	0.8575	0.8947	0.6290	0.6437
pFedBayes	0.8715	0.9092	0.8195	0.8385	0.8253	0.8496	**0.9036**	0.9121	0.6496	0.6678
DITTO	0.8893	0.9288	0.7941	0.8065	0.7926	0.8341	0.8689	0.8767	0.6133	0.6366
Our proposal	0.9254	0.9448	**0.8247**	**0.8513**	0.8570	**0.8777**	0.9013	**0.9206**	**0.6561**	**0.6824**

Definition 5.1 highlights a key feature of our algorithm: positive expected markers $\|\mu^+\|_2$ have a smaller norm than their negative marker $\|\mu^-\|_2$. Our loss function, aimed at clustering samples into groups with similar characteristics ($\|s_{ij}\|_2 \sim \delta(0)$), induces the model to align a positive marker $\mu^+ \in \mathcal{M}^+$ to the origin ($\|\mu^+\|_2 \sim \delta(0)$), as confirmed by our observations in Figure 5.5a. For example, in the MNIST dataset, we observe $\|\mu^+\|_2 = 0.071$ in contrast to $\|\mu^-\|_2 = 7.68$ with an expected ratio ($\|\mu^+\|_2/\|\mu^-\|_2$) equal to 0.92 %. We observed similar behavior for the Malimg and Maleviz datasets, with expected ratios of 2.37 % and 2.54 %, respectively.

Additionally, we found a relation between positive expected markers norm $\|\mu^+\|_2$ and F1-Score. As seen in Table 5.1, a lower $\|\mu^+\|_2$ value correlates with a higher F1-Score. For instance, our model achieves F1-Scores of 0.9525, 0.8564, and 0.7202 corresponding to $\|\mu^+\|_2$ values of 0.071, 0.401, and 0.519 for the MNIST, FMNIST, and CIFAR10 datasets, respectively. Therefore, this experiment found evidence that the positive norm position can be a proxy for the model's performance.



(a) Positive Markers Norm.



(b) Negative Markers Norm

Figure 5.5: Comparison of Positive and Negative Marker Norms across five datasets.

5.6 Related Work

PFL represents a significant approach to the development of FL in order to address the heterogeneity of data between diverse clients [Tan et al., 2023a]. This evolution has necessitated novel methodologies that aim to personalize models for individual clients, ensuring the robustness and efficiency of these models while accommodating the unique distributions of client data.

Global Model Personalization, a key strategy in PFL, involves training a universal FL model and subsequently customizing it for each client via local adaptation [Khodak et al., 2019, Yao and Sun, 2020]. This process, which directly confronts the issues of client drift and data diversity, depends on the global model's ability to generalize across varied datasets for successful personalization.

Loss regularization has emerged as a key technique in global model personalization, enhancing model stability and generalization [Li et al., 2020a]. It aids in achieving better convergence and facilitating personalized model creation by reducing the impact of disparate local updates. FedProx [Li et al., 2020b], for instance, incorporates a proximal term to reduce the discrepancy between global and local models, aiming to improve model integration through loss regularization. Similarly, SCAFFOLD [Karimireddy et al., 2020] employs variance reduction to refine gradient estimation and mitigate client drift, further enhanced by FedDyn [Durmus et al., 2021] extensions to these variance reduction techniques.

Considering federated meta-learning, novel approaches have been explored to enhance the adaptability of the global model and learning efficiency across varied tasks [Finn et al., 2017, Nichol et al., 2018]. Per-FedAvg [Fallah et al., 2020] integrates Model-Agnostic Meta-Learning (MAML) [Finn et al., 2017] with FL to enhance personalization across diverse client data, modifying the FedAvg algorithm to optimize a global model for quick personalization of individual clients' data with minimal extra training. Per-FedAvg has been extended in pFedMe [Dinh et al., 2020] to present a distinctive approach using Moreau envelopes and an l_2-norm regularization loss. This method balances personalization and generalization, offering a tailored learning paradigm that adjusts to these two objectives' trade-offs.

The Learning Personalized Models approach in PFL emphasizes customizing distinct models for each client, diverging from traditional methods that adapt a global model [Gupta and Raskar, 2018]. FedPer [Arivazhagan et al., 2019] proposed splitting the model into base and personalized layers. The clients privately maintain the deep personalized layers for localized training to acquire task-specific personalized representations. In contrast, the base layers are shared with the FL server, facilitating the learning of universal low-level features across clients. Similarly, FedRep [Collins et al., 2021] emphasizes

optimizing both local and global representations within the learning process to enhance learning outcomes.

Similarity-based methods in personalized federated learning leverage client data similarities to enhance model personalization [Smith et al., 2017, Huang et al., 2021]. By identifying and modeling the relationships between clients, these approaches enable the creation of tailored models for individual clients, with related clients learning similar models [Sattler et al., 2020]. The pFedSim [Tan et al., 2023b] focuses on client similarity for model personalization. It operates in two phases: a foundational generalization phase using FedAvg and a subsequent personalization phase that refines models based on client similarities.

Bayesian Personalized Federated Learning (BPFL) enhances traditional personalized FL by embedding Bayesian methods, introducing uncertainty quantification alongside model personalization [Corinzia et al., 2019]. This approach not only fine-tunes models to individual user data but also provides a measure of confidence in predictions, improving the robustness and transparency of FL systems.

Notable contributions include FedPop [Kotelevskii et al., 2022], which utilizes mixed-effects modeling and parallel MCMC to boost personalization and uncertainty quantification. pFedBayes [Zhang et al., 2022b] employs Bayesian variational inference to mitigate overfitting and refine personalization, especially in contexts of data diversity among clients. Building on pFedBayes, authors in Chen et al. [2023] and Liu et al. [2023a] introduce similar personalization techniques, such as personalization layers and amortized Bayesian inference, to adapt models to individual client data further.

Additionally, authors in Zhu et al. [2023] propose a Bayesian FL framework that leverages confidence values to manage non-identically distributed and variably sized datasets across clients, optimizing the aggregation process by accounting for uncertainty and model deviation.

This chapter advances PFL by introducing a Bayesian Neural Network framework with an auxiliary representation to enhance personalization. Unlike traditional methods that focus on variance reduction or meta-learning, our approach enhances latent feature representation, enabling deeper personalization for the diverse data distributions of individual clients. We propose an aggregation method designed explicitly for PFL, which aligns with Bayesian principles and provides theoretical guarantees. This strategy is designed to balance personalization and generalization across varied client datasets, tackling FL's challenge of data heterogeneity.

5.7 Final Remarks

This chapter presented a VI Personalized Federated Learning framework that combines variational inference with an auxiliary similarity space to improve personalization, uncertainty quantification, and robustness in heterogeneous federated settings. We provide theoretical guarantees, showing lower upper bounds on the KL divergence between local and global models compared to standard VI FL methods. Overall, this chapter advances personalized FL by improving theoretical bounds for Bayesian aggregation. Empirical results across five benchmark datasets demonstrated that our approach consistently outperforms some PFL methods in the literature. Finally, the analysis of marker norms revealed their potential as indicators of model quality.

Chapter 6

Rethinking variational inference in similarity spaces to personalized federated learning under malicious scenarios

Federated Learning shows challenges due to data heterogeneity, where each client may have a different data distribution [Zhang et al., 2022b]. To mitigate this issue, PFL modifies the global model to align with the unique data characteristics of individual clients. In these scenarios, distinguishing between legitimate outlier updates (resulting from data heterogeneity) and malicious updates (introduced by adversarial clients) can be a challenging task [Li et al., 2021d]. In this chapter, we extend our framework (Chapter 5) by rethinking variational inference. This uncertainty quantification enables us to derive theoretical results that help identify malicious updates. Therefore, by analyzing uncertainty in model updates, we can detect and mitigate the impact of malicious clients in the federated model.

Through theoretical analysis, we demonstrate the capability of our framework to achieve an optimal variational latent space, facilitating classification and robust predictions.

6.1 Our proposal

Probabilistic modeling provides a principled approach for capturing uncertainty in predictive models by defining a joint probability distribution $p_\theta(\boldsymbol{x}, \boldsymbol{m})$ over observed data $\boldsymbol{x}$ and latent variables (markers) $\boldsymbol{m}$. Bayesian inference involves computing the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{m}|\boldsymbol{x})$, which incorporates prior knowledge and observational data to characterize uncertainty in model parameters [Ranganath et al., 2014]. However,

direct computation of posterior distributions is typically intractable [Immer et al., 2021]. Variational inference (VI) addresses this challenge by approximating the true posterior with a parameterized variational distribution $q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})$, chosen from a tractable family.

The variational inference [Hotti et al., 2024] objective reduces to maximizing the Evidence Lower Bound (ELBO) defined as

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{m})}{q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})}\right], \tag{6.1}$$

where $\log p_{\boldsymbol{\theta}}(\boldsymbol{m},\boldsymbol{z})$ is the (unnormalized log-density) target with NNs parameters $\boldsymbol{\theta}$.

Some papers in the literature generalize traditional VI employing stochastic gradient estimates, enabling its application to complex models [Ranganath et al., 2014, Domke, 2020, Domke et al., 2023, Hotti et al., 2024]. Based on Eq. 6.1, we can apply the reparameterization trick, allowing unbiased stochastic gradient estimates of the ELBO for NN as

$$\mathcal{L}(\boldsymbol{\phi}) = \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x},\boldsymbol{m})]}_{\ell(\boldsymbol{\phi})} \quad \underbrace{-\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})}[\log q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})]}_{h(\boldsymbol{\phi})}, \tag{6.2}$$

where $\ell(\boldsymbol{\phi})$ represents the energy component and the second term is the entropy.

In VI, we iteratively optimize $\mathcal{L}(\boldsymbol{\phi})$ through $\boldsymbol{\phi}^{t+1} = \boldsymbol{\phi}^t + \eta\nabla\mathcal{L}(\boldsymbol{\phi}^t)$, where η is the step size, the gradient $\nabla\mathcal{L}$ is calculated with respect to variational parameters $\boldsymbol{\phi}^t$ at the time step t. In this chapter, we assume for the marker $\boldsymbol{m}$ that $q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})$, with parameters $\boldsymbol{\phi} = (\boldsymbol{\mu}, \boldsymbol{L})$ belongs to the Gaussian family with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^\intercal$. Besides, we assumed that each off-diagonal element of $\boldsymbol{L}$ is equal to zero (mean field parameterization) and exponential parameterization as in Section 5.2.2.

Formally, the reparameterization trick can be employed to obtain unbiased estimates of $\nabla\ell(\boldsymbol{\phi})$, where for a change of variable [Domke, 2020]

$$\begin{aligned}\nabla\ell(\boldsymbol{\phi}) &= \nabla\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{m}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x},\boldsymbol{m})] \\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla \log p_\theta(\boldsymbol{x}, t_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))],\end{aligned} \tag{6.3}$$

where we reparameterization mapping is define as

$$t_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}) = \boldsymbol{L}\boldsymbol{\epsilon} + \boldsymbol{\mu}, \tag{6.4}$$

with random vector $\boldsymbol{\epsilon}$ defined as

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_d) \in \mathbb{R}^d. \tag{6.5}$$

The components ϵ_n are assumed to be standard normal random variables, i.e. $\epsilon_n \sim \mathcal{N}(0,1)$, $\mathbb{E}[\epsilon_n] = 0$ and $\text{Var}[\epsilon_n] = 1$. Moreover, for a standard normal random variable, it holds that $\mathbb{E}[\epsilon^4] = 3$, which corresponds to the fourth moment (kurtosis) and will be used in **Lemma 6.1**. Finally, the expectation in Eq. 6.3 is based on the

distribution $p(\epsilon)$, and we use Monte Carlo integration to obtain unbiased estimates from $\nabla\ell(\boldsymbol{\phi})$.

Hotti et al. [2024] proposed the following **Lemma 6.1**, which establishes a bound on the gradient estimator produced by the exponential reparameterization trick.

Lemma 6.1. *(Ref. Hotti et al. [2024]) Let $\nabla g(\boldsymbol{\phi})$ be the gradient estimator of $\nabla\ell(\boldsymbol{\phi}) = \mathbb{E}[\nabla g(\boldsymbol{\phi})]$, $\boldsymbol{\phi} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the variational parameters, and assume that*

(1) each $\sigma_j \leq K_\star < \infty$,

(2) that $\bar{\boldsymbol{m}}$ is a stationary point (or maximum) of $p(x, .)$ and define $\bar{\boldsymbol{w}} = (\bar{\boldsymbol{m}}, 0)$,

(3) that $\boldsymbol{L}$ has a mean field (exponential) parameterization (see Section 5.2.2),

(4) M is a constant and $\mathbb{E}[\epsilon^4] = b$ (Eq. 6.5),

then

$$\mathbb{E}[\|\nabla g(\boldsymbol{\phi})\|_2^2] \leq M^2((K_\star^2 2\sqrt{db} + 1)\|\boldsymbol{\mu} - \bar{\boldsymbol{m}}\|_2^2 + (K_\star^2(\sqrt{db} + \sqrt{db}) + 1)\|\boldsymbol{L}\|_F^2),$$

when $\|.\|_F$ is frobenius norm.

Based on Eq. 6.2, we can therefore define a useful consequence of **Lemma 6.1**, which will be used in our subsequent theoretical analysis. The following corollary explicitly states a bound on the reparameterization gradient estimator, and this result provides a key ingredient for the stability and convergence guaranties that we analyze in Section 6.2.

Corollary 6.1. *Let $\nabla g(\boldsymbol{\phi})$ be the gradient estimator of $\nabla\ell(\boldsymbol{\phi}) = \mathbb{E}[\nabla g(\boldsymbol{\phi})]$, and assumptions in **Lemma 6.1** hold. If ϵ_i samples (Eq. 6.5) from PDF of $\mathcal{N}(0, 1)$, then*

$$\mathbb{E}[\|\nabla g(\boldsymbol{\phi})\|_2^2] \leq M^2(6K_\star^2\sqrt{d} + 1)\|\boldsymbol{\phi} - \bar{\boldsymbol{w}}\|_2^2,$$

where $\boldsymbol{\mu}$ and $\boldsymbol{L}$ are the variational parameter (Eq. 6.4); and $\|\boldsymbol{\phi} - \bar{\boldsymbol{w}}\|_2^2 = \|\boldsymbol{\mu} - \bar{\boldsymbol{m}}\|_2^2 + \|\boldsymbol{L}\|_F^2$ ($\|\boldsymbol{.}\|_F$ is the Frobenius norm).

Proof. This result directly follows from **Lemma 6.1**, where we consider $\mathbb{E}[\epsilon^4] = 3$ (kurtosis) for the standard normal PDF $\mathcal{N}(0, 1)$. Given that $d \geq 1$, we have the inequality

$$6K_\star^2\sqrt{d} + 1 > K_\star^2(\sqrt{3d} + 3\sqrt{d}) + 1 > K_\star^2 2\sqrt{3d} + 1,$$

which finalized the proof. □

Based in Zhu et al. [2023], we define a confidence-aware server aggregation strategy that explicitly accounts for client-specific uncertainty when combining local model updates in FL. Consider $j \in \{1, \ldots, N\}$ indexing the clients, and let $\boldsymbol{m} = t_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}) \in \mathbb{R}^d$ denote a

global latent variable representing the server-side model parameters $\boldsymbol{\phi}$ (Eq. 6.3). Thus, we assume an isotropic Gaussian conditional prior[1]

$$p_\theta(\boldsymbol{m}_j \mid \boldsymbol{m}, \rho_j^2) = \mathcal{N}\big(\boldsymbol{m}_j \,\big|\, \boldsymbol{m}, \rho_j^2 \mathbf{I}\big), \tag{6.6}$$

where the variance ρ_j^2 encodes client-specific uncertainty around the global model.

To approximate the intractable posterior $p(\boldsymbol{m}_j \mid \mathbf{x}_j)$, we employ a Gaussian variational family $q_{\boldsymbol{\phi}_j}(\boldsymbol{m}_j) = \mathcal{N}(\boldsymbol{m}_j \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, and define the evidence lower bound (ELBO) as

$$\mathcal{L}(\boldsymbol{\phi}) \;\propto\; \sum_{j=1}^{N} \mathbb{E}_{q_{\boldsymbol{\phi}_j}(\boldsymbol{m}_j)} \big[\log p_\theta(\boldsymbol{m}_j \mid \boldsymbol{m}, \rho_j^2)\big] \,, \tag{6.7}$$

which we want to maximize the ELBO with respect to $\boldsymbol{m}$.

Intuitively, based **Corollary 6.1**, $\boldsymbol{m}$ are parameters that concentrate q_{ϕ_j} entirely at a stationary point of ℓ (Eq. 6.2). Thus, $\mathbb{E}[\nabla g(\boldsymbol{\phi})]$ is bounded in terms of how far the average point sampled from $q_{\boldsymbol{\phi}_j}$ to the stationary point $\bar{\boldsymbol{w}}$. Finally, ℓ need not be convex, there can be multiple stationary points, and in this case, the **Corollary 6.1** and this conclusion hold simultaneously for all of them.

Inspired by Zhu et al. [2023] and **Corollary 6.1**, we proposed a variational FL approach to allow clients to estimate the confidence values over their local training results as

$$\rho_j^2 = \frac{\|\boldsymbol{\mu} - \boldsymbol{w}_{\boldsymbol{server}}\|_2^2 + \|\boldsymbol{L}\|_F}{d},$$

where we define that the confidence score $\lambda_j = 1/\rho_j^2$ is inversely proportional to its uncertainty ρ_j. Our method allows each client to estimate confidence values for their local training results based on model uncertainty $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ ($\|\boldsymbol{\Sigma} = \rho^2 \boldsymbol{I}\|_F$) and model disagreement with the server ($\|\boldsymbol{\mu}_j - \boldsymbol{w}_{server}\|_2$). In the aggregation step, the server uses these confidence values to compute a weighted average of model parameters.

Thus, each client contributes its local mean $\boldsymbol{\mu}_j$ weighted by a confidence score λ_j that is inversely proportional to its uncertainty ρ_j^2 as

$$\mathbf{m} = \frac{\sum_{j=1}^{N} \lambda_j \, \boldsymbol{\mu}_j}{\sum_{j=1}^{N} \lambda_j} \quad \text{with} \quad \lambda_j = \frac{1}{\rho_j^2}. \tag{6.8}$$

Finally, the proposed approach incorporates two key advantages, an uncertainty-aware weighting mechanism, where models with lower uncertainty have higher λ_j weights; and a mechanism that reduces the uncertainty weight for models that deviate significantly from the global model.

We define a server momentum to improve the quality of the aggregated stochastic gradient in each federated round. Let $j \in \{1, \ldots, N\}$ index clients and t denote the global

[1] A Gaussian conditional implies all clients' parameters are close to this (server) latent variable m, which is a reasonable assumption since all clients are running similar tasks (stationary)

(server) round. With aggregation weights $\{\lambda_j^t\}_{j=1}^N$ satisfying $\lambda_j^t \geq 0$ and $\sum_{j=1}^N \lambda_j^t = 1$, the server maintains an exponential moving average $\boldsymbol{g}^t$ of the aggregated gradient and updates the global model $\boldsymbol{\phi}$ as

$$\begin{aligned} \boldsymbol{\phi}^t &= \boldsymbol{\phi}^{t-1} + \gamma\, \boldsymbol{g}^t, \\ \boldsymbol{g}^t &= \beta\, \boldsymbol{g}^{t-1} + (1-\beta) \sum_{j=1}^N \lambda_j^t\, \nabla \widehat{\mathcal{L}}_j\big(\boldsymbol{\phi}^{t-1}\big)\,, \end{aligned} \tag{6.9}$$

where $\eta > 0$ is the server step size, $\beta \in [0, 1)$ is the momentum coefficient. Finally, setting $\beta = 0$ results in the vanilla FedAvg update.

Although $\boldsymbol{g}^t$ is a biased estimator of the true gradient, it is an exponential moving average of aggregated stochastic gradients. It therefore has a reduced variance compared to a single estimate $\nabla \widehat{\mathcal{L}}_j$. Using the direction

$$\beta\, \boldsymbol{g}^{t-1} + (1-\beta) \sum_{j=1}^N \lambda_j^t \nabla \widehat{\mathcal{L}}_j(\boldsymbol{\phi}^{t-1})$$

induces an anchoring effect, i.e., local updates are biased near the previous global descent direction. The literature shows evidence that this mitigates the client-drift phenomenon under data heterogeneity (variance reduction) [Sun et al., 2023]. Using the appropriate setting β, the momentum approach maintains the same convergence rate as FedAvg while removing the explicit heterogeneity assumption used in their analysis [Cheng et al., 2024].

6.2 Theoretical Results

> *The goal of this section is to clarify* why *our uncertainty mechanism is relevant and* what *can be guaranteed theoretically. In particular, we aim to (i) derive bounds for the reparameterized gradient estimator, showing how its variance grows with the distance to stationary points; (ii) use these bounds to formally motivate the confidence score λ_j (inversely proportional to uncertainty and disagreement with the server), thereby justifying the weighted aggregation proposal; and (iii) establish stability and convergence guarantees both in the centralized setting (Prox-SGD under variational inference) and in the federated setting, where we additionally control local drift and the effect of server-side momentum.*

In this section, we introduce further definitions and several technical lemmas that will be used in subsequent proofs. In particular, the following lemmas facilitate the

unrolling of recursions and the derivation of convergence rates in centralized and federated settings.

Definition 6.1. *(Smoothness) Let $f : \mathbb{R}^d \to \mathbb{R}$, and $P > 0$. We define that f(.) is **P-smooth** if it is differentiable and for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq P\|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

Definition 6.2. *(Convexity) We define that $f : \mathbb{R}^d \to \mathbb{R}$ is **convex** if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and for all $0 < t < 1$,*

$$f(t\boldsymbol{x} + (1-t)\boldsymbol{y}) \leq tf(\boldsymbol{x}) + (1-t)f(\boldsymbol{y}).$$

Definition 6.3. *(Strong convexity) Let $f : \mathbb{R}^d \to \mathbb{R}$, and $\mu > 0$. We say that f(.) is **μ-strongly convex** if, for every $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, and every $0 < t < 1$ we have that*

$$\mu\frac{t(1-t)}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + f(t\boldsymbol{x} + (1-t)\boldsymbol{y}) \leq tf(\boldsymbol{x}) + (1-t)f(\boldsymbol{y}).$$

Lemma 6.2. *(Ref. Garrigos and Gower [2023] – Lemma 2.14) If $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strong convex and differentiable function then for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$,*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2.$$

Lemma 6.3. *(Ref. Cover and Thomas [2005]) Let $\boldsymbol{w} \in \mathbb{R}^d$ be a random vector. Then, for any matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ and any vector $\boldsymbol{c} \in \mathbb{R}^d$, the differential entropy function $h(.)$ of $\boldsymbol{w}$ satisfies the following properties*

- $h(\boldsymbol{w} + \boldsymbol{c}) = h(\boldsymbol{w})$,
- $h(\boldsymbol{A}\boldsymbol{w}) = h(\boldsymbol{w}) + \log|\det(\boldsymbol{A})|$.

Lemma 6.4. *(Relaxed triangle inequality) Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t\}$ be t vectors in $\mathbb{R}^d$. Then the following are true*

- $\|\boldsymbol{v}_x + \boldsymbol{v}_y\|_2^2 \leq (1+a)\|\boldsymbol{v}_x\|_2^2 + \left(1 + \frac{1}{a}\right)\|\boldsymbol{v}_y\|_2^2, \forall a > 0$,
- $\|\sum_{i=1}^t \boldsymbol{v}_i\|_2^2 \leq t\sum_{i=1}^t \|\boldsymbol{v}_i\|_2^2$.

Definition 6.4. *(Prox-SGD) Let $\boldsymbol{\phi}^0$ be a fixed initial parameter, and let γ be the step size. The stochastic proximal gradient (Prox-SGD) method is given by*

$$\boldsymbol{\phi}^{t+1} = prox_{\gamma h}(\boldsymbol{\phi}^t - \gamma\nabla g(\boldsymbol{\phi}^t)),$$

where $\nabla g(\boldsymbol{\phi}^t)$ is a gradient estimator for $\nabla\ell(\boldsymbol{\phi}^t)$, and the proximal operator is defined as

$$prox_{\gamma h}(\boldsymbol{\phi}) = \arg\min_{\boldsymbol{v}} h(\boldsymbol{v}) + \frac{1}{2\gamma}\|\boldsymbol{\phi} - \boldsymbol{v}\|_2^2.$$

Lemma 6.5. *(Ref. [Garrigos and Gower, 2023] – Theorem 8.17 and 8.18) Let $\gamma > 0$ and $\boldsymbol{\phi}^*$ a solution for VI problem $\boldsymbol{\phi}^* = \arg\min_\phi \mathcal{L}(\boldsymbol{\phi})$ (Eq. 6.2). Assume $h(\boldsymbol{\phi})$ is convex and $\ell(\boldsymbol{\phi})$ is continuously differentiable at $\boldsymbol{\phi}^*$, then*

- *for all $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$, $\|prox_{\gamma h}(\boldsymbol{\phi}_1) - prox_{\gamma h}(\boldsymbol{\phi}_2)\|_2 \leq \|\boldsymbol{\phi}_1 - \boldsymbol{\phi}_2\|_2$,*
- *$\boldsymbol{\phi}^* = prox_{\gamma h}(\boldsymbol{\phi}^* - \gamma \nabla \ell(\boldsymbol{\phi}^*))$.*

Lemma 6.6. *(Ref. [Gower et al., 2019] – Theorem 3.2) Suppose we are given a sequence of $\boldsymbol{\phi}^t$ iterates such that for a step size $\gamma_t \leq \frac{1}{2L_c}$ and given constants $\mu > 0$ and $\sigma > 0$. Let*

$$\mathbb{E}[\|\boldsymbol{\phi}^{t+1} - \boldsymbol{\phi}^*\|^2] \leq (1 - \gamma_t \mu)\mathbb{E}[\|\boldsymbol{\phi}^t - \boldsymbol{\phi}^*\|_2^2] + 2\sigma^2\gamma_t^2.$$

By switching to a decaying stepsize according to

$$\gamma_t = \begin{cases} \frac{1}{2L_c} & \text{for } t < t^* \\ \frac{1}{\mu} \cdot \frac{2t+1}{(t+1)^2} & \text{for } t \geq t^* \end{cases}$$

where we define $t^* = 4\frac{L_c}{\mu} = 4C$. If $t \geq t^*$, then iteration satisfies

$$\mathbb{E}\left[\left\|\boldsymbol{\phi}^{t+1} - \boldsymbol{\phi}^*\right\|_2^2\right] \leq \frac{16C^2}{(t+1)^2}\left\|\boldsymbol{\phi}^0 - \boldsymbol{\phi}^*\right\|_2^2 + \frac{8\sigma^2}{\mu^2(t+1)}.$$

Constants $\mathcal{L}$ and C

- The constant L_c corresponds to the *expected smoothness constant* in Gower et al. [2019, Theorem 3.2], which satisfies the expected smoothness property

$$\mathbb{E}_v\left[\|\nabla \mathcal{L}_v(\boldsymbol{\phi}) - \nabla \mathcal{L}_v(\boldsymbol{\phi}^*)\|^2\right] \leq 2L_c\left(\mathcal{L}(\boldsymbol{\phi}) - \mathcal{L}(\boldsymbol{\phi}^*)\right).$$

- The constant C is defined as the ratio between the smoothness and strong convexity parameters,

$$C = \frac{L_c}{\mu},$$

and thus $t^* = 4C = 4\frac{L_c}{\mu}$. These constants appear in the stepsize switching rule and convergence rate given by Gower et al. [2019, Theorem 3.2].

How to read the remainder of this section.

The definitions above specify the regularity conditions (smoothness/strong convexity) under which we can obtain quantitative rates, while Lemmas 6.4–6.6 provide the technical tools used repeatedly to control cross terms, unroll recursions, and map one-step inequalities into iteration complexity bounds. In the next subsections, we first establish a centralized analysis for Prox-SGD under variational inference, showing how the stochasticity induced by the reparameterized estimator can be controlled in terms of the distance to a stationary point. We then extend the argument to the federated setting, where additional error terms arise from local drift and server-side momentum, and we show how these effects can be bounded under standard assumptions on smoothness and gradient noise.

6.2.1 Our (Centralized) Theoretical results

Lemma 6.7 shows that, under the exponential parameterization $\boldsymbol{\sigma} = \exp(\boldsymbol{p})$, the entropy regularizer contributes a gradient with *constant squared norm* equal to d. This is convenient for the analysis: the entropy term does not introduce an uncontrolled growth in the stochastic gradients.

Lemma 6.7. *Let $h(\boldsymbol{\phi})$ be the differential entropy of a random variable from the location-scale family defined as $\boldsymbol{\sigma} = \exp(\boldsymbol{p})$ (Section 5.2.2), then $\|\nabla h(\boldsymbol{\phi})\|_2^2 = d$.*

Proof. We use Lemma 6.3 as

$$\begin{aligned} h(\boldsymbol{\phi}) &= h(\epsilon \boldsymbol{L} + \boldsymbol{\mu}) \\ &= h(\epsilon) + \log(|\det(\boldsymbol{L})|) \\ &= h(\epsilon) + \sum_{i=1}^{d} \log(\exp(p_i)), \end{aligned}$$

and thus

$$\frac{\partial h(\boldsymbol{\phi})}{\partial p_i} = \frac{1}{\exp(p_i)} \exp(p_i).$$

Therefore, we conclude

$$\|\nabla h(\boldsymbol{\phi})\|_2^2 = \left(\sqrt{\sum_{i=1}^{d} \frac{1}{\exp(p_i)} \exp(p_i)} \right)^2 = d.$$

□

Proposition 6.1 makes explicit that the reparameterized gradient estimator is quadratically controlled by the distance to an optimal variational solution $\boldsymbol{\phi}^*$ (up to a constant term depending on $\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|$). In particular, as the iterates move closer to $\boldsymbol{\phi}^*$, the bound tightens, indicating that uncertainty in the stochastic gradients is smaller near stationary points. This relationship is one of the core motivations for using uncertainty/disagreement as a confidence signal in aggregation: clients whose local parameters drift far from the server model are expected to exhibit larger gradient variability.

Proposition 6.1. *Let $\nabla g(\boldsymbol{\phi})$ be the gradient estimator of $\nabla \ell(\boldsymbol{\phi}) = \mathbb{E}[\nabla g(\boldsymbol{\phi})]$ and assumptions in **Lemma 6.1** hold. In particular, $\log p_\theta(\boldsymbol{x}, .)$ is M-smooth and has a maximum (or stationary point) at $\bar{\boldsymbol{m}}$, and define $\bar{\boldsymbol{w}} = (\bar{\boldsymbol{m}}, 0)$. Then, for all $\boldsymbol{\phi}$ and every solution $\boldsymbol{\phi}^*$ of the VI problem, the following bound holds*

$$\mathbb{E}\|\nabla g(\boldsymbol{\phi})\|_2^2 \leq 2M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi} - \boldsymbol{\phi}^*\|_2^2 + 2M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2.$$

Proof. From **Corollary 6.1**, we have for all $\boldsymbol{\phi}$ that

$$\begin{aligned}\mathbb{E}\|\nabla g(\boldsymbol{\phi})\|_2^2 &\leq M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi} - \bar{\boldsymbol{w}}\|_2^2 \\ &= M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi} - \boldsymbol{\phi}^* + \boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2.\end{aligned}$$

We now apply **Lemma 6.4** to obtain

$$\begin{aligned}\mathbb{E}\|\nabla g(\boldsymbol{\phi})\|_2^2 &\leq M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi} - \boldsymbol{\phi}^* + \boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2 \\ &\leq 2M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi} - \boldsymbol{\phi}^*\|_2^2 + 2M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2,\end{aligned}$$

where $2M^2(6K_\star^2\sqrt{d}+1)$ and $2M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2$ are constants. This concludes this proof. □

Theorem 6.2 proof a standard contraction recursion, the term $(1 - \eta\mu)$ contracts the error toward $\boldsymbol{\phi}^*$, while the additive term proportional to η^2 captures the irreducible effect of stochasticity (the mismatch $\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|$). Consequently, smaller stepsizes improve stability but also slow down progress.

Proposition 6.2. *Let ℓ be a P-smooth function and μ-strongly convex, and assumptions in Proposition 6.1 hold. Let $\{\boldsymbol{\phi}^t\}_{t\in N}$ be generated by the Prox-SGD algorithm, with a constant stepsize $\eta \in \left(0, \frac{\mu}{4M^2(6K_\star^2\sqrt{d}+1)}\right)$. Then*

$$\mathbb{E}\left[\|\boldsymbol{\phi}^{t+1} - \boldsymbol{\phi}^*\|_2^2\right] \leq (1-\eta\mu)\mathbb{E}\|\boldsymbol{\phi}^t - \boldsymbol{\phi}^*\|_2^2 + 2\eta^2\left[(2M^2 6K_\star^2\sqrt{d} + 2M^2 + P^2)\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2\right].$$

Proof. This proof follows ideas in Gower et al. [2019, Theorem 3.1] and Garrigos and Gower [2023, Theorem 12.9]. First, considering the non-expansiveness of the proximal

operator (**Lemma 6.5**), we write

$$\begin{aligned}\left\|\boldsymbol{\phi}^{t+1}-\boldsymbol{\phi}^{*}\right\|_2^2 &= \left\|\operatorname{prox}_{\gamma h}(\boldsymbol{\phi}^t-\eta\nabla g(\boldsymbol{\phi}^t))-\operatorname{prox}_{\gamma h}(\boldsymbol{\phi}^*-\eta\nabla\ell(\boldsymbol{\phi}^*))\right\|_2^2\\ &\leq \left\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^*+\eta\left(\nabla\ell(\boldsymbol{\phi}^*)-\nabla g(\boldsymbol{\phi}^t)\right)\right\|_2^2\\ &= \left\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^*\right\|_2^2+\underbrace{2\eta\left\langle\boldsymbol{\phi}^t-\boldsymbol{\phi}^*,\nabla\ell(\boldsymbol{\phi}^*)-\nabla g(\boldsymbol{\phi}^t)\right\rangle}_{(ii)}+\underbrace{\eta^2\left\|\nabla\ell(\boldsymbol{\phi}^*)-\nabla g(\boldsymbol{\phi}^t)\right\|_2^2}_{(i)}.\end{aligned} \tag{6.10}$$

To handle term (i), we apply **Lemma 6.4** as

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{\phi}^*)-\nabla g(\boldsymbol{\phi}^t)\right\|_2^2\right] &\leq 2\mathbb{E}\left[\|\nabla\ell(\boldsymbol{\phi}^*)\|_2^2\right]+2\mathbb{E}\left[\left\|\nabla g(\boldsymbol{\phi}^t)\right\|_2^2\right]\\ &\leq 2\left\|\nabla\ell(\boldsymbol{\phi}^*)\right\|_2^2+2\mathbb{E}\left[\left\|\nabla g(\boldsymbol{\phi}^t)\right\|_2^2\right]\\ &\overset{a}{\leq} 2\left\|\nabla\ell(\boldsymbol{\phi}^*)-\nabla\ell(\bar{\boldsymbol{w}})\right\|_2^2+2\mathbb{E}\left[\left\|\nabla g(\boldsymbol{\phi}^t)\right\|_2^2\right]\\ &\overset{b}{\leq} 2P^2\left\|\boldsymbol{\phi}^*-\bar{\boldsymbol{w}}\right\|_2^2+2\mathbb{E}\left[\left\|\nabla g(\boldsymbol{\phi}^t)\right\|_2^2\right],\end{aligned}$$

where (a) uses the fact that $\nabla\ell(\bar{\boldsymbol{w}})=0$, since $\bar{\boldsymbol{w}}$ is a stationary point of ℓ, and (b) follows from the P-smoothness of ℓ (see **Definition 6.1**).

For term (ii), we use the strong convexity of ℓ (**Lemma 6.2**) as

$$\begin{aligned}\mathbb{E}\left[\left\langle\boldsymbol{\phi}^t-\boldsymbol{\phi}^*,\nabla\ell(\boldsymbol{\phi}^*)-\nabla g(\boldsymbol{\phi}^t)\right\rangle\right] &= -\left\langle\boldsymbol{\phi}^*-\boldsymbol{\phi}^t,\nabla\ell(\boldsymbol{\phi}^*)-\nabla\ell(\boldsymbol{\phi}^t)\right\rangle\\ &\leq -\mu\left\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^*\right\|_2^2,\end{aligned}$$

Substituting (i) and (ii) into Eq. (6.10) and taking the total expectation, we obtain

$$\mathbb{E}\left[\|\boldsymbol{\phi}^{t+1}-\boldsymbol{\phi}^*\|_2^2\right]\leq\mathbb{E}\left[\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^*\|_2^2\right]+2\eta^2P^2\|\boldsymbol{\phi}^*-\bar{\boldsymbol{w}}\|_2^2+2\gamma^2\mathbb{E}\left[\left\|\nabla g(\boldsymbol{\phi}^t)\right\|_2^2\right]-2\eta\mu\mathbb{E}\left[\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^*\|_2^2\right]. \tag{6.11}$$

Applying **Proposition 6.1** to bound Eq. 6.11, we get

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{\phi}^{t+1}-\boldsymbol{\phi}^*\|_2^2\right]\leq\left(1-2\eta\mu+4\eta^2M^2(6K_\star^2\sqrt{d}+1)\right)\mathbb{E}\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^*\|_2^2+\\ +2\eta^2(2M^26K_\star^2\sqrt{d}+2M^2+P^2)\|\boldsymbol{\phi}^*-\bar{\boldsymbol{w}}\|_2^2).\end{aligned} \tag{6.12}$$

We now verify the condition on η to ensure that the contraction term satisfies

$$\begin{aligned}1-2\eta\mu+4\eta^2M^2(6K_\star^2\sqrt{d}+1)&\leq 1-\eta\mu\\ 2\eta\mu-4\eta^2M^2(6K_\star^2\sqrt{d}+1)&\geq\eta\mu\\ 2\mu-4\eta M^2(6K_\star^2\sqrt{d}+1)&\geq\mu\\ 2\mu-\mu&\geq 4\eta M^2(6K_\star^2\sqrt{d}+1)\\ \mu&\geq 4\eta M^2(6K_\star^2\sqrt{d}+1)\\ \eta&\leq\frac{\mu}{4M^2(6K_\star^2\sqrt{d}+1)}.\end{aligned}$$

Under this condition, we obtain the final inequality

$$\mathbb{E}\left[\|\boldsymbol{\phi}^{t+1}-\boldsymbol{\phi}^*\|_2^2\right] \leq (1-\eta\mu)\mathbb{E}\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^*\|_2^2 + 2\eta^2(2M^2 6K_\star^2\sqrt{d}+2M^2+P^2)\|\boldsymbol{\phi}^*-\bar{\boldsymbol{w}}\|_2^2,$$

which is guaranteed by the choice of $\eta \leq \frac{\mu}{4M^2(6K_\star^2\sqrt{d}+1)}$ and completes the proof.

□

Proposition 6.3. *Let ℓ be a P-smooth function and μ-strongly convex. Then, for any $\xi > 0$, the VI problem with proximal SGD, a variational family satisfying* ***Proposition 6.2*** *with the exponential parameterization, guarantees $\mathbb{E}\|\boldsymbol{\phi}^T-\boldsymbol{\phi}^*\|_2^2 \leq \xi$ if*

$$\gamma_t = \begin{cases} \frac{1}{2L_c} & \text{for } t < 4C \\ \frac{1}{\mu}\cdot\frac{2t+1}{(t+1)^2} & \text{for } t \geq 4C \end{cases} \quad \text{and}$$

$$T \geq \max\left(\frac{16\eta^2(12M^2K_\star^2\sqrt{d}+2M^2+P^2)\|\boldsymbol{\phi}^*-\bar{\boldsymbol{w}}\|_2^2}{\mu^2\xi} + \frac{4C\|\boldsymbol{\phi}^0-\boldsymbol{\phi}^*\|}{\sqrt{\xi}}, 4C\right).$$

Proof. We begin by applying **Lemma 6.6** to the result of **Proposition 6.2**,

$$\mathbb{E}\left[\left\|\boldsymbol{\phi}^{t+1}-\boldsymbol{\phi}^*\right\|_2^2\right] \leq \frac{16C^2}{(t+1)^2}\left\|\boldsymbol{\phi}^0-\boldsymbol{\phi}^*\right\|^2 + \frac{8\sigma^2}{\mu^2(t+1)},$$

where the variance term $\sigma^2 = (12M^2K_\star^2\sqrt{d}+2M^2+P^2)\|\boldsymbol{\phi}^*-\bar{\boldsymbol{w}}\|_2^2$, and C is a constant.

Now, let $\xi > 0$ be an arbitrary error tolerance. We are interested in finding T that

$$\mathbb{E}\left[\left\|\boldsymbol{\phi}^{T}-\boldsymbol{\phi}^*\right\|_2^2\right] \leq \xi,$$

so it follows

$$\frac{16C^2}{T^2}\left\|\boldsymbol{\phi}^0-\boldsymbol{\phi}^*\right\|_2^2 + \frac{8\sigma^2}{\mu^2 T} \leq \xi.$$

Multiplying both sides of the inequality by T^2, we have the quadratic inequality

$$16C^2\left\|\boldsymbol{\phi}^0-\boldsymbol{\phi}^*\right\|_2^2 + \frac{8\sigma^2}{\mu^2}T \leq \xi T^2 \tag{6.13}$$

$$\xi T^2 - \frac{8\sigma^2}{\mu^2}T - 16C^2\left\|\boldsymbol{\phi}^0-\boldsymbol{\phi}^*\right\|_2^2 \geq 0. \tag{6.14}$$

Solving for the T that satisfies this inequality, we apply the quadratic formula

$$T \geq \frac{\frac{8\sigma^2}{\mu^2} + \sqrt{\left(\frac{8\sigma^2}{\mu^2}\right)^2 + 64\xi C^2\|\boldsymbol{\phi}^0-\boldsymbol{\phi}^*\|^2}}{2\xi}.$$

To simplify this expression, we use the concavity (and thus subadditivity) of the square root function ($\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$)

$$\begin{aligned}\frac{\frac{8\sigma^2}{\mu^2} + \sqrt{\left(\frac{8\sigma^2}{\mu^2}\right)^2 + 64\xi C^2\|\boldsymbol{\phi}^0 - \boldsymbol{\phi}^*\|_2^2}}{2\xi} &\leq \frac{\frac{8\sigma^2}{\mu^2} + \frac{8\sigma^2}{\mu^2} + \sqrt{64\xi C^2\|\boldsymbol{\phi}^0 - \boldsymbol{\phi}^*\|_2^2}}{2\xi} \\ &\leq \frac{\frac{16\sigma^2}{\mu^2} + 8C\|\boldsymbol{\phi}^0 - \boldsymbol{\phi}^*\|_2\sqrt{\xi}}{2\xi} \\ &\leq \frac{8\sigma^2}{\mu^2\xi} + \frac{4C\|\boldsymbol{\phi}^0 - \boldsymbol{\phi}^*\|_2}{\sqrt{\xi}}.\end{aligned}$$

Therefore, $\mathbb{E}\|\boldsymbol{\phi}^t - \boldsymbol{\phi}^*\|_2^2 \leq \xi$ can be satisfied with a number of iterations at least

$$T \geq \max\left(\frac{16\eta^2(12M^2K_\star^2\sqrt{d} + 2M^2 + P^2)\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2}{\mu^2\xi} + \frac{4C\|\boldsymbol{\phi}^0 - \boldsymbol{\phi}^*\|_2}{\sqrt{\xi}}, 4C\right),$$

where $\sigma^2 = (12M^2K_\star^2\sqrt{d} + 2M^2 + P^2)\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2$. □

The final bound provides an explicit iteration complexity to reach $\mathbb{E}\|\boldsymbol{\phi}^T - \boldsymbol{\phi}^*\|_2^2 \leq \xi$. It separates two effects: a fast-decaying term (driven by the deterministic contraction) and a slower term proportional to the effective variance σ^2, which depends on the reparameterization constants and on $\|\boldsymbol{\phi}^* - \bar{\boldsymbol{w}}\|_2^2$. Practically, this means that convergence is guaranteed under the stated regularity assumptions and appropriate stepsize control, and the rate degrades as the stochastic gradients become noisier or the optimum moves farther from the stationary reference $\bar{\boldsymbol{w}}$.

Previous work has shown similar convergence results for variational inference using linear reparameterization [Domke, 2020, Domke et al., 2023, Ko et al., 2024]. Although the linear assumptions approach simplifies the theoretical analysis, it can be limiting in real-world applications. In our chapter, we use an exponential reparameterization (non-negative variance estimation, Eq. 5.4) [Ganguly and Earp, 2021, Zhu et al., 2023, Barros et al., 2024c]. Building on these insights, our analysis extends the existing convergence results to this more practical setting.

6.2.2 Our (Federated) Theoretical results

Moving to federated optimization introduces two additional challenges out the centralized setting: (i) local drift, since clients perform multiple local steps before communi-

cating; and (ii) server-side momentum, which mixes gradient information across rounds and can amplify heterogeneity if not controlled.

Assumption 6.1. *(Lipschitz Gradient). The function $\mathcal{L}_i$ is L-smooth for all client i, i.e., $\|\nabla\mathcal{L}_i(\boldsymbol{\phi}_a) - \nabla\mathcal{L}_i(\boldsymbol{\phi}_b)\|_2 \leq L\|\boldsymbol{\phi}_a - \boldsymbol{\phi}_b\|_2$, $\forall\phi_a, \phi_b \in \mathbb{R}^d$.*

Assumption 6.2. *(Ref. Panchal et al. [2023b], Unbiasedness and Bounded Local Variance). For each client i and variational parameter $\boldsymbol{\phi}_i$, we assume the access to an unbiased stochastic gradient $\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi})$ of client's true gradient $\nabla\mathcal{L}_i(\boldsymbol{\phi})$, i.e., $\mathbb{E}[\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi})] = \nabla\mathcal{L}_i(\boldsymbol{\phi})$. The function $\widehat{\mathcal{L}}_i$ have S_ℓ-bounded (local) variance i.e., $\|\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi}) - \nabla\mathcal{L}_i(\boldsymbol{\phi})\|_2^2 \leq S_\ell^2$.*

Assumption 6.3. *(Ref. Panchal et al. [2023a], Bounded Global Variance). For each client i and variational parameter $\boldsymbol{\phi}_i$, we assume the access to an unbiased stochastic gradient $\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi})$ of client's true gradient $\nabla\mathcal{L}_i(\boldsymbol{\phi})$, i.e., $\mathbb{E}[\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi})] = \nabla\mathcal{L}_i(\boldsymbol{\phi})$. The function $\widehat{\mathcal{L}}$ have S_g-bounded (global) variance i.e., $\sum_{i=1}^N \lambda_i\mathbb{E}[\|\nabla\mathcal{L}_i(\boldsymbol{\phi}) - \nabla\widehat{\mathcal{L}}(\boldsymbol{\phi})\|_2^2] \leq S_g^2$, for $\sum_{i=1}^N \lambda_i = 1$.*

Assumption 6.4. *(Ref. Karimireddy et al. [2020], Bounded Gradient Dissimilarity) There exists constants such as $G \geq 0$ and $B \geq 1$ such that $\sum_{i=1}^N \lambda_i\|\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi})\|_2^2 \leq G^2 + B^2\|\nabla\widehat{\mathcal{L}}(\boldsymbol{\phi})\|_2^2$, for $\sum_{i=1}^N \lambda_i = 1$.*

Lemma 6.8. *(Upper bound of local drift). Let Assumptions in **Lemma 6.1** hold. For all client in $i \in \mathbb{N}$ with arbitrary local iteration steps k with local learning rate $\eta_i \leq \frac{1}{k}$, the local drift can be bounded as follows,*

$$\mathbb{E}[\|\boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq 36\eta_i(M^2(K_\star^2\sqrt{d}+1)\|\boldsymbol{\Delta}_i\|_2^2 + d),$$

where $\|\boldsymbol{\Delta}_i\|_2^2 = \|\boldsymbol{\phi}_i - \bar{\boldsymbol{w}}_i\|_2^2$.

Proof. We start with restating the SGD update

$$\boldsymbol{\phi}_i^{t,k} = \boldsymbol{\phi}_i^{t,k-1} - \eta_i\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi}_i^{t,k-1}),$$

which

$$\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi}_i^{t,k-1}) = \nabla g_i(\boldsymbol{\phi}_i^{t,k-1}) + \nabla h_i(\boldsymbol{\phi}_i^{t,k-1}).$$

Using the above update, we proceed as

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] &= \mathbb{E}[\|\boldsymbol{\phi}_i^{t,k-1} - \boldsymbol{\phi}^t - \eta_i\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi}_i^{t,k-1})\|_2^2] \\
&\overset{(1)}{\leq} \left(1 + \frac{1}{k-1}\right)\mathbb{E}[\|\boldsymbol{\phi}_i^{t,k-1} - \boldsymbol{\phi}^t\|_2^2] + k\eta_i^2(\mathbb{E}[\|\nabla\widehat{\mathcal{L}}_i(\boldsymbol{\phi}_i^{t,k})\|_2^2]) \\
&\overset{(2)}{\leq} \left(1 + \frac{1}{k-1}\right)\mathbb{E}[\|\boldsymbol{\phi}_i^{t,k-1} - \boldsymbol{\phi}^t\|_2^2] + 2k\eta_i^2(\mathbb{E}[\|\nabla g_i(\boldsymbol{\phi}_i^{t,k})\|_2^2] + \mathbb{E}[\|\nabla h_i(\boldsymbol{\phi}_i^{t,k})\|_2^2]) \\
&\overset{(3)}{\leq} \left(1 + \frac{1}{k-1}\right)\mathbb{E}[\|\boldsymbol{\phi}_i^{t,k-1} - \boldsymbol{\phi}^t\|_2^2] + 2k\eta_i^2(M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\Delta}_i\|_2^2 + d),
\end{aligned}$$

where inequality (1) and (2) follow from **Lemma 6.4**, and the last inequalities follow from **Corollaly 6.1** and **Lemma 6.7**.

Unrolling the recursion, we obtain

$$\mathbb{E}[\|\boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq \sum_{\tau=0}^{k-1} \left(1 + \frac{1}{k-1}\right)^{\tau} [2k\eta_i^2(M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\Delta}_i\|_2^2 + d)]$$
$$\leq (k-1)\left[\left(1 + \frac{1}{k-1}\right)^{k} - 1\right] [2k\eta_i^2(M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\Delta}_i\|_2^2 + d)].$$

As can see in Karimireddy et al. [2020, Lemma 14], the inequality

$$(k-1)\left[\left(1 + \frac{1}{k-1}\right)^{k} - 1\right] \leq 3k$$

for $k \geq 1$. Thus, we conclude that

$$\mathbb{E}[\|\boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq 6k\eta_i^2(M^2(6K_\star^2\sqrt{d}+1)\|\boldsymbol{\Delta}_i\|_2^2 + d)$$
$$\leq 36\eta_i(M^2(K_\star^2\sqrt{d}+1)\|\boldsymbol{\Delta}_i\|_2^2 + d),$$

where the last equation follow to $\eta_i \leq \frac{1}{k}$.

□

Lemma 6.8 shows that the deviation between a client's local iterate and the server iterate grows proportionally to the local stepsize and is amplified by two factors: (i) a term depending on the distance to the stationary reference (capturing model/gradient variability), and (ii) a dimension-dependent contribution coming from the entropy component. This bound formalizes the intuition that aggressive local updates (large η_i or many local steps) increase client–server mismatch, which later translates into larger disagreement and potentially higher uncertainty.

Lemma 6.9. *Suppose Assumptions 6.1–6.3 hold. For all clients $i \in \mathbb{N}$ with arbitrary local iteration steps k with local learning rate $\eta_i = \eta \leq \frac{1}{k}$, the local drift can be bounded as follows,*

$$\mathbb{E}[\|\sum_{i=1}^{N} \lambda_i \boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq 36\eta \left(M^2(K_\star^2\sqrt{d}+1)\|\overline{\boldsymbol{\Delta}}\|_2^2 + d\right).$$

Proof. Given the Jensen's inequality ($\|.\|_2^2$ is a convex function) and **Lemma 6.8** on the upper bound of local drift, we have

$$\mathbb{E}[\|\sum_{i=1}^{N} \lambda_i \boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq \sum_{i=1}^{N} \lambda_i \mathbb{E}[\|\boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq \sum_{i=1}^{N} 36\lambda_i\eta \left(M^2(K^3\sqrt{d}+1)\|\boldsymbol{\Delta}_i\|_2^2 + d\right).$$

Since η is the same for all clients, we can factor out the common terms

$$\mathbb{E}[\|\sum_{i=1}^{N} \lambda_i \boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq 36\eta \left(M^2(K_\star^2\sqrt{d}+1)\sum_{i=1}^{N}\lambda_i\|\boldsymbol{\Delta}_i\|_2^2 + d\right).$$

Let $\|\overline{\boldsymbol{\Delta}}\|_2^2 = \sum_{i=1}^N \lambda_i \|\boldsymbol{\Delta}_i\|_2^2$ represent the average squared norm over all clients. Then, we can simplify the expression to

$$\mathbb{E}[\|\sum_{i=1}^N \lambda_i \boldsymbol{\phi}_i^{t,k} - \boldsymbol{\phi}^t\|_2^2] \leq 36\eta \left(M^2(K_\star^2\sqrt{d}+1)\|\overline{\boldsymbol{\Delta}}\|_2^2 + d \right).$$

□

Theorem 2. *Let $\mathcal{L}(\boldsymbol{\phi})$ be the global objective and suppose Assumptions 6.1–6.3 hold. In communication round t, each client i performs k local SGD steps starting from $\boldsymbol{\phi}^t$ with stepsize $\eta_i = \eta \leq \min(\frac{1}{k}, \sqrt{\frac{1-3\beta^2}{9\beta^2 L^2}})$, producing $\boldsymbol{\phi}_i^{t,k}$, and the server forms $\boldsymbol{\phi}^{t+1} = \boldsymbol{\phi}^t - \gamma\, \boldsymbol{g}^t$, where*

$$\boldsymbol{g}^t = \beta \boldsymbol{g}^{t-1} + (1-\beta) \sum_{i=1}^N \lambda_i \nabla \widehat{\mathcal{L}}_i(\boldsymbol{\phi}_i^{t,k}).$$

Then

$$\sum_{i=1}^N \lambda_i \mathbb{E}\Big[\big\| \boldsymbol{g}^t - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t) \big\|_2^2 \Big] \leq 3S_g^2\big[(1+3L^2\eta^2)\beta^2 + (1-\beta)^2\big] + 9\beta^2 L^2 \eta^2 S_\ell^2 + \\ + 108\beta^2 L^2 \eta^2 \left[M^2\big(K_\star^2\sqrt{d}+1\big)\, \|\overline{\boldsymbol{\Delta}}\|_2^2 + \frac{d}{6} \right].$$

Proof. By definition of the momentum variable $\boldsymbol{g}^t = \beta \boldsymbol{g}^{t-1} + (1-\beta)\sum_{i=1}^N \lambda_i \nabla \widehat{\mathcal{L}}_i(\boldsymbol{\phi}^t)$, we

have

$$
\begin{aligned}
&\sum_{i=1}^{N} \lambda_i \mathbb{E}\left[\left\|\boldsymbol{g}^t - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right] \\
&= \sum_{i=1}^{N} \lambda_i \mathbb{E}\left[\left\|\beta \boldsymbol{g}^{t-1} + (1-\beta)\sum_{j=1}^{N} \lambda_j \nabla \widehat{\mathcal{L}}_j(\boldsymbol{\phi}^t) - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right] \\
&= \sum_{i=1}^{N} \lambda_i \mathbb{E}\left[\left\|\beta \boldsymbol{g}^{t-1} + (1-\beta)\sum_{j=1}^{N} \lambda_j \nabla \widehat{\mathcal{L}}_j(\boldsymbol{\phi}^t) - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t) \pm \beta \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1})\right\|_2^2\right] \\
&= \sum_{i=1}^{N} \lambda_i \mathbb{E}\left[\left\|\beta(\boldsymbol{g}^{t-1} - \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1})) + (1-\beta)\sum_{j=1}^{N} \lambda_j \nabla \widehat{\mathcal{L}}_j(\boldsymbol{\phi}^t) - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t) + \beta \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1})\right\|_2^2\right] \\
&= \sum_{i=1}^{N} \lambda_i \mathbb{E}\left[\left\|\beta(\boldsymbol{g}^{t-1} - \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1})) + (1-\beta)\sum_{j=1}^{N} \lambda_j \nabla \widehat{\mathcal{L}}_j(\boldsymbol{\phi}^t) - \beta \nabla \mathcal{L}_i(\boldsymbol{\phi}^t) - \right.\right. \\
&\qquad\qquad \left.\left. - (1-\beta)\nabla \mathcal{L}_i(\boldsymbol{\phi}^t) + \beta \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1})\right\|_2^2\right] \\
&= \sum_{i=1}^{N} \lambda_i \mathbb{E}\left[\left\|\beta(\boldsymbol{g}^{t-1} - \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1})) + (1-\beta)\left(\sum_{j=1}^{N} \lambda_j \nabla \widehat{\mathcal{L}}_j(\boldsymbol{\phi}^t) - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\right) - \right.\right. \\
&\qquad\qquad \left.\left. - \beta(\nabla \mathcal{L}_i(\boldsymbol{\phi}^t) - \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right] \\
&\leq \sum_{i=1}^{N} 3\lambda_i \left\{ \mathbb{E}\left[\left\|\beta(\boldsymbol{g}^{t-1} - \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right] + \mathbb{E}\left[\left\|(1-\beta)\left(\nabla \widehat{\mathcal{L}}(\boldsymbol{\phi}^t) - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\right)\right\|_2^2\right] + \right. \\
&\qquad\qquad \left. + \mathbb{E}\left[\left\|\beta(\nabla \mathcal{L}_i(\boldsymbol{\phi}^t) - \nabla \mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right] \right\}, \qquad (6.15)
\end{aligned}
$$

where the last inequality follows from **Lemma 6.4**.

Apply **Assumption 6.1** in **Eq. 6.15** to bound the next inequality as

$$\begin{aligned}
\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{g}^t-\nabla\mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right] &\leq \sum_{i=1}^{N}3\lambda_i\beta^2\mathbb{E}\left[\left\|(\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right]+\\
&\quad+\sum_{i=1}^{N}3\lambda_i(1-\beta)^2\mathbb{E}\left[\left\|\left(\nabla\widehat{\mathcal{L}}(\boldsymbol{\phi}^t)-\nabla\mathcal{L}_i(\boldsymbol{\phi}^t)\right)\right\|_2^2\right]+\\
&\quad+\sum_{i=1}^{N}3\lambda_i\beta^2L^2\mathbb{E}\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^{t-1}\|_2^2\\
&\leq \sum_{i=1}^{N}3\lambda_i\beta^2\mathbb{E}\left[\left\|(\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right]+3(1-\beta)^2\sum_{i=1}^{N}\lambda_iS_g^2+\sum_{i=1}^{N}3\lambda_i\beta^2L^2\mathbb{E}\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^{t-1}\|_2^2\\
&\overset{(*)}{\leq} \sum_{i=1}^{N}3\lambda_i\beta^2\mathbb{E}\left[\left\|(\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right]+3(1-\beta)^2S_g^2+3\beta^2L^2\mathbb{E}\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^{t-1}\|_2^2\\
&\overset{(**)}{\leq} \sum_{i=1}^{N}3\lambda_i\beta^2\mathbb{E}\left[\left\|(\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right]+3(1-\beta)^2S_g^2+3\beta^2L^2\gamma^2\mathbb{E}\|\boldsymbol{g}^{t-1}\|_2^2,
\end{aligned}$$

where for inequality $(*)$, we apply **Assumption 6.3**, which provides a bound on the variance between the local gradients and the global gradient, and for inequality $(**)$, we use the update rule $\boldsymbol{\phi}^t=\boldsymbol{\phi}^{t-1}+\gamma\boldsymbol{g}^{t-1}$ to express the distance between consecutive server iterates as

$$\|\boldsymbol{\phi}^t-\boldsymbol{\phi}^{t-1}\|_2^2=\|\gamma\boldsymbol{g}^{t-1}\|_2^2=\gamma^2\|\boldsymbol{g}^{t-1}\|_2^2.$$

Next, we use **Lemma 6.10** to bound the next fist inequality as

$$\begin{aligned}
&\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{g}^t-\nabla\mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right]\\
&\leq \sum_{i=1}^{N}3\lambda_i\beta^2\mathbb{E}\left[\left\|(\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right]+3(1-\beta)^2S_g^2+\\
&+3\beta^2L^2\gamma^2\left[\sum_{i=1}^{N}3\lambda_i\mathbb{E}[\|\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1})\|_2^2]+3S_\ell^2+36\left[M^2(K_\star^2\sqrt{d}+1)\|\overline{\boldsymbol{\Delta}}\|_2^2+\frac{d}{6}\right]\right]\\
&\leq (1+3L^2\gamma^2)\sum_{i=1}^{N}3\lambda_i\beta^2\mathbb{E}\left[\left\|(\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right]+3(1-\beta)^2S_g^2+9\beta^2L^2\gamma^2S_\ell^2+\\
&\qquad+108\beta^2L^2\gamma^2\left[M^2(K_\star^2\sqrt{d}+1)\|\overline{\boldsymbol{\Delta}}\|_2^2+\frac{d}{6}\right]. \qquad (6.16)
\end{aligned}$$

Letting $J=3(1-\beta)^2S_g^2+9\beta^2L^2\gamma^2S_\ell^2+108\beta^2L^2\gamma^2\left[M^2(K_\star^2\sqrt{d}+1)\|\overline{\boldsymbol{\Delta}}\|_2^2+\frac{d}{6}\right]$, we can write **Eq. 6.16** as

$$\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{g}^t-\nabla\mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right]\leq(1+3L^2\gamma^2)\sum_{i=1}^{N}3\lambda_i\beta^2\mathbb{E}\left[\left\|(\boldsymbol{g}^{t-1}-\nabla\mathcal{L}_i(\boldsymbol{\phi}^{t-1}))\right\|_2^2\right]+J.$$

Unrolling the recursion, we obtain

$$\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{g}^t-\nabla\mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right]\leq(1+3L^2\gamma^2)^t(3\beta^2)^t\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{m}^0-\nabla\mathcal{L}_i(\boldsymbol{\phi}^0)\right\|_2^2\right]+ \\ +J\sum_{k=0}^{t-1}(1+3L^2\gamma^2)^k(3\beta^2)^k. \quad (6.17)$$

Now, note that the geometric series converges to

$$\sum_{k=0}^{t-1}(1+3L^2\gamma^2)^k(3\beta^2)^k=\frac{\left[(1+3L^2\gamma^2)(3\beta^2)\right]^t-1}{(1+3L^2\gamma^2)(3\beta^2)-1}.$$

Thus, the inequality follows directly from Eq. 6.17

$$\begin{aligned}
&\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{g}^t-\nabla\mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right]\\
&\leq\left[(1+3L^2\gamma^2)3\beta^2\right]^t\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{m}^0-\nabla\mathcal{L}_i(\boldsymbol{\phi}^0)\right\|_2^2\right]+J\frac{\left[(1+3L^2\gamma^2)3\beta^2\right]^t-1}{(1+3L^2\gamma^2)(3\beta^2)-1}\\
&\leq\left[(1+3L^2\gamma^2)3\beta^2\right]^t\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\nabla\widehat{\mathcal{L}}(\boldsymbol{\phi}^0)-\nabla\mathcal{L}_i(\boldsymbol{\phi}^0)\right\|_2^2\right]+J\frac{\left[(1+3L^2\gamma^2)3\beta^2\right]^t-1}{(1+3L^2\gamma^2)(3\beta^2)-1}\\
&\leq\left[(1+3L^2\gamma^2)3\beta^2\right]^tS_g^2+J\frac{\left[(1+3L^2\gamma^2)(3\beta^2)\right]^t-1}{(1+3L^2\eta^2)(3\beta^2)-1}.
\end{aligned}$$

Moreover, condition $(1+3L^2\eta^2)3\beta^2\leq 1$ can be easily satisfied by appropriately choosing η and β. Consequently, the inequality follows

$$\sum_{i=1}^{N}\lambda_i\mathbb{E}\left[\left\|\boldsymbol{g}^t-\nabla\mathcal{L}_i(\boldsymbol{\phi}^t)\right\|_2^2\right]\leq 3S_g^2[(1+3L^2\gamma^2)\beta^2+(1-\beta)^2]+9\beta^2L^2\gamma^2S_\ell^2+ \\ +108\beta^2L^2\gamma^2\left[M^2(K_\star^2\sqrt{d}+1)\|\overline{\boldsymbol{\Delta}}\|_2^2+\frac{d}{6}\right].$$

□

Theorem 2 decomposes the discrepancy between the momentum direction $\boldsymbol{g}^t$ and each client gradient into contributions from (i) global stochastic variance, (ii) local gradient noise, and (iii) client drift amplified through smoothness and momentum. The condition on η ensures that the recursion remains stable (preventing the momentum/heterogeneity terms from exploding), which is essential for establishing convergence of the server updates.

Lemma 6.10. *Let $\mathcal{L}(\boldsymbol{\phi})$ be the global objective and suppose Assumptions 6.1–6.3 hold. In communication round t, each client i performs k local SGD steps starting from $\boldsymbol{\phi}^t$ with*

stepsize $\eta_i = \eta \leq \frac{1}{k}$, *producing* $\boldsymbol{\phi}_i^{t,k}$, *and the server forms* $\boldsymbol{\phi}^{t+1} = \boldsymbol{\phi}^t - \gamma\, \boldsymbol{g}^t$, *where*

$$\boldsymbol{g}^t = \beta \boldsymbol{g}^{t-1} + (1-\beta) \sum_{i=1}^{N} \lambda_i \nabla \widehat{\mathcal{L}}_i(\boldsymbol{\phi}_i^{t,k}).$$

Then

$$\mathbb{E}\big[\|\boldsymbol{g}^t\|_2^2\big] \leq \sum_{i=1}^{N} 3\lambda_i \mathbb{E}\big[\|\boldsymbol{g}^t - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\|_2^2\big] + 3S_\ell^2 + 36\Big[M^2\Big(K^2\sqrt{d}+1\Big)\,\|\overline{\boldsymbol{\Delta}}\|_2^2 + \tfrac{d}{6}\Big].$$

Proof. By the **Lemma 6.4**, we can decompose

$$\begin{aligned}
\mathbb{E}\big[\|\boldsymbol{g}^t\|_2^2\big] &= \mathbb{E}\left[\Big\|\boldsymbol{g}^t \pm \sum_{i=1}^{N} \lambda_i \nabla \mathcal{L}_i(\boldsymbol{\phi}^t) \pm \sum_{i=1}^{N} \lambda_i \nabla \widehat{\mathcal{L}}_i(\boldsymbol{\phi}^t)\Big\|_2^2\right] \\
&\leq 3\,\mathbb{E}\left[\Big\|\boldsymbol{g}^t - \sum_{i=1}^{N} \lambda_i \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\Big\|_2^2\right] + 3\,\mathbb{E}\left[\Big\|\sum_{i=1}^{N} \lambda_i \Big(\nabla \mathcal{L}_i(\boldsymbol{\phi}^t) - \nabla \widehat{\mathcal{L}}_i(\boldsymbol{\phi}^t)\Big)\Big\|_2^2\right] \\
&\quad + 3\,\mathbb{E}\left[\Big\|\sum_{i=1}^{N} \lambda_i \nabla \widehat{\mathcal{L}}_i(\boldsymbol{\phi}^t)\Big\|_2^2\right] \\
&\overset{(1)}{\leq} \sum_{i=1}^{N} 3\lambda_i \mathbb{E}\big[\|\boldsymbol{g}^t - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\|_2^2\big] + 3S_\ell^2 + 3\sum_{i=1}^{N} \lambda_i \mathbb{E}\Big[\|\nabla \widehat{\mathcal{L}}_i(\boldsymbol{\phi}^t)\|_2^2\Big] \\
&= \sum_{i=1}^{N} 3\lambda_i \mathbb{E}\big[\|\boldsymbol{g}^t - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\|_2^2\big] + 3S_\ell^2 + 3\sum_{i=1}^{N} \lambda_i \mathbb{E}\big[\|\nabla g_i(\boldsymbol{\phi}^t) + \nabla h_i(\boldsymbol{\phi}^t)\|_2^2\big] \\
&\overset{(2)}{\leq} \sum_{i=1}^{N} 3\lambda_i \mathbb{E}\big[\|\boldsymbol{g}^t - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\|_2^2\big] + 3S_\ell^2 + 6\sum_{i=1}^{N} \lambda_i \mathbb{E}\big[\|\nabla g_i(\boldsymbol{\phi}^t)\|_2^2\big] + 6\sum_{i=1}^{N} \lambda_i \mathbb{E}\big[\|\nabla h_i(\boldsymbol{\phi}^t)\|_2^2\big] \\
\mathbb{E}\big[\|\boldsymbol{g}^t\|_2^2\big] &\overset{(3)}{\leq} \sum_{i=1}^{N} 3\lambda_i \mathbb{E}\big[\|\boldsymbol{g}^t - \nabla \mathcal{L}_i(\boldsymbol{\phi}^t)\|_2^2\big] + 3S_\ell^2 + 36\Big[M^2\Big(K_\star^2\sqrt{d}+1\Big)\,\|\overline{\boldsymbol{\Delta}}\|_2^2 + \tfrac{d}{6}\Big].
\end{aligned} \tag{6.18}$$

where we use Jensen's inequality ($\|.\|_2^2$ is a convex function) and **Assumption 6.2** to bound (1), and **Lemma 6.4** to bound (2). Finally, in (3), we replaces the unbiased stochastic gradient with **Corollary 6.1**, **Lemma 6.7** and $\|\overline{\boldsymbol{\Delta}}\|_2^2 = \sum_{i=1}^{N} \lambda_i \|\boldsymbol{\Delta}_i\|_2^2$.

□

6.3 Experimental Details

Each client returns a variational posterior $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, where $\boldsymbol{\mu}_j$ is the personalized solution, and $\boldsymbol{\Sigma}_j$ quantifies uncertainty. The confidence weight $\lambda_j = 1/\rho_j^2$, decreases with posterior variance and disagreement with the server model, and the server aggregates client

models using confidence-weighted averaging. We additionally employ server momentum to reduce variance and client drift across rounds.

Finally, we evaluate our uncertainty personalization method in non-IID FL under both benign and malicious participation. The goal is to verify (i) whether the proposed variational formulation improves personalization under heterogeneity, and (ii) whether the induced uncertainty can be used to mitigate malicious client contribution during aggregation.

6.3.1 Parameters initialization and network architecture

We performed our experiments using the Flower framework [Beutel et al., 2020b] in an FL setting characterized by non-IID clients (quantity-based label imbalance) [He et al., 2024]. For each dataset, we sorted the data by labels and divided it into N clients. Each client was assigned $\#S$ random non-overlapping subsets (shards), each containing an equal number of samples [Zhang et al., 2022a].

Thus, we used a server and 50/100 clients in our experiment to evaluate our model, and we trained our method with one NVIDIA RTX 6000 Ada Generation (48 GB) for 500 FL epochs. For each training round, the server selects 10 % of clients to train for five local epochs of the user model. We use the *F1-Score*, a commonly used metric in classification tasks, which can be directly computed from the confusion matrix.

The NN architecture used for all FL approaches, including our proposed method, is identical. Following Zhang et al. [2022b], Zhu et al. [2023], the network dimensions are m-100-n for the FMNIST datasets, where m represents the number of input features and d denotes the latent space representation dimension. We employed SGD with a learning rate of 0.01 for all experiments. All baseline models were configured using the hyperparameters recommended in their respective original publications.

Finally, to evaluate robustness against adversarial participants, we inject model-poisoning clients using the Fang attack [Fang et al., 2020], which crafts malicious local updates that evade common Byzantine-robust rules while degrading global model performance. For any baseline approach that was **not** proposed natively for malicious FL, we deploy it under a robust server-side rule by applying the Krum aggregator [Blanchard et al., 2017] during aggregation to improve resilience to such malicious updates.

Table 6.1: F1-Score of various PFL approaches with **50** and **100** clients with differentes malicious clients number $\#S$. The best results are in **bold**, and the second-best are underlined.

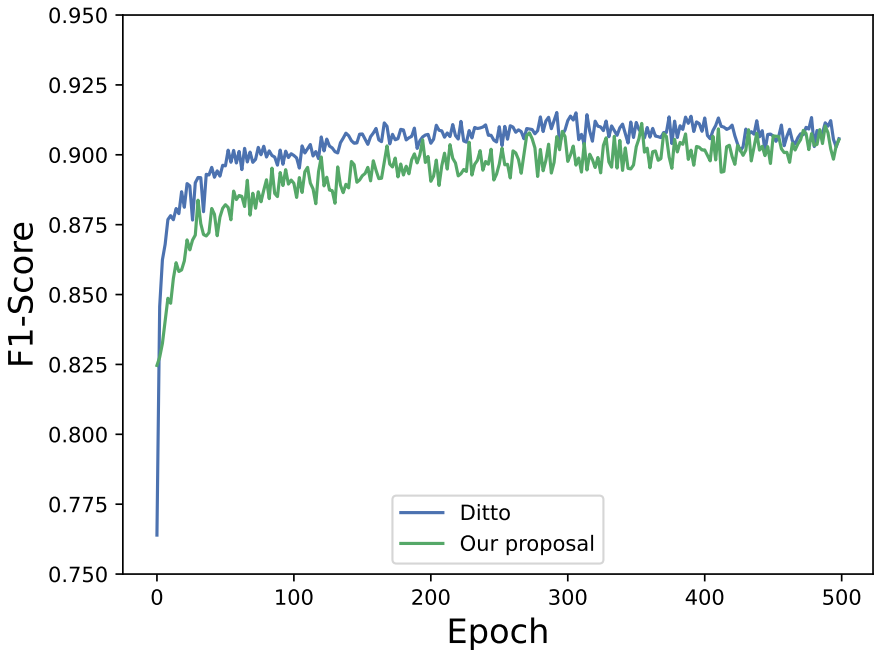
Proposal	50 Clients				100 Clients			
	$\#M = 0$	$\#M = 2$	$\#M = 5$	$\#M = 10$	$\#M = 0$	$\#M = 2$	$\#M = 5$	$\#M = 10$
FedAvg. (GM)	0.6656	0.4689	0.4407	0.4634	0.5921	0.4528	0.4497	0.4084
FedRep	0.8999	0.8310	0.8244	0.8035	0.8729	0.8069	0.7844	0.7695
pFedBayes	0.8975	0.7832	0.7791	0.7516	0.8628	0.7894	0.7601	0.7276
FedAvg. (PM)	0.8470	0.8338	0.8295	0.8105	0.8284	0.7822	0.7751	0.7429
APFL	0.9064	0.8179	0.8128	0.8091	0.8643	0.8472	0.8350	0.7998
LGFedAvg	0.8933	0.8218	0.8231	0.7992	0.8460	0.8136	0.8006	0.7717
pfedvem	0.8835	0.8339	0.8257	0.8475	0.8178	0.8153	0.8573	0.8256
DITTO	**0.9072**	0.8449	0.8358	0.8195	0.8549	0.8693	0.8377	0.8159
Our proposal	0.9003	**0.8982**	**0.8901**	**0.8634**	**0.8991**	**0.8871**	**0.8770**	**0.8562**

6.4 Results

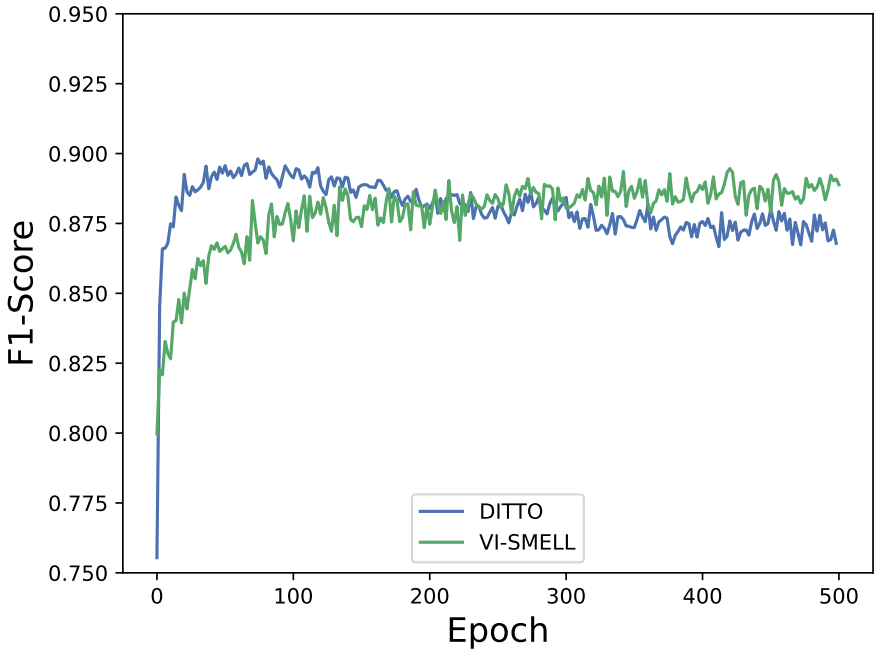
In this chapter, we compare our proposal with six federated learning approaches focused on personalization (FedRep, pFedBayes, APFL, LGFedAvg, pfedvem, and DITTO) and two variants of FedAvg (global model "GM" and personalized model "PM"). Table 6.1 reports the F1-scores obtained with 50 and 100 clients with different numbers of malicious participants ($\#M$). This setup allows us to evaluate the baseline performance in benign environments and the robustness of each approach when exposed to poisoning attacks.

For the case of 50 clients without malicious clients ($\#M = 0$), DITTO achieved the best overall performance (0.9072), followed by APFL (0.9064). Our approach achieved 0.9003, ranking as the third-best approach. In contrast, FedAvg (GM) reached only 0.6656, highlighting the poor adaptability of a global model under heterogeneous data distributions. When setting to 100 clients, our proposal achieved the best result for this FL setting without malicious clients, reaching 0.8991, surpassing all other techniques. FedRep obtained the second-best result (0.8729), followed by APFL (0.8643). Besides, several approaches that performed well in the 50-client experiment have lower results with 100 clients. For example, DITTO decreased by 4.73% (from 0.9072 to 0.8549), while APFL decreased by 4.64% (from 0.9064 to 0.8643). Similar degradations were observed for pFedBayes and LGFedAvg. In contrast, our method decreased from 0.9003 to 0.8991 (0.133%), showing resilience to the increased number of participants (data heterogeneity). Finally, we evaluated the robustness with different numbers of malicious clients. As expected, all methods degraded as $\#M$ increased. However, our proposal consistently delivered superior results on both client scales. For example, with 50 clients and $\#M = 2$, our approach achieved an F1-score of 0.8982, compared to 0.8449 for the second-best method (DITTO). At $\#M = 10$, our method still retained 0.8634, while DITTO and

APFL dropped to 0.8195 and 0.8091, respectively. These conclusions are consistent across 100 client scenarios. Figure 6.1 shows the F1-score in epochs for our method versus Ditto in the FMNIST dataset.



(a) No malicious clients.



(b) 2 malicious clients out of 100 total (2%).

Figure 6.1: F1-score vs. epoch on FMNIST dataset, comparing our method to Ditto under two federation settings: (a) benign (no attackers) and (b) with two malicious clients (2% of 100).

6.5 Related Work

Security issues in machine learning systems, and consequently FL, have been extensively studied [Zhao et al., 2017, Wang et al., 2020, Rodríguez-Barroso et al., 2022]. In traditional machine learning, the learning phase is typically protected and centralized in a unique system [Lamport et al., 1982]. Specifically, approaches in the literature usually consider that malicious clients act during inference, i.e., the attacked model is already in production [Jagielski et al., 2018]. In FL, malicious clients usually exploit the vulnerability of the models during the learning phase [Bhagoji et al., 2019]. In general, malicious clients attacking an FL model have one of two adversarial goals:

Case I. Reconstruct or learn client/model information based on data transmitted in the federated training process and

Case II. Force the model to behave differently than intended, invalidate, or train it for a specific purpose (e.g., poisoning attack).

In this chapter, we focus on the problems related to attacks that aim to degrade the performance of the aggregated model (type II attacks).

Several works in the literature propose a poisoning attack method to FL models to degrade the training process. Zhang et al. [2021b] propose using generative adversarial networks (GANs) to generate examples without making any assumptions about accessing the participants' training data. Similarly, Zhang et al. [2019] inserts adversarial poison samples assigned with the wrong label to the local training dataset to degrade the aggregate model. Sun et al. [2022] studies the vulnerability of FL models in IoT systems. Thus, the authors use a bilevel optimization consideration, which injects poisoned data samples to maximize the deterioration of the aggregate model. Besides, Defense mechanisms for distributed poisoning attacks typically draw ideas from robust estimation and anomaly detection [Alistarh et al., 2018, Shejwalkar et al., 2022]. Some works are based on aggregation functions robust to outliers, such as median [Wu et al., 2022], mean with exclusion [Fang et al., 2020], geometric mean [Pillutla et al., 2022], and clustering of nearby gradients [Blanchard et al., 2017].

Additionally, assuming scenarios where all clients train the network model but use their data is paramount. More often than not, we observe that data models differ across clients. Therefore, the resulting models will be abstractions of different real-world conditions. Ultimately, we need to propose solutions that address such situations, taking into account the challenges of accounting for these differences without assuming malicious intent, especially when only a small subset of clients presents data models that differ from the majority.

6.6 Final Remarks

This chapter proposes a novel VI model for PFL under malicious participation. We introduced an uncertainty-aware framework that (i) yields bounds for reparameterized gradient estimators, (ii) motivates confidence-weighted aggregation, and (iii) integrates server momentum to reduce variance and client drift. Our analysis derivates convergence guaranties under exponential reparameterization and standard smoothness/convexity assumptions, extending prior results beyond linear parameterizations.

Our theory results consider smoothness and strong-convexity surrogates and mean-field Gaussian variational families. Relaxing these assumptions is a promising direction. Extending the approach to adaptive participation, stronger Byzantine adversaries, and communication-efficient uncertainty sharing (e.g., low-rank covariance summaries) are open next steps. Finally, evaluating diverse modalities (vision, language, tabular) and real-world federations with system heterogeneity will further validate practicality and scalability.

Chapter 7

Final remarks

This thesis examines how to improve the security and reliability of FL, particularly when the FL network is vulnerable to adversarial threats, such as model poisoning attacks. Thus, in this thesis, we proposed some different approaches to quantify uncertainty in FL to detect and mitigate malicious behavior without compromising user privacy or system performance.

We proposed and evaluated three different uncertainty strategies; (i) using Laplace approximations based on second-order derivatives, (ii) introducing the SMELL (Ad-Hoc) framework with similarity-based learning, and (iii) leveraging VI to enhance the SMELL framework. Each of these methods tackles uncertainty from a different perspective, helping to understand how confident (or uncertain) a model is about its predictions.

Specifically, Chapter 2 explored the use of Laplace approximations for uncertainty estimation in Neural Networks. This method applies a second-order derivative to provide a way to estimate the confidence a model has in its predictions. Applying this to FL settings, we show that uncertainty estimates can help identify clients whose updates deviate from expected patterns (i.e., label flip attacks), suggesting potential malicious activity.

Chapter 3 and 4 introduced the SMELL framework, a novel approach that takes advantage of model update similarity to detect adversaries. We hypothesize that trustworthy clients tend to behave similarly over time, although malicious clients produce updates that are inconsistent or anomalous. By tracking these patterns, SMELL was able to detect poisoning attacks.

Chapter 5 focused on BNN and variational inference to build more expressive uncertainty-aware models. Although more computationally demanding, this approach enables the mathematical formal modeling of uncertainty, particularly in FL data settings. We demonstrated that SMELL Bayesian models have a lower bound limit compared to distributed BNN models for quantifying uncertainty.

Lastly, chapter 6 combines the previous chapter in FL attack scenarios. We evaluated the trade-offs between detection accuracy and robustness for each method presented and present convergence analyses for both centralized and distributed SMELL-VI approaches. The results showed evidence that uncertainty quantification improves the

resilience of FL systems under attack.

7.1 Future Directions

We intend to tackle the following activities after this PhD thesis:

Task I. **Score Matching for Federated Learning** aims to use score matching for identifying authentic versus adversarial updates, enhancing model security, and handling out-of-distribution data effectively, as we describe in Section 7.1.1.

Task II. Contribute to advancing **Federated Continual Learning**, as we describe in Section 7.1.2.

7.1.1 Score Matching for Federated Learning

Integrating Score matching into FL can mitigate challenges, particularly the issue of non-IID (heterogeneous local data distributions) inherent in FL environments [Li et al., 2024]. FL, characterized by its distributed machine learning framework, enables model training across numerous devices while maintaining data privacy by keeping the data localized. Score matching [Hyvärinen and Dayan, 2005] provides a mathematical foundation to navigate these challenges effectively. It approximates the probability distributions by minimizing the Score-matching objective, formally defined as

$$J(\theta) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[\|\nabla_x \log p_{model}(\boldsymbol{x} \mid \theta) - \nabla_x \log p_{data}(\boldsymbol{x})\|_2^2\right],$$

where p_{data} represents the data distribution, $p_{model}(\boldsymbol{x} \mid \theta)$ denotes the model distribution parameterized by θ, and ∇_x is the gradient with respect to the input $\boldsymbol{x}$.

Furthermore, the strategic application of score-matching within FL settings is advantageous for enhancing system robustness against adversarial attacks. Utilizing score matching to discern between malicious updates and outliers (such as updates from clients with legitimate, though significantly different, data distribution) is crucial for maintaining system integrity.

Research problem to be investigated: Through an evaluation of update gradients in alignment with the anticipated distribution gradients, score matching

offers a method to validate the authenticity of model updates. This approach can be a framework to identify adversarial updates while safeguarding insights from out-of-distribution (OOD) data contributions.

7.1.2 Robust Federated Continual learning

Continuous learning (or incremental learning) is the paradigm that accumulates previous knowledge to solve a sequence of tasks (e.g., new classes discovered). Continuous learning systems must be designed to train tasks incrementally without revisiting all previous data at each stage. A fundamental challenge in this paradigm is to avoid forgetting previous knowledge, namely catastrophic forgetting [Yoon et al., 2021]. Due to privacy restrictions imposed by FL environments, federated continual learning presents new challenges to continuous learning, such as utilizing knowledge from other clients while preventing interference from irrelevant knowledge [Yoon et al., 2021].

For example, one of the most relevant privacy problems is inferring whether a program has malicious intent (malware software) [Antonakakis et al., 2017]. Today, cyberattacks are one of the main concerns in computer networks. Many attacks ranging from naïve viruses to ransomware and then to sophisticated malware like Stuxnet [Langner, 2011] jeopardize individual users' privacy/safety and endanger entire nations' sovereignty. Cyberattacks are so relevant nowadays that governments are prioritizing cybersecurity[1]. The detection of malicious software that "error"– or *malware* for short [Petrik et al., 2018]– is one of the most relevant security problems. For instance, a single malicious software can cause up to millions of dollars in damage [Anderson et al., 2013]. Software codes, and consequently bytes of binary code, exhibit several types of spatial correlation. These correlations, in turn, have discontinuities in function calls and jump commands, making identifying malware a hard task [Raff et al., 2018].

Even though Antivirus is one of the most popular approaches for malware detection, new types of malware are released quickly, making most techniques for detecting them quickly obsolete [Anderson et al., 2013]. Thus, regular Antivirus typically fails to detect new malware until their signature is incorporated into their database [Barros et al., 2022a]. Nevertheless, new techniques to identify unknown new malware are necessary to protect systems even at day zero of a malware release.

A significant challenge in federated continual learning is the inability to revisit previous tasks for validation due to privacy constraints, which could allow malicious attacks to propagate undetected to subsequent tasks. This presents a unique vulnerability

[1] https://www.voanews.com/usa/us-biden-voice-new-alarm-about-cyberattack

in federated continual learning, where safeguarding against such threats while adhering to privacy requirements becomes a pressing concern.

Research problem to be investigated: Given this problem, we quantify data adherence to the previous and new knowledge simultaneously and, with this quantifier, perform local model training using these two objective functions (e.g., meta-learning approaches). Bayesian neural networks are also helpful in online learning; previous posteriors can be recycled as priors when new data becomes available to avoid catastrophic forgetting [Ritter et al., 2018]. The federated continuous learning paradigm, which is a promising open direction, can benefit many applications (e.g., DDoS attack or malware detection).

7.2 Publications

In the following sections, we list the publications resulting from this thesis. The list is divided into three categories: (i) journal papers, (ii) conference papers, and (iii) papers under submission. Publications marked with an (X) indicate a direct contribution or extension in this thesis.

7.2.1 Periodical papers

- (X) P. H. Barros, J. C. Guevara, L. Villas, D. Guidoni, N. L. S. d. Fonseca, and H. S. Ramos. A novel federated meta-learning approach for discriminating sedentary behavior from wearable data. *IEEE Internet of Things Journal*, 11(19):31909–31916, 2024b. doi: 10.1109/JIOT.2024.3420891

- (X) P. H. Barros, E. T. Chagas, L. B. Oliveira, F. Queiroz, and H. S. Ramos. Malware-SMELL: A zero-shot learning strategy for detecting zero-day vulnerabilities. *Computers & Security*, 120:102785, 2022a

- (X) P. H. Barros, F. Queiroz, F. Figueiredo, J. A. D. Santos, and H. Ramos. A new similarity space tailored for supervised deep metric learning. *ACM Transactions on Intelligent Systems and Technology*, 14(1), nov 2022b. ISSN 2157-6904

- F. A. Silva, O. Orang, F. J. Erazo-Costa, P. C. Silva, P. H. Barros, R. P. Ferreira, and F. G. Guimarães. Time series classification using federated convolutional neural networks and image-based representations. *IEEE Access*, 2025a

- I. Cardoso-Pereira, J. B. Borges, P. H. Barros, A. F. Loureiro, O. A. Rosso, and H. S. Ramos. Leveraging the self-transition probability of ordinal patterns transition network for transportation mode identification based on gps data. *Nonlinear Dynamics*, 107(1):889–908, 2022

7.2.2 Conference papers

- (X) P. H. Barros, J. C. Guevara, T. Polido, L. Villas, D. Guidoni, N. L. S. da Fonseca, and H. S. Ramos. Personalized federated learning for sedentary behavior classification with heterogeneous feature distributions under adversarial threats. *International Joint Conference on Neural Networks (IJCNN)*, TBD(TBD):TBD, TBD 2025a

- (X) P. H. Barros, F. Murai, A. Houmansadr, A. A. F. Loureiro, A. C. Frery, and H. S. Ramos. Mitigação de envenenamento de rótulos em sistemas de detecção de ddos federados. In *Anais do XLIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, page TBD. SBC, 2025b

- (X) P. H. Barros, F. Murai, A. Houmansadr, A. C. Frery, and H. S. Ramos. Variational inference in similarity spaces: A bayesian approach to personalized federated learning. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*, 2024c. URL https://openreview.net/forum?id=VOEVnNjJza

- (X) P. H. Barros, J. C. Guevara, L. Villas, D. Guidoni, N. L. S. da Fonseca, and H. S. Ramos. Hierarchical federated learning based on ordinal patterns for detecting sedentary behavior. In *2024 International Joint Conference on Neural Networks (IJCNN 2024)*, Yokohama, Japan, June 2024a

- (X) P. H. Barros, M. Q. Oliveira, O. Orang, F. A. da Silva, F. J. Erazo-Costa, A. T. Bastos, P. C. Silva, G. S. dos Santos, A. A. Loureiro, M. G. Ravetti, et al. Flautim: A federated learning platform using k8s and flower. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 87–90. SBC, 2024e

- (X) P. H. Barros, F. Murai, and H. S. Ramos. Bayes and laplace versus the world: A new label attack approach in federated environments based on bayesian neural

networks. In M. C. Naldi and R. A. C. Bianchi, editors, *Intelligent Systems*, pages 449–463, Cham, 2023. Springer Nature Switzerland

- (X) P. H. Barros and H. S. Ramos. A novel aggregation method to promote safety security for poisoning attacks in federated learning. In *GLOBECOM - IEEE Global Communications Conference*, pages 3869–3874, 2022

- O. Orang, P. H. Barros, G. Z. de Castro, and F. G. Guimarães. An efficient one-shot federated medical imaging via variational inference parametric feature transfer. In *Medical Imaging meets EurIPS: MedEurIPS 2025*, 2025

- P. C. L. Silva, O. Orang, P. H. Barros, F. A. R. da Silva, H. S. Ramos, and F. G. Guimarães. Driver Maneuver Classification Based on Multivariate Fuzzy Time Series. In *Proceedings of the 2025 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2025b

- G. Riqueti, P. H. Barros, J. Borges, F. Cunha, O. Rosso, and H. Ramos. SAXJS: An online change point detection for wearable sensor data. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 351–364, Porto Alegre, RS, Brasil, 2023. SBC

- J. Jorge, P. H. Barros, R. Yokoyama, D. Guidoni, H. S. Ramos, N. Fonseca, and L. Villas. Applying federated learning in the detection of freezing of gait in parkinson's disease. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, pages 195–200, 2022

- M. Gomes, P. H. Barros, and H. Ramos. Explorando a correlação espaço-temporal no agrupamento de sensores de cidades inteligentes. In *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 43–55, Porto Alegre, RS, Brasil, 2022. SBC

7.2.3 UnderSubmission

- (X) P. H. Barros, F. Murai, A. Houmansadr, A. A. F. Loureiro, and H. S. Ramos. Towards a federated DDoS detection system robust to label poisoning attacks. TBD (TBD):TBD, TBD 2024d. ISSN 1941-0018. doi: TBD. On going work

Bibliography

E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3908–3916, USA, 2015. IEEE.

B. E. Ainsworth, W. L. Haskell, S. D. Herrmann, N. Meckes, D. R. Bassett Jr, C. Tudor-Locke, J. L. Greer, J. Vezina, M. C. Whitt-Glover, and A. S. Leon. 2011 compendium of physical activities: a second update of codes and met values. *Medicine & science in sports & exercise*, 43(8):1575–1581, 2011.

M. N. Ali, M. Imran, M. S. u. din, and B.-S. Kim. Low rate ddos detection using weighted federated learning in sdn control plane in iot network. *Applied Sciences*, 13(3), 2023.

D. Alistarh, Z. Allen-Zhu, and J. Li. Byzantine stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

R. Anderson, C. Barton, R. Böhme, R. Clayton, M. V. Eeten, M. Levi, T. Moore, and S. Savage. Measuring the cost of cybercrime. In *The economics of information security and privacy*, pages 265–300. Springer, 2013.

M. Antonakakis et al. Understanding the mirai botnet. In *USENIX Security Symposium*, pages 1093–1110, 2017.

M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 2938–2948. PMLR, 2020.

J. Bai, Q. Song, and G. Cheng. Efficient variational inference for sparse deep learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 33: 466–476, 2020.

O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga. mhealthdroid: a novel framework for agile development of mobile health applications. In *Ambient Assisted Living and Daily Activities: 6th International Work-Conference, IWAAL, Belfast, UK.*, pages 91–98. Springer, 2014.

Y. Bansal et al. For self-supervised learning, rationality implies generalization, provably. In *International Conference on Learning Representations (ICLR)*, 2020.

P. H. Barros and H. S. Ramos. A novel aggregation method to promote safety security for poisoning attacks in federated learning. In *GLOBECOM - IEEE Global Communications Conference*, pages 3869–3874, 2022.

P. H. Barros, E. T. Chagas, L. B. Oliveira, F. Queiroz, and H. S. Ramos. MalwareSMELL: A zero-shot learning strategy for detecting zero-day vulnerabilities. *Computers & Security*, 120:102785, 2022a.

P. H. Barros, F. Queiroz, F. Figueiredo, J. A. D. Santos, and H. Ramos. A new similarity space tailored for supervised deep metric learning. *ACM Transactions on Intelligent Systems and Technology*, 14(1), nov 2022b. ISSN 2157-6904.

P. H. Barros, F. Murai, and H. S. Ramos. Bayes and laplace versus the world: A new label attack approach in federated environments based on bayesian neural networks. In M. C. Naldi and R. A. C. Bianchi, editors, *Intelligent Systems*, pages 449–463, Cham, 2023. Springer Nature Switzerland.

P. H. Barros, J. C. Guevara, L. Villas, D. Guidoni, N. L. S. da Fonseca, and H. S. Ramos. Hierarchical federated learning based on ordinal patterns for detecting sedentary behavior. In *2024 International Joint Conference on Neural Networks (IJCNN 2024)*, Yokohama, Japan, June 2024a.

P. H. Barros, J. C. Guevara, L. Villas, D. Guidoni, N. L. S. d. Fonseca, and H. S. Ramos. A novel federated meta-learning approach for discriminating sedentary behavior from wearable data. *IEEE Internet of Things Journal*, 11(19):31909–31916, 2024b. doi: 10.1109/JIOT.2024.3420891.

P. H. Barros, F. Murai, A. Houmansadr, A. C. Frery, and H. S. Ramos. Variational inference in similarity spaces: A bayesian approach to personalized federated learning. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*, 2024c. URL https://openreview.net/forum?id=VOEVnNjJza.

P. H. Barros, F. Murai, A. Houmansadr, A. A. F. Loureiro, and H. S. Ramos. Towards a federated DDoS detection system robust to label poisoning attacks. TBD(TBD):TBD, TBD 2024d. ISSN 1941-0018. doi: TBD. On going work.

P. H. Barros, M. Q. Oliveira, O. Orang, F. A. da Silva, F. J. Erazo-Costa, A. T. Bastos, P. C. Silva, G. S. dos Santos, A. A. Loureiro, M. G. Ravetti, et al. Flautim: A federated learning platform using k8s and flower. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 87–90. SBC, 2024e.

P. H. Barros, J. C. Guevara, T. Polido, L. Villas, D. Guidoni, N. L. S. da Fonseca, and H. S. Ramos. Personalized federated learning for sedentary behavior classification with heterogeneous feature distributions under adversarial threats. *International Joint Conference on Neural Networks (IJCNN)*, TBD(TBD):TBD, TBD 2025a.

P. H. Barros, F. Murai, A. Houmansadr, A. A. F. Loureiro, A. C. Frery, and H. S. Ramos. Mitigação de envenenamento de rótulos em sistemas de detecção de ddos federados. In *Anais do XLIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, page TBD. SBC, 2025b.

O. Barut, L. Zhou, and Y. Luo. Multitask lstm model for human activity recognition and intensity estimation using wearable sensor data. *IEEE Internet of Things Journal*, 7 (9):8760–8768, 2020. doi: 10.1109/JIOT.2020.2996578.

D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane. Flower: A friendly federated learning research framework, 2020a.

D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020b.

A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, volume 97, 2019. 36th International Conference on Machine Learning (ICML), Long Beach, CA, JUN 09-15, 2019.

P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, page 1613–1622, 2015.

J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 737–744, 1994.

F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff. Deep metric learning to rank. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1861–1870, USA, 2019. IEEE.

X. Cao, Y. Ge, R. Li, J. Zhao, and L. Jiao. Hyperspectral imagery classification with deep metric learning. *Neurocomputing*, 356:217 – 227, 2019.

I. Cardoso-Pereira, J. B. Borges, P. H. Barros, A. F. Loureiro, O. A. Rosso, and H. S. Ramos. Leveraging the self-transition probability of ordinal patterns transition network for transportation mode identification based on gps data. *Nonlinear Dynamics*, 107(1): 889–908, 2022.

E. T. C. Chagas, A. C. Frery, J. Gambini, M. M. Lucini, H. S. Ramos, and A. A. Rey. Statistical properties of the entropy from ordinal patterns. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(11):113118, 11 2022.

H. Chen, H. Liu, L. Cao, and T. Zhang. Bayesian personalized federated learning with shared and personalized uncertainty representations. *arXiv preprint arXiv:2309.15499*, 2023.

J. Chen, Q. Guo, Z. Fu, Q. Shang, H. Ma, and D. Wu. Campus network intrusion detection based on federated learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022.

K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie. A semisupervised recurrent convolutional attention model for human activity recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5):1747–1756, 2020.

L.-Y. Chen, T.-C. Chiu, A.-C. Pang, and L.-C. Cheng. Fedequal: Defending model poisoning attacks in heterogeneous federated learning. In *IEEE Global Communications Conference (GLOBECOM)*, 2021.

Z. Cheng, X. Huang, P. Wu, and K. Yuan. Momentum benefits non-iid federated learning simply and provably, 2024. URL https://arxiv.org/abs/2306.16504.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

L. Corinzia, A. Beuret, and J. M. Buhmann. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*, 2019.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*, chapter 8, pages 243–259. John Wiley & Sons, Ltd, 2005. ISBN 9780471748823.

N.-N. Dao, T. V. Phan, U. Sa'ad, J. Kim, T. Bauschert, D.-T. Do, and S. Cho. Securing heterogeneous IoT with intelligent DDoS attack behavior learning. *IEEE Systems Journal*, 16(2):1974–1983, 2022.

E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103, 2021.

R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

M. Deudon. Learning semantic similarity in a continuous space. In *Advances in Neural Information Processing Systems 31*, pages 986–997, 2018.

M. Dimolianis, D. K. Kalogeras, N. Kostopoulos, and V. Maglaris. Ddos attack detection via privacy-aware federated learning and collaborative mitigation in multi-domain cyber infrastructures. In *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, pages 118–125, 2022.

C. T. Dinh, N. H. Tran, and T. D. Nguyen. Personalized federated learning with moreau envelopes. In *Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020.

K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *International Conference on Computer Vision (ICCV)*, pages 5747–5756, Italy, Oct 2017. IEEE.

J. Domke. Provable smoothness guarantees for black-box variational inference. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2587–2596. PMLR, 13–18 Jul 2020.

J. Domke, R. Gower, and G. Garrigos. Provable convergence guarantees for black-box variational inference. *Advances in neural information processing systems*, 36:66289–66327, 2023.

A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

L. F. Eliyan and R. Di Pietro. Dos and ddos attacks in software defined networks: A survey of existing solutions and research challenges. *Future Generation Computer Systems*, 122:149–171, 2021.

A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

M. Fang, X. Cao, J. Jia, and N. Gong. Local model poisoning attacks to Byzantine-Robust federated learning. In *29th USENIX Security Symposium*, pages 1605–1622. USENIX Association, 2020.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

G. Gad and Z. Fadlullah. Federated learning via augmented knowledge distillation for heterogenous deep human activity recognition systems. *Sensors*, 23(1):6, 2023.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

A. Ganguly and S. W. Earp. An introduction to variational inference. *arXiv preprint arXiv:2108.13083*, 2021.

G. Garrigos and R. M. Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521 (7553):452–459, 2015.

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 513–520, Columbia, 2005. Curran Associates.

M. Gomes, P. H. Barros, and H. Ramos. Explorando a correlação espaço-temporal no agrupamento de sensores de cidades inteligentes. In *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 43–55, Porto Alegre, RS, Brasil, 2022. SBC.

T. Gönül, O. D. Incel, and G. Isiklar Alptekin. Human activity recognition with smart watches using federated learning. In *International Conference on Intelligent and Fuzzy Systems*, pages 77–85, 2022.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik. Sgd: General analysis and improved rates. In *International conference on machine learning*, pages 5200–5209. PMLR, 2019.

T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, ICML'17, 2017.

O. Gupta and R. Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

M. Hao et al. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 16(10):6532–6542, 2020.

Z. He, Y. Li, D. Seo, and Z. Cai. Fedcpd: Addressing label distribution skew in federated learning with class proxy decoupling and proxy regularization. *Information Fusion*, 110:102481, 2024.

D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile systems*, pages 53–60, USA, 1999.

A. M. Hotti, L. A. Van der Goten, and J. Lagergren. Benefits of non-linear scale parameterizations in black box variational inference through smoothness results and gradient variance bounds. In S. Dasgupta, S. Mandt, and Y. Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3538–3546. PMLR, 02–04 May 2024.

R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen. Interaction-and-aggregation network for person re-identification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9317–9326, USA, June 2019. IEEE.

Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang. Personalized cross-silo federated learning on non-iid data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7865–7873, May 2021.

A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

A. Immer et al. Scalable marginal likelihood estimation for model selection in deep learning. In *International Conference on Machine Learning (ICML)*, volume 139, pages 4563–4573, 18–24 Jul 2021.

S. Inaba, C. T. Fakhry, R. V. Kulkarni, and K. Zarringhalam. A free energy based approach for distance metric learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 5–13, New York, NY, USA, 2019. ACM.

W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari. Blockchain-based federated learning for securing internet of things: A comprehensive survey. *ACM Computing Surveys*, 55(9), jan 2023.

M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35, 2018.

J. Jorge, P. H. Barros, R. Yokoyama, D. Guidoni, H. S. Ramos, N. Fonseca, and L. Villas. Applying federated learning in the detection of freezing of gait in parkinson's disease. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, pages 195–200, 2022.

L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.

S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143, 2020.

M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar. Adaptive gradient-based meta-learning methods. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22, 2019.

B. Kim and J. C. Ye. Energy-based contrastive learning of visual representations. *Advances in Neural Information Processing Systems*, 35:4358–4369, 2022.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

J. Ko, K. Kim, W. C. Kim, and J. R. Gardner. Provably scalable black-box variational inference with structured variational families. *arXiv preprint arXiv:2401.10989*, 2024.

G. Koch et al. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

N. Kotelevskii, M. Vono, A. Durmus, and E. Moulines. Fedpop: A bayesian approach for personalised federated learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8687–8701. Curran Associates, Inc., 2022.

A. Kristiadi, M. Hein, and P. Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, 2020.

L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security Privacy*, 9(3): 49–51, 2011.

H. Leutheuser, S. Doelfel, D. Schuldhaus, S. Reinfelder, and B. M. Eskofier. Performance comparison of two step segmentation algorithms using different step activities. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks*, pages 143–148, 2014.

A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen. Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, page 42–55, New York, NY, USA, 2021a. ISBN 9781450390972.

C. Li, D. Niu, B. Jiang, X. Zuo, and J. Yang. Meta-har: Federated representation learning for human activity recognition. In *Proceedings of the Web Conference 2021*, WWW '21, page 912–922, 2021b.

F. Li, H. Qiao, and B. Zhang. Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition*, 83:161–173, 2018.

J. Li, Z. Zhang, Y. Li, X. Guo, and H. Li. FIDS: Detecting DDoS through federated learning based method. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 856–862, 2021c.

J. Li et al. FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT. *IEEE Transactions on Industrial Informatics*, 18(6):4059–4068, 2022a.

Q. Li, Y. Diao, Q. Chen, and B. He. Federated learning on non-iid data silos: An experimental study. In *IEEE International Conference on Data Engineering*, 2022b.

T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, pages 50–60, 2020a.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020b.

T. Li, S. Hu, A. Beirami, and V. Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021d.

Z. Li, Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang. Feature matching data synthesis for non-iid federated learning. *IEEE Transactions on Mobile Computing*, 2024.

P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency. Think locally, act globally: Federated learning with local and global representations, 2020.

W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 22(3):2031–2063, 2020.

Y. Lin, J. Jiang, and S. Lee. A similarity measure for text classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1575–1590, 2014.

S. Liu, S. Lv, D. Zeng, Z. Xu, H. Wang, and Y. Yu. Personalized federated learning via amortized bayesian meta-learning. *arXiv preprint arXiv:2307.02222*, 2023a.

X. Liu, Y. Li, C. Wu, and C.-J. Hsieh. Adv-BNN: Improved adversarial defense through robust bayesian neural network. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rk4Qso0cKm.

Z. Liu, C. Guo, D. Liu, and X. Yin. An asynchronous federated learning arbitration model for low-rate DDoS attack detection. *IEEE Access*, 11:18448–18460, 2023b.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

D. Lv, X. Cheng, J. Zhang, W. Zhang, W. Zhao, and H. Xu. Ddos attack detection based on cnn and federated learning. In *2021 Ninth International Conference on Advanced Cloud and Big Data (CBD)*, pages 236–241, 2022.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems 32*, pages 480–491. Curran Associates, Inc., Canada, 2019.

A. Mashhadi, J. Sterner, and J. Murray. Deep embedded clustering of urban communities using federated learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. doi: 10.1109/IJCNN52387.2021.9534268.

B. McFee and G. R. Lanckriet. Metric learning to rank. In *International Conference on Machine Learning (ICML)*, pages 775–782, 2010.

B. McMahan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. In *Proceedings of the International Conference on Machine Learning*, pages 4615–4625, 2019.

S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers. *Statistical Science*, 27(4):538 – 557, 2012.

E. C. P. Neto, S. Dadkhah, and A. A. Ghorbani. Collaborative DDoS detection in distributed multi-tenant iot using federated learning. In *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*, pages 1–10, 2022.

B. Nguyen, C. Morell, and B. D. Baets. Supervised distance metric learning through maximization of the jeffrey divergence. *Pattern Recognition*, 64:215 – 225, 2017. ISSN 0031-3203.

D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021.

A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

O. Orang, P. H. Barros, G. Z. de Castro, and F. G. Guimarães. An efficient one-shot federated medical imaging via variational inference parametric feature transfer. In *Medical Imaging meets EurIPS: MedEurIPS 2025*, 2025.

K. Panchal, S. Choudhary, S. Mitra, K. Mukherjee, S. Sarkhel, S. Mitra, and H. Guan. Flash: Concept drift adaptation in federated learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 26931–26962. PMLR, 23–29 Jul 2023a.

K. Panchal, S. Choudhary, N. Parikh, L. Zhang, and H. Guan. Flow: Per-instance personalized federated learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 18712–18755. Curran Associates, Inc., 2023b.

H. Park, H. Hosseini, and S. Yun. Federated learning with metric loss. In *Workshop on Federated Learning for User Privacy and Data Confidentiality in ICML21*, 2021.

X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi. Deep subspace clustering with sparsity prior. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 1925–1931. AAAI Press, 2016.

R. Petrik, B. Arik, and J. Smith. Towards architecture and os-independent malware detection via memory forensics. In *ACM Conference on Computer and Communications Security (SIGSAC)*, CCS '18, page 2267–2269, New York, NY, USA, 2018. Association for Computing Machinery.

K. Pillutla, S. M. Kakade, and Z. Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

N. G. Polson and V. Ročková. Posterior concentration for sparse deep learning. *Advances in Neural Information Processing Systems*, 31, 2018.

W. Possos, R. Cruz, J. D. Cerón, D. M. López, and C. H. Sierra-Torres. Open dataset for the automatic recognition of sedentary behaviors. *Studies in health technology and informatics*, page 107—114, 2017.

R. Presotto, G. Civitarese, and C. Bettini. Federated clustering and semi-supervised learning: A new partnership for personalized human activity recognition. *Pervasive and Mobile Computing*, 88:101726, 2023.

E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas. Malware detection by eating a whole exe. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

R. Ranganath, S. Gerrish, and D. Blei. Black Box Variational Inference. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland, 22–25 Apr 2014. PMLR.

A. Reiss and D. Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*. Association for Computing Machinery, 2012.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of the 31st International Conference on Machine Learning*, 32(2):1278–1286, 22–24 Jun 2014.

G. Riqueti, P. H. Barros, J. Borges, F. Cunha, O. Rosso, and H. Ramos. SAXJS: An online change point detection for wearable sensor data. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 351–364, Porto Alegre, RS, Brasil, 2023. SBC.

H. Ritter, A. Botev, and D. Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS, 2018.

N. Rodríguez-Barroso, E. Martínez-Cámara, M. V. Luzón, and F. Herrera. Dynamic defense against byzantine poisoning attacks in federated learning. *Future Generation Computer Systems*, 133:1–9, 2022.

O. A. Rosso, H. A. Larrondo, M. T. Martin, A. Plastino, and M. A. Fuentes. Distinguishing noise from chaos. *Physical Review Letters*, 99:154102, Oct 2007.

M. Sarhan, S. Layeghy, and M. Portmann. Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications*, 27(1):357–370, feb 2022.

A. Sarkar, T. Sen, and A. K. Roy. Grafehty: Graph neural network using federated learning for human activity recognition. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1124–1129, 2021. doi: 10.1109/ICMLA52953.2021.00184.

F. Sattler, K.-R. Müller, and W. Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, 2020.

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

I. Sharafaldin et al. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116, 2018.

V. Shejwalkar, A. Houmansadr, P. Kairouz, and D. Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1354–1371, 2022.

F. A. Silva, O. Orang, F. J. Erazo-Costa, P. C. Silva, P. H. Barros, R. P. Ferreira, and F. G. Guimarães. Time series classification using federated convolutional neural networks and image-based representations. *IEEE Access*, 2025a.

P. C. L. Silva, O. Orang, P. H. Barros, F. A. R. da Silva, H. S. Ramos, and F. G. Guimarães. Driver Maneuver Classification Based on Multivariate Fuzzy Time Series. In *Proceedings of the 2025 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2025b.

V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Federated multi-task learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4427–4437, 2017.

K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems 29*, pages 1857–1865. Curran Associates, Inc., 2016.

D. Song, G. Shen, D. Gao, L. Yang, X. Zhou, S. Pan, W. Lou, and F. Zhou. Fast heterogeneous federated learning with hybrid client selection. In R. J. Evans and I. Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 2006–2015, 31 Jul–04 Aug 2023.

K. Sozinov, V. Vlassov, and S. Girdzijauskas. Human activity recognition using federated learning. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing (ISPA/IUCC)*, pages 1103–1111, 2018. doi: 10.1109/BDCloud.2018.00164.

D. Su and Z. Qu. Detection ddos of attacks based on federated learning with digital twin network. In G. Memmi, B. Yang, L. Kong, T. Zhang, and M. Qiu, editors, *Knowledge Science, Engineering and Management*, pages 153–164, Cham, 2022.

G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, and J. Liu. Data poisoning attacks on federated machine learning. *IEEE Internet of Things Journal*, 9(13):11365–11375, 2022.

J. Sun, X. Wu, H. Huang, and A. Zhang. On the role of server momentum in federated learning, 2023. URL https://arxiv.org/abs/2312.12670.

J. L. Suárez, S. García, and F. Herrera. pydml: A python library for distance metric learning. *Journal of Machine Learning Research*, 21(96):1–7, 2020. URL http://jmlr.org/papers/v21/19-864.html.

A. Z. Tan, H. Yu, L. Cui, and Q. Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9587–9603, 2023a.

J. Tan, Y. Zhou, G. Liu, J. H. Wang, and S. Yu. pfedsim: Similarity-aware model aggregation towards personalized federated learning. *arXiv preprint arXiv:2305.15706*, 2023b.

Q. Tian, C. Guang, C. Wenchao, and W. Si. A lightweight residual networks framework for DDoS attack classification based on federated learning. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6, 2021.

J. Toldinas, A. Venčkauskas, A. Liutkevičius, and N. Morkevičius. Framing network flow for anomaly detection using image recognition and federated learning. *Electronics*, 11 (19), 2022.

J. E. Van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

R. Vogel, A. Bellet, and S. Clémençon. A probabilistic theory of supervised similarity learning for pointwise ROC curve optimization. In *International Conference on Machine Learning (ICML)*, pages 5065–5074, 2018.

F. Wang and C. Zhang. Feature extraction by maximizing the average neighborhood margin. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, USA, 2007. IEEE.

H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020.

J. Wang, X. Gao, Q. Wang, and Y. Li. Prodis-contshc: learning protein dissimilarity measures and hierarchical context coherently for protein-protein comparison in protein database retrieval. *BMC Bioinformatics*, 13(7):S2, May 2012.

X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *The Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a.

X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, USA, 2019b. IEEE.

Y. Wang, Y. Wang, J. Yang, and Z. Lin. A unified contrastive energy-based model for understanding the generative ability of adversarial training. In *International Conference on Learning Representations*, 2021.

K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 10:207–244, June 2009.

C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie. Fedattack: Effective and covert poisoning attack on federated recommendation via hard sampling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, page 4164–4172, New York, NY, USA, 2022.

L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu. Learning bregman distance functions for semi-supervised clustering. *IEEE Transactions on Knowledge and Data Engineering*, 24(3):478–491, 2012.

S. Xiang, F. Nie, and C. Zhang. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12):3600 – 3612, 2008. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2008.05.018.

Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao. A federated learning system with enhanced feature extraction for human activity recognition. *Knowledge-Based Systems*, 229:107338, 2021a.

Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao. A federated learning system with enhanced feature extraction for human activity recognition. *Knowledge-Based Systems*, 229:107338, 2021b.

J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, pages 478–487, 2016.

E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 521–528. MIT Press, 2003.

X. Yao and L. Sun. Continual local training for better initialization of federated models. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1736–1740. IEEE, 2020.

D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the International Conference on Machine Learning*, volume 80, pages 5650–5659, 2018.

Z. Yin, K. Li, and H. Bi. Trusted multi-domain ddos detection based on federated learning. *Sensors*, 22(20), 2022.

J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning (ICML)*, pages 12073–12086, 2021.

F. Yu, A. S. Rawat, A. Menon, and S. Kumar. Federated learning with only positive labels. In *International Conference on Machine Learning*, pages 10946–10956, 2020.

H. Yu, Z. Chen, X. Zhang, X. Chen, F. Zhuang, H. Xiong, and X. Cheng. Fedhar: Semi-supervised online learning for personalized federated human activity recognition. *IEEE Transactions on Mobile Computing*, 22(6):3318–3332, 2023.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2016.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, feb 2021a.

J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu. Poisoning attack in federated learning using generative adversarial nets. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 374–380, 2019.

J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal*, 8(5):3310–3322, 2021b.

J. Zhang, Y. Wu, and R. Pan. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In *Proceedings of the Web Conference 2021*, pages 947–956, 2021c.

J. Zhang, P. Yu, L. Qi, S. Liu, H. Zhang, and J. Zhang. FLDDoS: DDoS attack detection model based on federated learning. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 635–642, 2021d.

J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu. Federated learning with label distribution skew via logits calibration. In *International Conference on Machine Learning ICML*, volume 162, pages 26311–26329, 17–23 Jul 2022a.

X. Zhang, Y. Li, W. Li, K. Guo, and Y. Shao. Personalized federated learning via variational bayesian inference. In *International Conference on Machine Learning*, pages 26293–26310. PMLR, 2022b.

M. Zhao, B. An, W. Gao, and T. Zhang. Efficient label contamination attacks against black-box learning models. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3945–3951, 2017.

Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu. Multi-task network anomaly detection using federated learning. In *Proceedings of the 10th International Symposium on Information and Communication Technology*, SoICT '19, page 273–279, 2019.

J. Zhu, X. Ma, and M. B. Blaschko. Confidence-aware personalized federated learning via variational expectation maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24542–24551, June 2023.

Appendix A

A New Similarity Space Tailored for Supervised Deep Metric Learning

A.1 Introduction

A *distance metric* is a function that provides a way to measure how far apart two elements of a set are from each other. Among various works involving machine learning applications, the most commonly used metric is the Euclidean distance [De Maesschalck et al., 2000]. Methods that use Euclidean distance usually consider that all variables' covariance is zero, i.e., there is no correlation among them, but this assumption is hardly found in the real world [Xiang et al., 2008]. Euclidean distance and cosine similarity are popular for many applications. For instance, the cosine similarity is vastly used for text mining [Lin et al., 2014]. Even showing its effectiveness in several applications, the cosine similarity assumes equal weight for every dimension, limiting its application [Lin et al., 2014].

Euclidean and cosine distance are known as data-independent techniques once they are defined without prior knowledge about the data. Learning distances, a.k.a Metric Learning (MeL), from data is a common attempt to improve machine learning approaches [Weinberger and Saul, 2009, Deudon, 2018, Inaba et al., 2019]. In modern machine learning research, MeL is a fundamental technique for several different applications such as sorting [McFee and Lanckriet, 2010], classification (e.g., k-nearest neighbors), clustering [Wu et al., 2012], and ranking [Vogel et al., 2018].

MeL aims to estimate distance function parameters based on a given training set. A common approach is to frame MeL as a convex optimization problem [Xing et al., 2003]. Thus, a distance d can be defined as $d_{\boldsymbol{M}}(x, y) = \sqrt{(x-y)^T \boldsymbol{M}(x-y)}$, in which $\boldsymbol{M}$ is a positive semi-definite matrix. In the case $\boldsymbol{M}$ is the covariance matrix, we have the *Mahalanobis* distance [De Maesschalck et al., 2000]. Classic methods proposed for metric learning use $d_{\boldsymbol{M}}$ to search for the best linear space that captures the semantics of the data (e.g., in a classification setting, we search for $\boldsymbol{M}$ that minimizes the miss-

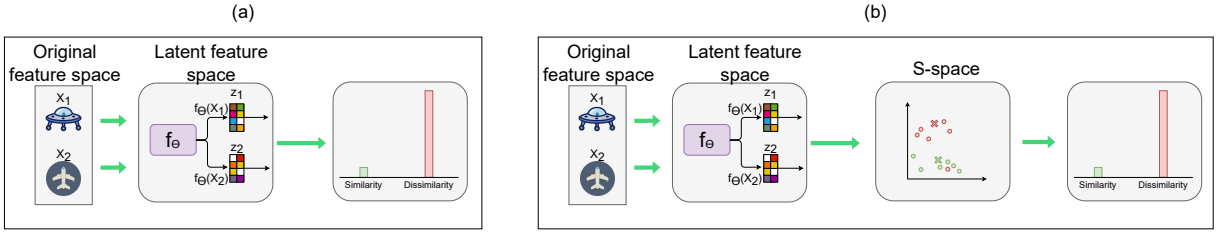


Figure A.1: Comparison between the canonical model of DMeL and the model used in this work : (a) Canonical scheme of DMeL; (b) Our scheme of DMeL with S-space.

classification loss). However, the linear transformation has some limitations, as it cannot model high-order correlations between the original data dimensions [Cao et al., 2019]. Using MeL, we can define metrics that consider the covariance of attributes. Additionally, MeL approaches do not necessarily assume linear relationships, although classical MeL techniques like the *Mahalanobis* distance [De Maesschalck et al., 2000] assume a linear space. Moreover, MeL does not assume equal weights for every attribute [Lin et al., 2014]. The assumption that MeL can be treated as a convex optimization problem can also be relaxed using the appropriate model.

Deep learning techniques are currently being used for MeL to tackle the issues mentioned above [Hou et al., 2019, Mao et al., 2019]. Since these proposals seek to learn a non-linear feature representation, they usually overperform standard techniques in the literature. Neural Networks (NNs) are natural candidates and are typically used to learn similarity metrics [Chopra et al., 2005, Hadsell et al., 2006].

The representation of compressed data found by a Neural Network (NN) is commonly named latent feature space, and the data in this space is latent data, as we can see in Figure A.1a. Our work hypothesizes that the latent feature space captured by NNs can be improved with an auxiliary space. For instance, typical NNs-based Deep Metric Learning (DMeL) approaches extract a latent space that encodes similar and dissimilar *points*, but not the separability among them. However, this single representation is limited, as it does not capture *pairwise information*.

Unlike the literature, our approach employs NNs, fed by labeled original pairwise data, to find a *latent pairwise space with markers*. This approach is shown in Figure A.1b as we now detail. In our method, data comes in pairs of vectors $(\mathbf{x}_i, \mathbf{x}_j)$ which are deemed as similar ($\boldsymbol{x}_i$ has the same label as $\boldsymbol{x}_j$) or dissimilar ($\boldsymbol{x}_i$ has the different label as $\boldsymbol{x}_j$). The first part of our architecture is an autoencoder. After encoding the pair of input objects, our major novelty is on converting *data pairs* to a new Similarity-space (called S-space). A data point $\mathbf{x}_i$ is mapped into $\mathbf{z}_i = f_\Theta(\mathbf{x}_i)$ in the latent space, where Θ are model parameters. The S-Space for a pair of points i and j is composed of two novel ideas. Firstly, we represent points as a similarity vector between pairs, i.e., $\mathbf{s}_{ij} = |\mathbf{z}_i - \mathbf{z}_j|$. Secondly, *and more importantly*, we define markers that act as reference points to similar ($\boldsymbol{\mu}_p^+ \in \mathcal{M}^+$) and dissimilar ($\boldsymbol{\mu}_n^- \in \mathcal{M}^-$) regions. Markers' positions are learned in the

optimization process.

Our loss function is comprised of three parts. Firstly, an autoencoder loss function takes care of data encoding and decoding. The second loss function captures the sum of distances between similarity vectors ($\mathbf{s}_{ij}$) and markers ($\boldsymbol{\mu}_m \in \mathcal{M}^+ \cup \mathcal{M}^-$), in this work, we used a kernel-based discrete Cauchy distribution to estimate this distance, and we apply a cross-entropy loss function between the input labels and the model output. The last part of our loss function is called a repulsive regularizer. It is inversely proportional to the distance of the markers of the same class. This loss function ensures that markers are different (the loss increases as markers become similar), ensuring some diversity level on the marker set. It attempts that markers capture complex similarity regions such as disjoint similarity/dissimilarity regions.

We named our approach as *Supervised distance MEtric Learning encoder with simiLarity space* (SMELL). Our method is herein described as supervised learning, but it can be appropriately extended to unsupervisedand semi-supervised learning with some pseudo-label approaches [Chen et al., 2020, Dizaji et al., 2017]. For example, in pseudo-label [Chen et al., 2020], the labeled data are categorized via k-means clustering, and patterns' data are sampled to train multimodal. In addition, for unlabeled data, if multimodal reach an agreement on predicting a sample, this sample is labeled; otherwise, keep it unlabeled. Through a wide range of experiments on 28 datasets, we show that SMELL provides gains over the state-of-the-art in all of them. To explain its accuracy, we show evidence supporting the following two hypotheses.

Hypothesis A.1. *(**HA.1**) SMELL groups data points considered similar (in our context, which have the same labels) and dissimilar (different labels) into disjoint regions in S-space.*

Hypothesis A.2. *(**HA.2**) SMELL increases the input pairs' separability in the latent feature space for different pairs (similar/dissimilar) types.*

Overall, the main contributions of our work are:

(i) a new data representation space called *Similarity space* (S-space) that separates regions where similar/dissimilar objects lie together and help the convergence of the model. We also investigate interpretability and data visualization in this space. S-space can capture complex regions that can model similar points in disjoint regions;

(ii) a new distance metric learning method that simultaneously learns a latent representation of the data and the markers' position in the S-space;

(iii) we found evidence that the number of markers is a virtual hyperparameter of the model and does not need to be tuned.

(iv) a new regularization function to avoid model overfitting called *repulsive regularizer*.

Table A.1: Notation used in this article.

Notation	Description
$\mathcal{X}$	input data examples set
$\boldsymbol{x}_i$	m-dimensional single element in $\mathcal{X}$
$\mathcal{Y}$	Label set for set $\mathcal{X}$
y_i	single element in $\mathcal{Y}$
$\mathcal{Z}$	Latent Feature Space from $\mathcal{X}$
$\boldsymbol{z}_i$	n-dimensional single element in $\mathcal{Z}$
f_Θ	Encoder function
Θ	set of weights for encoder
$f_{\Theta'}$	Decoder function
Θ'	set of weights for decoder
l	label function for a element in set $\mathcal{X}$
l'	label function for a element in set $\mathcal{Z}$
$\mathcal{S}$	The *similarity space* from $\mathcal{Z}$
$\boldsymbol{s}_{ij}$	n-dimensional single element in $\mathcal{S}$
$\mathcal{M}$	The *markers set*
μ_i	n-dimensional single element in $\mathcal{M}$
f^S	Function that maps a pair in $\mathcal{X}$ to an element in $\mathcal{S}$
ψ	The similarity function
Σ	Set of parameters of ψ (Θ, Θ' and $\mathcal{M}$)

A.2 Background and notation

In SMELL, we map pairwise input data into a latent space and a Similarity space. In this section, we provide some technical background about data representation with autoencoders and a mathematical notation essential to the proposed method understanding.

Throughout the chapter, we apply the following notation. We denote vectors by boldface lowercase letters, such as $\boldsymbol{x}$, $\boldsymbol{z}$ and $\boldsymbol{\mu}$; all scalars by lowercase letters, such as m and n; sets of parameters by greek uppercase letters, such as Θ and Σ; and sets by calligraphic uppercase letters, such as $\mathcal{X}$ and $\mathcal{Z}$. The zero-mean normal distribution will be denoted by $\mathcal{N}(\mu = 0, \sigma)$. Table A.1 summarizes this notation.

Let the set $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^v$, with $\boldsymbol{x}_i \in \mathbb{R}^m$, be v data examples defined in an m-dimensional feature space. For each $\boldsymbol{x}_i \in \mathcal{X}$ there is an associated label $y_i \in \mathcal{Y} = \{y_i\}_{i=1}^v$, where $y_i \in \{1, ..., b\}$. In this way, the pair $(\boldsymbol{x}_i, y_i)$ indicates which of b classes a input $\boldsymbol{x}_i$ belongs to. In a supervised Machine Learning classification problem, we seek to find a function $l : \mathcal{X} \to \mathcal{Y}$ that maps an unlabeled example $\boldsymbol{x}_i$ into their respective label y_i. To develop the proposed work, we introduce here some important definitions:

Definition A.1. *(The latent feature space) Consider the set $\mathcal{X}$ as the original feature space and the representation function $f_\Theta : \mathcal{X} \to \mathcal{Z}$, in which $f_\Theta(\boldsymbol{x}_i) = \boldsymbol{z}_i \implies l(\boldsymbol{x}_i) = l'(\boldsymbol{z}_i) = y_i$ and the function $l' : \mathcal{Z} \to \mathcal{Y}$, which maps the latent data into their respective labels. We defined the representation space $\mathcal{Z}$ called latent feature space from $\mathcal{X}$ as $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^v$, with $\boldsymbol{z}_i \in \mathbb{R}^n$.*

An autoencoder is a Neural Network trained to attempt to copy a data input to its output. It can be seen as consisting of two parts: an encoder and a decoder that produces an input-based reconstruction.

An encoder is a representation learning algorithm that seeks to find a representation function $f_\Theta : \mathcal{X} \to \mathcal{Z}$ for a set of weights Θ that maps the set $\mathcal{X}$ to the *latent feature space* $\mathcal{Z}$.Similarly, the decoder function can be defined as the inverse encoder function $f_{\Theta'}^{-1} : \mathcal{Z} \to \mathcal{X}$ where Θ' is a set of weights for the decoder. Autoencoders are trained to minimize reconstruction errors (typically, Mean Squared Errors - MSE), and their training is performed through *Backpropagation* of the error, just like a regular Feedforward Neural Network.

A Siamese neural network model [Bromley et al., 1994, Wang et al., 2019b] receives a pair of input examples $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{X} \times \mathcal{X}$ and transform each of them to a latent data $(\boldsymbol{z}_i, \boldsymbol{z}_j) \in \mathcal{Z} \times \mathcal{Z}$ through the encoder f_Θ. In the context of supervised learning, for a data pairwise $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{X} \times \mathcal{X}$, we say they are similar iff $l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j)$. Analogously, they are dissimilar iff $l(\boldsymbol{x}_i) \neq l(\boldsymbol{x}_j)$.

Definition A.2. *(The similarity space) The representation space called Similarity space (or S-space) is a space built from the set $\mathcal{X} \times \mathcal{X}$. So, let f^S be the function $f^S : \mathcal{X} \times \mathcal{X} \to \mathcal{S}$, the similarity space is defined as $\mathcal{S} = \{\boldsymbol{s}_{ij}\}$, with $\boldsymbol{s}_{ij} \in S \subset \mathbb{R}^n$, where $\boldsymbol{s}_{ij}$ represents the similarity vector if $l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j)$, and if $l(\boldsymbol{x}_i) \neq l(\boldsymbol{x}_j)$, then $\boldsymbol{s}_{ij}$ represents the dissimilarity vector.*

In this chaper, we define the map function $f^S : \mathcal{X} \times \mathcal{X} \to \mathcal{S}$ for a pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ by the following element-wise absolute value operation:

$$\boldsymbol{s}_{ij} = f^S(\boldsymbol{x}_i, \boldsymbol{x}_j) = |f_\Theta(\boldsymbol{x}_i) - f_\Theta(\boldsymbol{x}_j)| = |\boldsymbol{z}_i - \boldsymbol{z}_j| = (|z_i^1 - z_j^1|, |z_i^2 - z_j^2|, \cdots, |z_i^n - z_j^n|), \tag{A.1}$$

it is worth noting that since $\boldsymbol{s}_{ij}$ is obtained by an element-wise process, it has the same dimension as $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, where z_i^n is the n-th feature of the i-th data example in a latent space representation $\mathcal{Z}$ (see Definition A.1). In *S-Space*, we define similarity and dissimilarity markers to group $\boldsymbol{s}_{ij}$ vectors into their respective representations. The markers are vectors in the S-space, which have their positions estimated by SMELL, i.e., in training the model, we estimate the neural network's weights and the markers' position. Thus, the closer $\boldsymbol{s}_{ij}$ is to the similarity/dissimilarity marker, the greater the probability that the $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is similar/dissimilar.

Definition A.3. *(The Markers set) In S-space, we defined the markers set $\mathcal{M} \subset \mathbb{R}^n$ (same space as $\mathcal{S}$) to improve similarity calculations. We define the set $\mathcal{M}^+$ representing the set of markers responsible for quantifying the similarity between the input pairs. Likewise,*

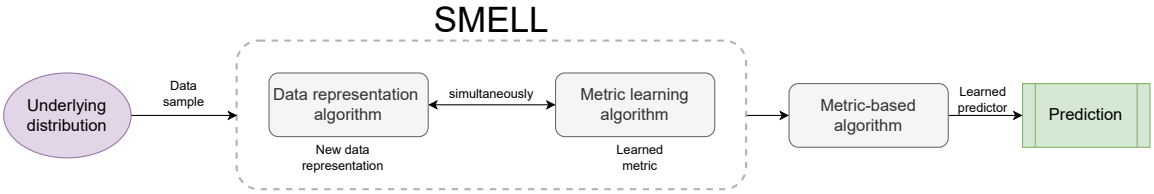


Figure A.2: Simple black box schematic for our proposal.

markers in set $\mathcal{M}^-$ *quantify the dissimilarity. The Markers set is defined as* $\mathcal{M} = \mathcal{M}^+ \cup \mathcal{M}^- = \{\boldsymbol{\mu}_i^+\}_{i=1}^k \cup \{\boldsymbol{\mu}_j^-\}_{j=k+1}^w$.

Therefore, in this work, we seek to calculate the similarity function $\psi_\Sigma : \mathcal{X} \times \mathcal{X} \to [0, 1]$. The parameters of ψ are defined by the set $\Sigma = \{\Theta, \Theta', \mathcal{M}\}$, respectively the weights of the encoder, decoder, and the Markers set in S-space. SMELL relies upon simultaneously learning all elements of Σ.

A.3 Supervised distance MEtric Learning encoder with simiLarity space (SMELL)

Our proposal, SMELL, simultaneously optimizes a latent data representation (using a DMeL model) and a similarity function that indicates the similarity of two objects in the learned data S-space. This technique can be useful for various applications, such as feeding a predictor (e.g., a classifier) with a new metric learned from the data. This section details our proposal. Figure A.2 shows a simple schematic for our proposal.

A.3.1 Metric learning algorithm

There are several ways to find a similarity metric ψ_Σ [Wang et al., 2012, Ahmed et al., 2015]. In this chapter, we propose ψ_Σ being estimated from the latent representation obtained by the encoder $f_\Theta : \mathcal{X} \to \mathcal{Z}$.

As seen in Definition A.2, we define a new representation space, S-space $\mathcal{S}$, which quantifies the similarity between pairs of objects. In Equation A.1, we propose a map function $f^S : \mathcal{X} \times \mathcal{X} \to \mathcal{S}$ for a data pairwise $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as being an element-wise absolute value operation representing the pairwise difference between the pair of data. Note, in

Equation A.1, that $\boldsymbol{s}_{ij} \in \mathbb{R}^n$ (same dimension as *latent representation space*).

Regarding the pairwise labeling, we have two options for a given pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$: similar or dissimilar.

Thus, we define the *Markers set* $\mathcal{M}$ so that each marker of $\mathcal{M}^+$ or $\mathcal{M}^-$ represents one of these possibilities (see Definition A.3). The closer the vector $\boldsymbol{s}_{ij}$ is to a marker $\boldsymbol{\mu}^+ \in \mathcal{M}^+$ or $\boldsymbol{\mu}^- \in \mathcal{M}^-$, the greater the probability that the elements of the pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are similar or dissimilar to each other, respectively. Then, we have, in this case, k similarity markers and $w - k$ dissimilarity markers for $\mathcal{M}$, and $\mathcal{M}^+ \cap \mathcal{M}^- = \emptyset$.

Inspired by [Maaten and Hinton, 2008, Xie et al., 2016, Li et al., 2018] we use the Student's t-distribution with one degree of freedom (Cauchy distribution) as a kernel to measure the similarity between $\boldsymbol{s}_{ij}$ and a specific marker $\boldsymbol{\mu}_m \in \mathcal{M}$, as

$$q_{ij}^m = \frac{(1 + ||\boldsymbol{s}_{ij} - \boldsymbol{\mu}_m||_2^2)^{-1}}{\sum_{\mu_{m'} \in \mathcal{M}} (1 + ||\boldsymbol{s}_{ij} - \boldsymbol{\mu}_{m'}||_2^2)^{-1}}, \tag{A.2}$$

where $q_{ij}^m \in \mathbb{R}$ is the similarity/dissimilarity of $\boldsymbol{s}_{ij}$ in relation to the markers μ_m (it is normalized by the sum of all markers in $\mathcal{M}$). So, we calculate $q_{ij}^+ = \sum_p q_{ij}^p$ for all $\boldsymbol{\mu}_p \in \mathcal{M}^+$ and $q_{ij}^- = \sum_n q_{ij}^n$ for all $\boldsymbol{\mu}_n \in \mathcal{M}^-$. In other words, q_{ij}^+ is the probability of $\boldsymbol{x}_i$ have the same label as $\boldsymbol{x}_j$ and q_{ij}^- is the probability of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have different labels. Since $\mathcal{M}^+$ and $\mathcal{M}^-$ are two disjoint sets, we have $q_{ij}^+ + q_{ij}^- = 1$.

It is worth noting that we use a different version of the Deep Metric Learning canonical model. Thus, we use the representation of the difference vector $\boldsymbol{s}_{ij}$ defined in S-pace. In Section A.4.2 we show more details about this choice. Finally, As shown in equation A.2, we conduct an iterative sum after obtaining data' embeddings (in S-space). So, our proposal adds the time complexity $O(w)$, where d is the latent feature space dimension (typically $d < 1024$, and w is the number of markers (typically $w < 8$). Furthermore, $O(dw) \ll O(N)$, where $O(N)$ is the complexity of the neural network training (encoder function).

A.3.2 Loss function

SMELL relies on simultaneously learning a latent representation of the data (with parameters Θ and Θ' for the encoder and decoder functions, respectively) and the positioning of the markers of the set $\mathcal{M}$ in S-space. Therefore, finding the parameters $\Sigma = \{\Theta, \Theta', \mathcal{M}\}$ of the function $\psi_\Sigma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is defined as an optimization problem. Our loss function is composed of the weighted sum of three loss functions. Let our loss function be j, we estimate the optimal parameters set Σ^* with Cross-entropy loss H_c (first loss

function). We define a regularization functions R_r (second loss function) and R_d (third loss function) to avoid overfitting in the training process. In training, cross-entropy is applied between the output of SMELL and the object's classes.

As the cross-entropy loss focuses only on improving the object's separability in the S-space, a loss function composed by the cross-entropy loss alone may lead the S-space to distort the *latent feature space*, weakening the representativeness of embedded features. As shown in Peng et al. [2016], autoencoders can preserve the local structure of underlying data distribution. So, similarly to Dizaji et al. [2017], R_r regards the autoencoder's reconstruction error. In our proposal, for all training pairs $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and for all reconstructed pairs $(\boldsymbol{x}'_i, \boldsymbol{x}'_j)$ we have $R_r(\boldsymbol{x}_i, \boldsymbol{x}_j) = r_r \left[||\boldsymbol{x}_i - \boldsymbol{x}'_i||_2^2 + ||\boldsymbol{x}_j - \boldsymbol{x}'_j||_2^2\right]$, where r_r is a constant to calibrate the loss reconstruction function, and N is the number of pairs in training the dataset. Under this condition, fine-tuning SMELL will not cause corruption in *latent feature space*.

When we use more than one maker as reference points to the similarity/dissimilarity regions, markers of the same set $\mathcal{M}^+$ (or $\mathcal{M}^-$) tend to group altogether, hindering the efficiency of our method. In this context, we propose a new regularization term R_d we called *Repulsive Regularizer* to avoid this undesirable behavior. It is defined as

$$R_d^+ = \frac{1}{c^+}\left[\sum_{\mu_i \in \boldsymbol{M}^+}\sum_{\mu_j \in \boldsymbol{M}^+} \frac{1}{||\mu_i - \mu_j||_2^2 + \epsilon}\right], \tag{A.3}$$

where $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}_j$ and c^+ is a constant value defined as $c^+ = \binom{k}{2}$, in which k is the number of elements in $\mathcal{M}^+$ (see Definition A.3). R_d is inversely proportional to the square distance of the markers. To avoid ill-formed problems, we added a corrective term ϵ to the denominator that prevents division by 0. We conducted a manual investigation with grid search and adopted $\epsilon = 10^{-1}$ for our experiments. In the same way, we define R_d^-, and with that, we have $R_d = r_d \left[R_d^+ + R_d^-\right]$,with a constant value r_d for calibration. Note that $R_d^+ = 0$ if we have a single positive marker $k = 1$. In the same way, if we have a single negative marker, $R_d^- = 0$ if $w - k = 1$.

Let $\mathcal{Q} = \{q_{ij}\}$, the SMELL output, be the set that contains the pairs $q_{ij} = (q_{ij}^+, q_{ij}^-)$ corresponding to the probability of the elements of a pairwise input $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are similar or dissimilar, respectively. The optimal hyperparameters set can be defined as $\Sigma^* = \arg\min_{\boldsymbol{\Sigma}} j$, where

$$j = R_d + \sum_{i \in \mathcal{X}}\sum_{j \in \mathcal{X}} J(\boldsymbol{x}_i, \boldsymbol{x}_j) = R_d + \sum_{i \in \mathcal{X}}\sum_{j \in \mathcal{X}} \left[r_{HC} H_c(\boldsymbol{u}_{ij}, \boldsymbol{q}_{ij}) + R_r(\boldsymbol{x}_i, \boldsymbol{x}_j)\right], \tag{A.4}$$

where r_{HC} is a constant for calibration, $H_c(\boldsymbol{u}_{ij}, \boldsymbol{q}_{ij})$ is the binary cross-entropy loss and $\boldsymbol{u}_{ij} \in \mathcal{U}$ is defined as $\boldsymbol{u}_{ij} = (1, 0)$ if i has same label as j and $\boldsymbol{u}_{ij} = (0, 1)$, otherwise.

SMELL simultaneously learns all parameters in the set Σ^*. The representation found in S-space aims at grouping the elements $\boldsymbol{s}_{ij}$ around their respective markers, as

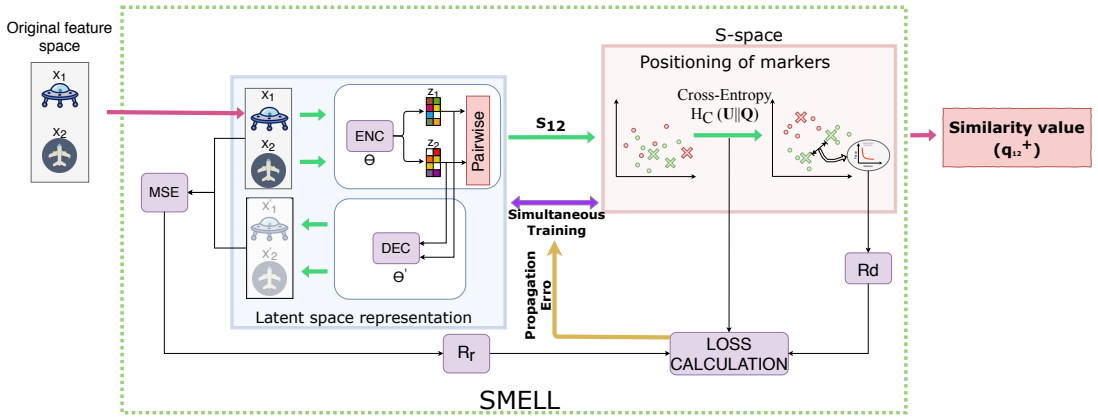


Figure A.3: The left side represents the Encoder with reconstruction, and the right side represents the optimization process for the markers' position for $\mathcal{M} = \{\boldsymbol{\mu}_1^+, \boldsymbol{\mu}_2^+, \boldsymbol{\mu}_3^-\}$. In this example, we used two positive and one negative marker. Green and red crosses represent $\boldsymbol{\mu}_1^+$, $\boldsymbol{\mu}_2^+$, and $\boldsymbol{\mu}_3^-$, respectively, green and red dots represent the similar and dissimilar input pairs. The rightmost green arrow represents the markers' position optimization step by using Cross-Entropy divergence and some regularization functions. Observe that the number of positive and negative markers are hyperparameters.

defined in Loss Function j (Equation A.4). According to the similarity function, the closer the $\boldsymbol{s}_{ij}$ is to the similarity/dissimilarity marker, the greater the probability that the S-vector is similar/dissimilar. So, to minimize the loss function H_c, SMELL tends to approximate the objects in the *S-Space* around their respective markers as well as reposition the markers according to S-vectors position, i.e., S-vectors are grouped close to the similarity markers; we also observe an analogous behavior to dissimilar S-vectors.

The impact of the attractive behavior is controlled by the constant r_{HC}, i.e., the higher the r_{HC}, the greater is the tendency to group the points $\boldsymbol{s}_{ij}$ closer to the respective markers. Also, note that the regularization functions operate in different spaces, i.e. R_r operate in *latent feature space*, R_d operates in *S-space* and H_c operates in *latent feature space* and *S-space* simultaneously.

Figure A.3 depicts the more detailed schematic of our proposal using a toy example (two positive markers and one negative). Observe that the number of positive and negative markers is a hyperparameter.

A.3.3 Optimization

To find the Σ^* set, we use mini-batch stochastic gradient descent (SGD) and back-propagation. First, we note that the decoder weights Θ' are only affected by the R_r component of the loss function J. So, we can use $\partial R_r / \partial \Theta'$ to update Θ'. Then, given a

mini-batch $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{B}$ with g samples and learning rate λ, Θ' is updated by

$$\Theta' = \Theta' - \frac{\lambda}{g} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{B}} \frac{\partial R_r(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \Theta'}.$$

To optimize the markers, consider that

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_t - \frac{\lambda}{g} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{B}} \frac{\partial J(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\mu}_t} = \boldsymbol{\mu}_t - \frac{\lambda}{g} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{B}} \left(\frac{\partial H_C(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\partial \boldsymbol{\mu}_t} \right) + \frac{\partial R_d}{\partial \boldsymbol{\mu}_t},$$

where $\partial H_C(\boldsymbol{x}_i, \boldsymbol{x}_j)/\partial \boldsymbol{\mu}_t$ can be calculated (as can see in Li et al. [2018]) for a given $\boldsymbol{\mu}_t$ and $\boldsymbol{s}_{ij}$ as

$$\frac{\partial H_C(x_i, x_j)}{\partial \boldsymbol{\mu}_t} = 2 \frac{(\boldsymbol{q}_{ij}^t - \boldsymbol{u}_{ij})(\boldsymbol{s}_{ij} - \boldsymbol{\mu}_t)}{1 + ||\boldsymbol{s}_{ij} - \boldsymbol{\mu}_t||_2^2}$$

and,

$$\frac{\partial R_d}{\partial \boldsymbol{\mu}_t} = -2 \sum_{\boldsymbol{\mu}_s \in M} \left[\text{sign}(\boldsymbol{\mu}_s) \frac{||\boldsymbol{\mu}_t - \boldsymbol{\mu}_s||_2}{(||\boldsymbol{\mu}_t - \boldsymbol{\mu}_s||_2^2 + \epsilon)^2} \right],$$

where $\text{sign}(\boldsymbol{\mu}_s) = 1$ if $\boldsymbol{\mu}_s \neq \boldsymbol{\mu}_t$ and $\boldsymbol{\mu}_s$ has same semantic (similarity or dissimilarity) than $\boldsymbol{\mu}_t$, and $\text{sign}(\boldsymbol{\mu}_s) = 0$, otherwise. For training SMELL, we randomly selected the mini-batch with m pairs of elements (half are similar, and the other half are dissimilar). Also, our proposal does not have any specific batch selection criteria, unlike some specialized techniques in the literature [Schroff et al., 2015, Sohn, 2016].

A.4 Results and discussion

In this section, we present the results of the SMELL's assessment. We also discuss the interpretability of the similarity space (S-space) and conduct a performance evaluation comparing SMELL with three distance metric learning approaches from pyDML [Suárez et al., 2020][1]: ANMM [Wang and Zhang, 2007], NCA [Goldberger et al., 2005] and KDMLMJ [Nguyen et al., 2017]; five deep metric learning approaches: Contrastive [Hadsell et al., 2006], Triplet [Schroff et al., 2015], NPair [Sohn, 2016], FastAP [Cakir et al., 2019] and MSLoss [Wang et al., 2019b]; and Euclidean distance.

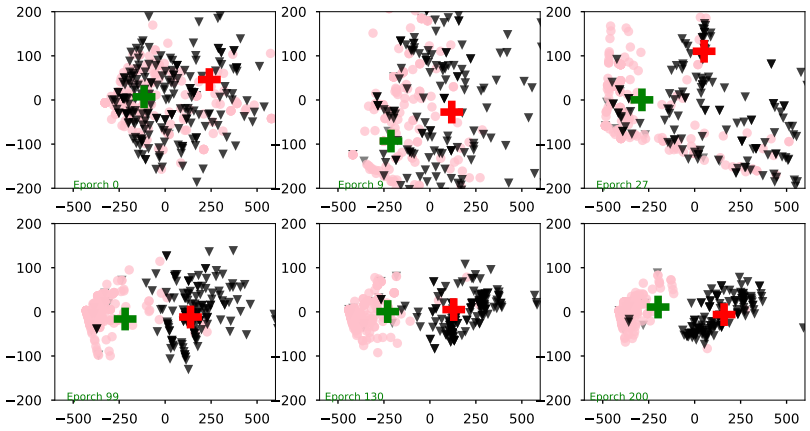
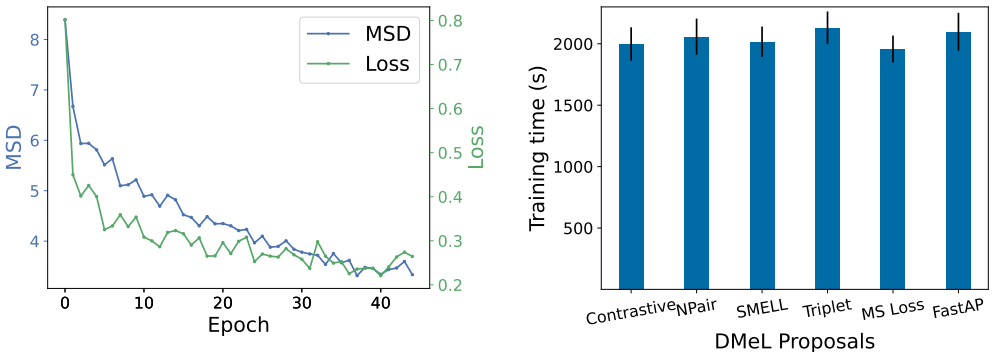


Figure A.4: Simultaneous training of μ^+ (green) and μ^- (red) markers' position and data representation in S-space for some training epochs.



(a) Loss and Mean Squared Distance. (b) Training time.

Figure A.5: FASHION-MNIST train: (a) Loss and Mean Squared Distance (between S-vector and marker) for each train epoch, and (b) DMeL proposals training time for 10 rounds.

A.4.1 Behavior analysis

Our proposal is based on optimizing the parameter set $\boldsymbol{\Sigma} = \{\Theta, \Theta', \boldsymbol{M}\}$ using markers (with a Cauchy kernel). SMELL learns a representation of input pairs that groups the points with similar and dissimilar labels around their respective markers. We can observe this behavior in Figure A.4. In this figure, the input pairs of similar and dissimilar labels are represented by pink circles and gray triangles. In addition, $\boldsymbol{\mu}^+$ and $\boldsymbol{\mu}^-$ markers are represented by green and red crosses, respectively. We plot some $\boldsymbol{s}_{ij}$ vectors for input pairs $(\boldsymbol{x}_i,\ \boldsymbol{x}_y)$ of the test set for the Balance dataset. After training the autoencoder, a two-dimensional plot was created with the aid of PCA before (first figure)

[1]https://pydml.readthedocs.io/en/latest/index.html

and after (the other figures) the optimization process.

The points do not present a well-defined cluster structure in the first plot in Figure A.4 (before adjusting the markers' positions). This behavior changes when we analyze the last plot in Figure A.4 (after adjusting the markers). In the last plot in Figure A.4, we can see well-defined groups around the markers. Moreover, by comparing the scale of the Figures A.4, we see that in the last case, points are more spaced, i.e., our proposal tends to group points around their respective markers. This behavior corroborates our initial hypothesis described in (**HA.1**).

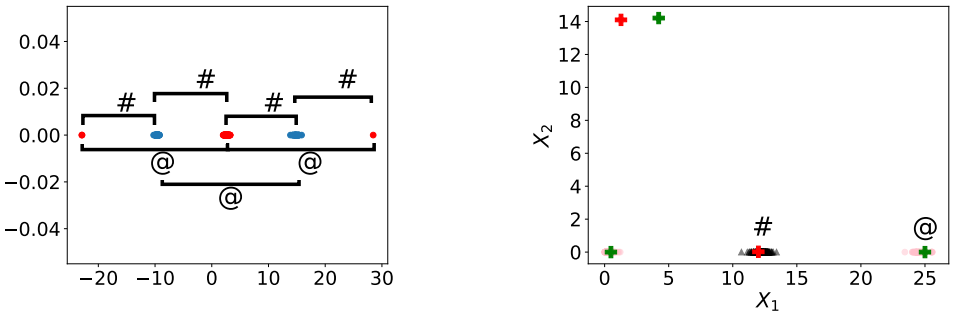
We observed that our proposal acts as an attractive potential. In this sense, the marker "pulls" the favorable points (the similarity mark "pulls" similar points). Therefore, their movement resembles a Group Mobility Model [Hong et al., 1999], i.e., the marker is being positioned, and the points go "following" the leader as a "caravan" of nomads. At the same time, the markers tend to repulse themselves.

To quantify the clustering of vectors around a reference point (e.g., centroid), we can use the Mean Squared Distance (MSD). So, we calculate the MSD during the training of our proposal for each S-vector to its respective marker, i.e., the euclidian distance between similar/dissimilar s_{ij} and similarity/dissimilarity marker. In addition, we also calculate loss function j, as shown in Figure A.5a. We observed that as loss decreases, the MSD also decreases, indicating that, during training, our model groups s_{ij} close to their respective markers. This can be seen as such an intense attracting field, which locks the movement dynamics of the points closest to the markers.

Figure A.5b depicts the training duration of the DMeL proposals in Fashion-MNIST dataset (63000 train samples). DMeL has similar training times because the encoder function (neural network) training is the primary time bottleneck. Regarding the prediction's time, similarity metric extraction proposals use based-metrics algorithms (e.g., KNN) to perform inference, as we can see in Schroff et al. [2015], Sohn [2016], Cakir et al. [2019]. Thus, some heuristics can improve the proposal's prediction time effectiveness, as seen in Schroff et al. [2015]. For example, the time prediction for the Fashion-MNIST dataset (7000 test samples) is 17.39 s.

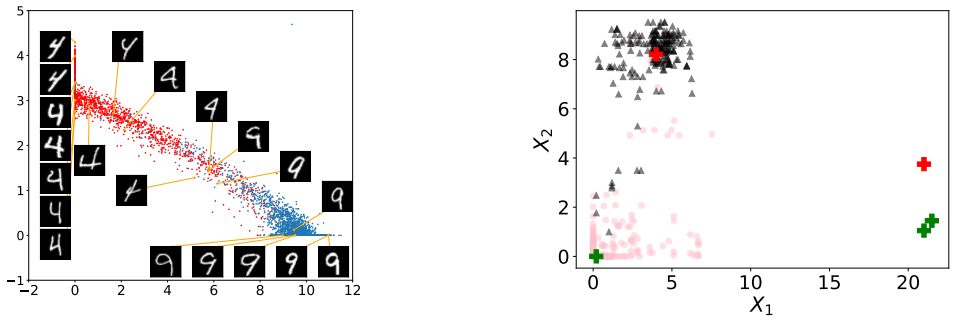
A.4.2 Latent space and S-space analysis

For a better understanding of the latent space found by SMELL, we analyzed the behavior of our proposal using the sonar and MNIST datasets as shown in Figure A.7 and A.6. For the sake of visualization, in this analysis, we use the setup discussed in Section 2.2 with $n = 2$ (latent dimension). Figures A.6a and A.7a show the *latent feature*



(a) Latent space ($n = 2$) (b) S-space ($n = 2$)

Figure A.6: Sonar Datasets: Latent Space and S-space Analysis for SMELL.



(a) Latent space ($n = 2$) (b) S-space ($n = 2$)

Figure A.7: MNIST Datasets: Latent Space and S-space Analysis for SMELL.

space (output of encoder). Observe that in these figures, points represent individual objects. Red and blue points represent different classes. There are two classes in the sonar dataset, and we show only two classes of MNIST (handcraft digits 4 and 9). These *latent feature spaces* result from the joint optimization process of the autoencoder and the S-space.

In Figure A.6a, we observe that red points are grouped in different regions far apart at a distance approximately constant, denoted by @. Similarly, blue points are apart at a distance approximately constant, @. Different clusters are apart at a distance approximately constant, denoted as #. Figure A.6b and A.7b show a random sample of 400 data pairs from the sonar dataset mapped to S-space (200 similar and 200 dissimilar pairs). In S-space, points represent a pair of objects. Pink circles and black triangles represent similar and dissimilar labels, respectively. Also, similarity and dissimilarity markers are represented by green and red crosses, respectively.

Figure A.6b show some clustered regions. The region grouped by the similarity marker (closer to the origin) is responsible for grouping elements of similar classes with a distance closer to 0. This result corroborates with Proposition 3.1. The same behavior is found in Figure A.7b, where we observe a green cross close to the origin.

However, in Figure A.6b, we observe some similar objects mapped to points with a distance close to @, instead of zero. The green cross at @ is responsible for creating

the similarity region that represents this situation. Depending on the data complexity, other regions of similarity and/or dissimilarity can occur and are represented by other green/red crosses. Dissimilarity regions are depicted as #. Therefore, in the space found by SMELL, we see the behavior of multiple groups, separated by distances determined by the similarity/dissimilarity markers (labels # and @).

Observe in Figure A.7a the soft transition from digit 4 to 9, which shows that the S-space preserves the connection between these two similar digits. This effect is captured even though we do not use any data-specific feature extractor, such as convolution layers. In SMELL, the encoder can be switched by any feature extractor (e.g. vision transformer) tailored explicitly for the input data.

In Figure A.7a, we observe that the handcrafted digits four are grouped (on the left). Even in this group, the similarities between the digits remain. The first two digits in the top-left region correspond to numbers with thicker writing and slightly rotated, and as we go down in the latent space, the digit's shape starts to thinner. This behavior indicates a gradient that represents the thickness of the object. This same behavior occurs similarly to digit 9. There is a transition from groups of digits 4 to 9, i.e., there is a semantic in this transition. As we gradually move along the diagonal that connects the two groups, digits 4 resemble 9, so it is tough to differentiate between them in the middle of the diagonal. It is also worth noting that the further away from the denser regions of the points cloud, the less readable the numbers are. For instance, the two digits four are depicted below the transition diagonal. We see that our proposal uses markers to help in the convergence and finds a latent space that preserves the semantics of the original data. This behavior corroborates our initial hypothesis described in (**HA.2**).

It is also worth noting that our proposal has no sensitive learning in the presence of multiple markers, i.e., even in this experiment where we have defined three similarity markers and two dissimilarity markers, our proposal does not use all. This behavior is emphasized in Figures A.6b and A.7b Where our proposal removes excessive markers from the groupings by locating these markers far away from the data, this behavior indicates that the number of markers is a virtual parameter of the model.

We hypothesize that markers group data points considered similar (in our context, which have the same labels) and dissimilar (different labels) in disjoint regions. Figures A.7b and A.6b show this behavior, where we can see similar and dissimilar groups in distinct (and disjoint) regions in S-space. In addition, we can notice in Figure A.6a that our proposal allows different clusters for the same class (*optimal latent space*), as mentioned in Definition 3.1.