

APPRENTISSAGE FÉDÉRÉ : IMPACT DE LA  
GÉNÉRALISATION ET DE LA PERSONNALISATION SUR LA  
PERFORMANCE ET LA SÉCURITÉ

APPLICATIONS MÉDICALES

THÈSE PRÉSENTÉE À LA FACULTÉ DES ÉTUDES  
SUPÉRIEURES ET DE LA RECHERCHE EN VUE DE  
L'OBTENTION DE LA MAÎTRISE ÈS SCIENCES  
(INFORMATIQUE)

**SINDA BESROUR**

DÉPARTEMENT D'INFORMATIQUE  
FACULTÉ DES SCIENCES  
CAMPUS DE MONCTON  
UNIVERSITÉ DE MONCTON

Novembre 2024

# Composition du Jury

Président du jury :

**Dr Mustapha Kardouchi**

Directeur du dépt. Informatique, Université de Moncton,  
Campus de Moncton

Examineur externe :

**Dr Mohsen Guizani**

Professeur, Mohamed Bin Zayed University of Artificial  
Intelligence

Examineur interne :

**Dr Habib Hamam**

Professeur, Université de Moncton, Campus de Moncton

Directeur de thèse :

**Dr Jalal Almhana**

Professeur, Université de Moncton, Campus de Moncton

# Remerciements

Je tiens à exprimer ma profonde gratitude et mon respect envers Professeur Dr. Jalal Almhana, qui a accepté de m'encadrer durant ma maîtrise et m'a guidé tout au long de mon cursus universitaire au sein de l'Université de Moncton. Je lui suis profondément reconnaissante pour son dévouement en tant que professeur et superviseur. Son soutien constant, sa supervision attentive, et son engagement ont été d'une aide précieuse pour l'achèvement de ce projet de recherche.

Je souhaite également exprimer ma reconnaissance sincère au Président du jury, Dr. Mustapha Kardouchi, ainsi qu'aux autres membres du jury, Dr. Habib Hamam et Dr. Mohsen Guizani, pour le temps qu'ils ont consacré à l'évaluation de ce travail. Je remercie également tous les professeurs du département d'informatique, ainsi que la secrétaire Édith Cormier pour son soutien moral et son aide précieuse. Mes remerciements s'adressent aussi à M. Zikuan Liu, Gaël S. Mubibya, Suvam Dey, Yogesh Surapaneni et Chayma Ben Abdeljalil pour leurs contributions aux travaux publiés et acceptés.

Je tiens à exprimer ma profonde reconnaissance envers l'association Mitacs pour m'avoir accordé la bourse aux cycles supérieurs Globalink, à la Faculté des études supérieures et de la recherche (FESR) pour la bourse du Patrimoine canadien, ainsi qu'au Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) pour son soutien via une subvention accordée au Dr. Jalal Almhana.

Finalement, je remercie ma famille qui m'a soutenue dans les moments de joie et de difficulté, me permettant ainsi de croire en la réalisation de mes ambitions. Effectivement, ce travail est le fruit de mes efforts ainsi que de ceux qui m'ont encouragé tout au long de ce parcours. Je tiens à mentionner en particulier mon père Mohammed Sadok B., ma mère Lamia B., mon frère Alaeddine B., mon amie Sadie Baramikova, et ma propriétaire-bailleur Shoko Leger.

Sinda BESROUR  
Novembre 2024

# Avertissement : Thèse par Publication

Cette thèse est structurée autour de trois articles que nous avons soumis à des conférences à comité de lecture. Chaque article représente une unité indépendante dans laquelle nous avons exposé les motivations, les contributions, la méthodologie et les résultats de notre recherche. Ainsi, les premières pages de cette thèse seront consacrées à une présentation succincte du contexte. Pour plus de détails, nous invitons les personnes intéressées à se référer aux publications individuelles.

# Table des matières

<b>Résumé</b>	<b>6</b>
<b>Abstract</b>	<b>8</b>
<b>Introduction Générale</b>	<b>10</b>
1. Motivations . . . . .	12
a. Chapitre 1 : Prise en compte du contexte lors de la classification	12
b. Chapitre 2 : Hétérogénéité des données dans l'apprentissage fédéré . . . . .	12
c. Chapitre 3 : Sécurité du réseau fédéré . . . . .	12
2. Contributions . . . . .	13
a. Chapitre 1 . . . . .	13
b. Chapitre 2 . . . . .	13
c. Chapitre 3 . . . . .	14
3. Approches . . . . .	14
a. Données . . . . .	14
b. Plateformes et langages de programmation . . . . .	14
4. Articles préparés . . . . .	15
<b>1 Context-Aware Hard and Slow Fall Detection</b>	<b>17</b>
1.1. Introduction . . . . .	18
1.2. Literature Review . . . . .	19
1.3. Approach . . . . .	20
A. Generation of fall data in a realistic context . . . . .	20
B. Model selection . . . . .	22
C. Implementation of HF and PF detection . . . . .	23
D. Implementation of SF detection . . . . .	24
1.4. Experimental Results . . . . .	26
1.5. Concluding Remarks and Future Works . . . . .	28

<b>2</b>	<b>Generalization vs Personalization : A Trade-off for better Data Heterogeneity impact Mitigation in FL</b>	<b>30</b>
2.1.	Introduction . . . . .	31
2.2.	Literature Review . . . . .	33
2.3.	Approach . . . . .	34
2.4.	Experimental Settings and Results . . . . .	39
A.	Experimental settings . . . . .	39
a.	Data collection . . . . .	39
b.	Data preprocessing . . . . .	40
c.	Data distribution across client nodes . . . . .	40
B.	Experimental results . . . . .	41
2.5.	Concluding Remarks and Future Works . . . . .	44
<b>3</b>	<b>A Non-Linear Personalized Approach to Mitigate Poisoning Attack Coalitions in FL</b>	<b>45</b>
3.1.	Introduction . . . . .	46
3.2.	Literature Review . . . . .	48
3.3.	Approach . . . . .	49
A.	Data . . . . .	49
B.	Selecting the appropriate DL model . . . . .	49
C.	Establishing the FL architecture . . . . .	49
D.	Implementing the AC . . . . .	51
E.	Implementing our solution to counter the effect of the AC . . . . .	53
3.4.	Experimental Results . . . . .	55
A.	Impact of the AC on the accuracy and F1-score . . . . .	55
B.	Proposed solution to counter the AC's damaging effect . . . . .	56
C.	Comparison with the traditional aggregation strategy FedAvg . . . . .	57
3.5.	Concluding Remarks and Future Works . . . . .	57
	<b>Conclusion Générale</b>	<b>59</b>
	<b>Sigles Abréviations</b>	<b>61</b>
	<b>Liste des tableaux</b>	<b>63</b>
	<b>Table des figures</b>	<b>64</b>
	<b>Liste des Algorithms</b>	<b>65</b>
	<b>Bibliographie</b>	<b>66</b>

# Résumé

Avec l'émergence de l'intelligence artificielle et de l'Internet des objets, des recherches approfondies ont été menées sur d'innombrables solutions de pointe, notamment les assistants vocaux, les véhicules autonomes, les jumeaux numériques et la télésanté. La plupart de ces études ont adopté des approches traditionnelles d'apprentissage automatique et profond, où les données sont collectées et stockées sur un serveur dans le cloud pour le prétraitement, l'extraction de caractéristiques et l'entraînement du modèle. Le principal défi de cette approche centralisée est d'assurer la confidentialité des données, en particulier celles liées à la santé. En effet, les données des patients sont extrêmement sensibles, et toute fuite ou falsification pourrait entraîner de graves conséquences. L'apprentissage fédéré s'est révélé être une alternative intéressante à l'approche centralisée. Le travail présenté dans cette thèse couvre trois aspects correspondant à trois chapitres, qui représentent trois publications : 1) Une application médicale : La détection des chutes, 2) La détection des chutes dans le cadre de l'apprentissage fédéré, et 3) La sécurité de l'apprentissage fédéré.

Chapitre 1 (Context-Aware Hard and Slow Fall Detection) : Dans ce chapitre, nous démontrons qu'il est impossible d'atteindre des taux élevés d'exactitude dans la détection des chutes lorsque celles-ci sont précédées d'activités de la vie quotidienne.

Chapitre 2 (Generalization vs Personalization : A Trade-off for Better Data Heterogeneity impact Mitigation in FL) : Dans ce chapitre, nous abordons la détection des chutes dans le cadre de l'apprentissage fédéré afin de répondre aux problèmes de confidentialité. Plus précisément, nous développons une solution personnalisée pour résoudre le problème de l'hétérogénéité des données.

Chapitre 3 (A Non-Linear Personalized Approach to Mitigate Poisoning Attack Coalitions in FL) : Dans ce chapitre, nous nous concentrons sur l'aspect de la sécurité dans l'apprentissage fédéré. Plus précisément, nous mettons en œuvre une approche personnalisée non linéaire pour atténuer l'effet des attaques par em-

empoisonnement des données et des modèles sur les métriques de prédiction.

**Mots clés :** Intelligence Artificielle, Internet des Objets, Capteurs Portables, Télésanté, Confidentialité des Données, Apprentissage Fédéré, Hétérogénéité des Données, Empoisonnement des Données et des Modèles, Personnalisation.

# Abstract

With the recent artificial intelligence and Internet of Things boom, extensive research has been conducted on countless cutting-edge solutions including voice assistants, autonomous vehicles, digital twins, e-health, etc. Most of these studies have adopted traditional machine and deep learning approaches where data is collected and stored in a central node for pre-processing, feature extraction, and model training. The main challenge of this centralized approach is ensuring data privacy, particularly concerning health-related information. Indeed, patient data is extremely sensitive, and any breach or falsification could have serious consequences. Federated learning has emerged as an interesting alternative to the centralized approach. The work presented in this thesis covers three aspects corresponding to three chapters, which represent three publications : 1) A medical application : Fall detection, 2) Fall detection within the framework of federated learning, and 3) The security of federated learning.

Chapter 1 (Context-Aware Hard and Slow Fall Detection) : In this chapter, we demonstrate that it is impossible to achieve high accuracy rates in fall detection when activities of daily living precede falls.

Chapter 2 (Generalization vs. Personalization : A Trade-off for Better Data Heterogeneity Impact Mitigation in FL) : In this chapter, we address fall detection within the framework of federated learning to tackle privacy issues. More specifically, we develop a personalized solution to address the problem of data heterogeneity.

Chapter 3 (A Non-Linear Personalized Approach to Mitigate Poisoning Attack Coalitions in FL) : This chapter focuses on the security aspect of federated learning. More specifically, we implement a non-linear personalized approach to mitigate the effects of data and model poisoning attacks on prediction metrics.

**Keywords :** Artificial Intelligence, Internet of Things, Wearable Sensors, E-health, Data Privacy, Federated Learning, Data Heterogeneity, Data and Model Poisoning, Personalization.

# Introduction Générale

L'intelligence artificielle (IA) et l'Internet des objets (IoT) ont révolutionné de nombreux domaines, notamment le transport, la finance, le commerce, le secteur manufacturier et la santé, en créant un monde plus connecté et plus intelligent [1]. L'IA a amélioré les systèmes IoT en analysant de grandes quantités de données collectées à partir d'appareils interconnectés, ce qui a permis de prendre des décisions plus intelligentes et de bénéficier de capacités prédictives. En particulier, dans le domaine de la télésanté, des appareils IoT tels que les détecteurs de rythme cardiaque, les accéléromètres, les gyroscopes, les capteurs d'activité électrodermale et les thermomètres corporels peuvent surveiller les signes vitaux des patients en temps réel. Un système d'IA peut utiliser les données de ces capteurs pour prédire des problèmes de santé potentiels, tels que le niveau de stress ou le risque de chute, avant qu'ils ne surviennent, ce qui permet de prodiguer des soins proactifs ou, au minimum, de réduire les risques [2]. La majorité des solutions de télésanté disponibles sur le marché emploient une approche d'apprentissage automatique traditionnel, où d'énormes quantités de données personnelles sont collectées et stockées sur un serveur central, généralement situé au niveau cloud, pour l'entraînement des modèles d'IA [3], comme illustré sur la FIGURE 1.

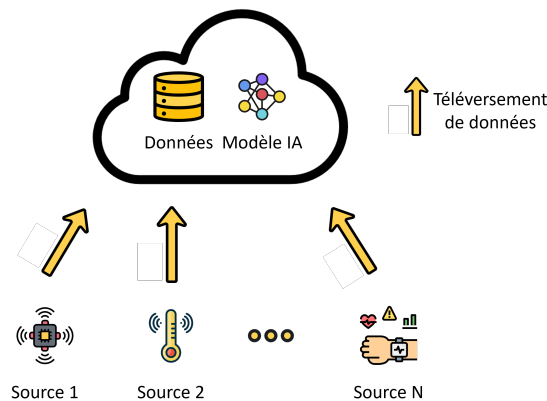


FIGURE 1 – Topologie traditionnelle de l'apprentissage automatique.

Cette centralisation expose les données à des risques d'accès non autorisé, de violation ou d'utilisation abusive, car elles sont regroupées en un seul endroit, où elles peuvent être plus facilement compromises. L'apprentissage fédéré (AF), une solution distribuée, a été proposé, entre autres, pour mieux préserver la confidentialité de ces données, ainsi que des modèles qui y sont associés. La FIGURE 2 montre l'architecture de l'AF : on y voit un agrégateur qui distribue les gradients initiaux du modèle d'apprentissage automatique ou profond à plusieurs nœuds clients, qui effectuent l'entraînement sur leurs données locales. Les gradients résultants, également connus sous le nom de modèles locaux, sont ensuite renvoyés à l'agrégateur, qui les fusionne en un seul modèle global, lequel sera ultérieurement renvoyé à tous les nœuds clients pour la prédiction [4].

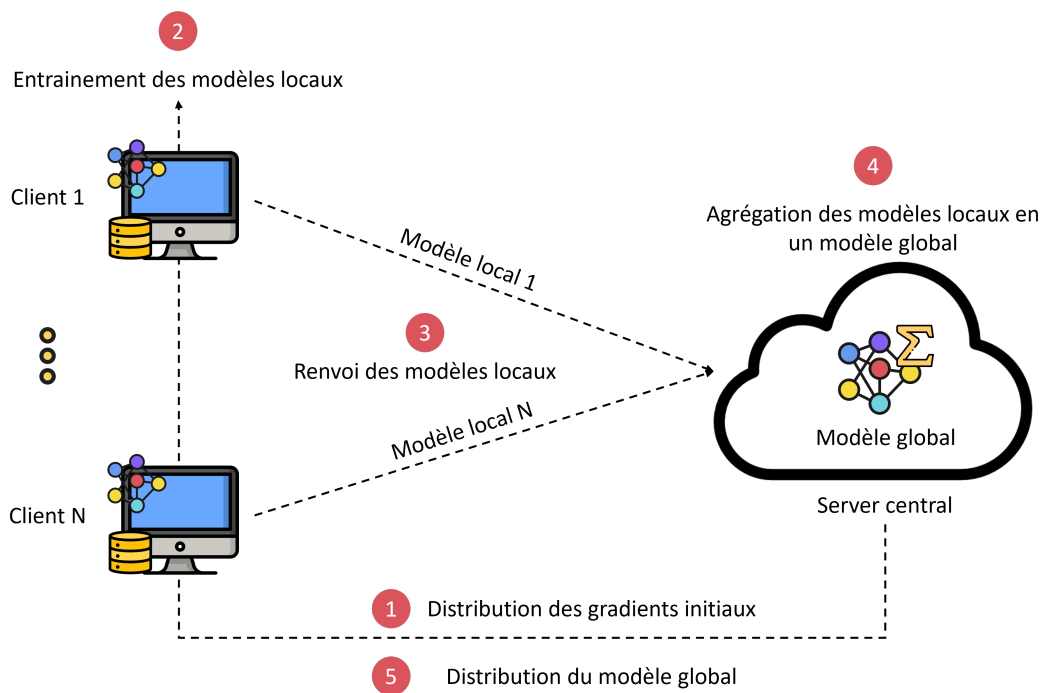


FIGURE 2 – Architecture générale de l'apprentissage fédéré.

Bien que l'apprentissage fédéré représente une alternative intéressante pour remédier au problème de la confidentialité des données, de nouveaux défis se posent concernant la capacité prédictive du modèle global et la sécurité du réseau fédéré. Les deux derniers chapitres sont dédiés à l'exploration de ces deux défis. Le premier chapitre, en revanche, constitue une étape préliminaire au deuxième. En effet, il se concentre sur le développement d'une méthode sensible au contexte pour la détection des chutes, tandis que le deuxième chapitre se focalise sur la mise en œuvre d'une approche d'apprentissage fédéré pour la détection des chutes.

# 1. Motivations

Comme indiqué précédemment, chaque chapitre aborde un défi spécifique. Par conséquent, nous identifions trois motivations principales :

## a. Chapitre 1 : Prise en compte du contexte lors de la classification

La prise en compte du contexte lors de la classification est un élément essentiel pour développer des solutions robustes qui reflètent la réalité. En particulier, dans le domaine de la détection des chutes, il est impératif d'intégrer les données sur les chutes avec celles des activités de la vie quotidienne. Les études précédentes n'ont pas pris en compte le contexte, ce qui a conduit à des métriques élevées. De tels résultats ne sont pas réalisables dans un contexte réel, où les chutes sont souvent précédées par des activités de la vie quotidienne telles que marcher, courir, etc.

## b. Chapitre 2 : Hétérogénéité des données dans l'apprentissage fédéré

Le problème de l'hétérogénéité des données est étroitement lié à l'apprentissage fédéré et impacte considérablement la capacité prédictive du modèle global. Cette hétérogénéité provient du fait que les nœuds clients disposent d'une variété de sources de données et de contextes. Bien que ce sujet ait été largement traité dans la littérature, plusieurs aspects restent à explorer, notamment l'hétérogénéité du contexte et le taux de déséquilibre entre les nœuds clients. Les études précédentes ont utilisé la personnalisation pour aborder le défi de l'hétérogénéité. Cependant, la personnalisation seule n'est pas suffisante ; il est essentiel de trouver un compromis entre la généralisation et la personnalisation.

## c. Chapitre 3 : Sécurité du réseau fédéré

Bien que l'apprentissage fédéré ait été introduit pour réduire le risque de violation de la confidentialité grâce à son architecture décentralisée, plusieurs failles de sécurité peuvent être exploitées par des personnes malveillantes, entraînant ainsi une détérioration de la performance. En particulier, un individu malveillant peut transformer un groupe de nœuds clients légitimes en zombies. Ces zombies peuvent alors exécuter une série d'attaques, affectant ainsi la confidentialité, l'intégrité, et/ou la disponibilité du réseau fédéré. Plusieurs recherches ont été menées dans ce contexte. Cependant, aucune n'a exploré le potentiel d'une coalition d'attaques où plus de la moitié du réseau est transformée en nœuds clients malveillants et

lance différents types d’attaques simultanément. De plus, la majorité des approches proposées dans la littérature utilisent des techniques complexes telles que la blockchain et la détection des nœuds malveillants. Même les approches employant la personnalisation utilisent des stratégies d’agrégation inappropriées.

## 2. Contributions

### a. Chapitre 1

- Génération d’une base de données pour la détection des chutes, prenant en compte le contexte en intégrant les données de chutes avec celles des activités de la vie quotidienne.
- Étude de la détection des chutes lentes et brutales dans un contexte réaliste, avec une comparaison de nos résultats à ceux publiés dans la littérature. Plusieurs algorithmes d’apprentissage automatique ont été utilisés et comparés. La détection des chutes ne semble pas aussi facile que le suggèrent les résultats publiés précédemment.
- Évaluation de la possibilité de détection précoce des chutes (pré-chute), ce qui pourrait aider à réduire l’impact des chutes.

### b. Chapitre 2

- Nous proposons une approche robuste d’apprentissage fédéré pour atténuer l’impact de l’hétérogénéité des données grâce à un compromis entre généralisation et personnalisation. Dans ce contexte, nous abordons trois aspects de l’hétérogénéité, notamment les données non identiquement et indépendamment distribuées (non-IID), l’hétérogénéité du contexte, et le taux de déséquilibre, y compris les nœuds clients à une seule étiquette, qui sont perpétuellement exclus du processus d’apprentissage. La généralisation consiste à mettre en œuvre une stratégie d’agrégation personnalisée qui attribue aux nœuds clients des coefficients de pondération basés sur leurs contributions locales, tandis que la personnalisation compense les lacunes des clients moins performants.
- Nous mettons en œuvre notre approche dans le cadre de la détection des chutes lentes et brutales dans un contexte réaliste après avoir collecté des données contextuelles à partir de capteurs tels que l’accéléromètre et le capteur de rythme cardiaque.
- Nous comparons notre approche aux stratégies d’agrégation traditionnelles, telles que la moyenne fédérée (FedAvg), et constatons sa performance supérieure en termes de score F1.

- Nous réalisons une étude comparative avec les recherches précédentes sur la détection des chutes.

**Remarque :** Des détails supplémentaires concernant le chapitre 2, y compris la conception UML ainsi que l’implémentation des parties mobile et cloud, sont disponibles dans le rapport technique [5].

### c. Chapitre 3

- Nous introduisons le concept général de coalition d’attaques, en l’illustrant à travers diverses formes. Pour notre approche, nous nous concentrons sur une forme particulière où nous augmentons progressivement le pourcentage de nœuds clients malveillants de 14% à 57%. Nous exécutons une série d’attaques par empoisonnement de données et de modèles, notamment une inversion d’étiquettes, un déséquilibre sévère des données, une disponibilité minimale des données, et des gradients bruyants.
- Nous analysons l’impact de la coalition d’attaques et proposons une approche personnalisée d’apprentissage fédéré basée sur une stratégie d’agrégation non linéaire qui réduit l’effet des gradients malveillants sur le modèle global. Nous renforçons l’effet de notre stratégie par une interpolation entre le modèle global et le modèle local du côté client.
- Nous mettons en œuvre notre approche dans le cadre de la détection du stress des infirmières. Nos résultats montrent une diminution substantielle des métriques de prédiction après la coalition d’attaques, et une hausse significative suite à l’application de notre approche.

## 3. Approches

### a. Données

Pour la détection des chutes, nous avons collecté des données représentant des chutes ainsi que des activités de la vie quotidienne telles que marcher, courir, etc., à partir de capteurs tels que Wahoo [6] et MetaMotionS [7]. Nous avons également inclus une base de données publique [8]. Concernant la détection du stress chez les infirmières, nous avons utilisé une base de données disponible sur Kaggle [9].

### b. Plateformes et langages de programmation

Dans cette thèse, nous utilisons Python 3.9.13 comme langage de programmation. Les versions des frameworks et des bibliothèques sont détaillées dans la

TABLE 1. En ce qui concerne les ressources, nous utilisons un ordinateur portable HP Pavilion équipé d’un processeur AMD Ryzen™ 7, d’une carte graphique NVIDIA® GeForce® RTX™ 3050 Ti, et de 16 Go de mémoire.

TABLE 1 – Bibliothèques et frameworks utilisés dans cette thèse.

Bibliothèque	Version	Usage
Flower	1.4.0	Framework d’apprentissage fédéré
Tensorflow-Keras	2.11.0	Framework d’apprentissage profond
Scikit-learn	1.2.0	Framework d’apprentissage automatique
Numpy	1.23.5	Manipulation des tableaux
Pandas	1.5.2	Manipulation des Dataframes et des fichiers CSV
Scipy	1.10.0	Manipulation des signaux
Librosa	0.9.2	Traitement de signal
Matplotlib	3.6.2	Visualisation
FastAPI	0.95.1	Connexion client/server pour les implémentations en temps réel

## 4. Articles préparés

Chapitre 1 : S. Besrou, G. S. Mubibya, Z. Liu and J. Almhana, “Context-Aware Hard and Slow Fall Detection,” 2024 International Wireless Communications and Mobile Computing (IWCMC), Ayia Napa, Cyprus, 2024. (publié)

Chapitre 2 : S. Besrou, G. S. Mubibya, C. Ben Abdeljelil and J. Almhana, “Generalization vs Personalization : A Trade-off for better Data Heterogeneity impact Mitigation in FL,” GLOBECOM 2024 - 2024 IEEE Global Communications Conference, Cape Town, South Africa, 2024. (accepté)

Chapitre 3 : S. Besrou and J. Almhana, “A Non-Linear Personalized Approach to Mitigate Poisoning Attack Coalitions in FL,” ICC 2025 - IEEE International Conference on Communications, Montreal, Canada, 2025. (soumis)

## **Autres articles publiés**

Bien que cette thèse se limite à trois publications, j'ai pu participer à trois autres publications durant ma préparation de maîtrise :

S. Besrou, S. Dey, G. S. Mubibya and J. Almhana, "Subject Identification Using Behavioral Cues and Machine Learning," ICC 2024 - IEEE International Conference on Communications, Denver, CO, USA, 2024.

S. Besrou, Y. Surapaneni, G. S. Mubibya, F. Ashkar and J. Almhana, "A Transformer-Based Approach for Better Hand Gesture Recognition," 2024 International Wireless Communications and Mobile Computing (IWCMC), Ayia Napa, Cyprus, 2024.

G. S. Mubibya, S. Besrou and J. Almhana, "A Real-Time IoT System and ML algorithms : A Comparative Study," ICC 2022 - IEEE International Conference on Communications, Seoul, Republic of Korea, 2022.

# Chapitre 1

## Context-Aware Hard and Slow Fall Detection

# Abstract

Fall is one of the main causes of injuries for the elderly, and fall detection (FD) for senior monitoring has received considerable attention from both the academic community and healthcare industries. In recent years, there has been an increasing interest in using wearable sensors, such as accelerometers to monitor the subject's body movement and apply Machine Learning (ML) methods to detect and prevent falls. Since it is extremely difficult to collect accelerometer data of real falls during activities of daily living (ADL), researchers tended to rely on simulating falls in well-protected environments. They collected ADLs separately, applied ML algorithms to classify falls and ADLs, and reported very high FD accuracy rates. However, these studies cannot be applied in a real fall context. In this paper, instead of classifying ADL and fall separately, we propose to incorporate fall data within ADL data to obtain more realistic datasets and apply ML to detect falls. Several ML algorithms including CatBoost (CB), Decision Tree (DT), Random Forest (RF), and XGBoost (XGB) were applied to the datasets. Experimental results show a fall detection accuracy of 88.70%. We also extend our work to cover slow fall which, to the best of our knowledge, was not extensively addressed in previous works.

## 1.1. Introduction

Based on the surveys conducted by the World Health Organisation (WHO) [10], fall is considered the second most common cause of death globally. Additionally, each year, 37.3 million falls are severe enough to require medical attention, and most of them often occur among adults over 60 years of age. As such, detecting fall, pre-fall in particular, is a top priority. With the rapid technological advancement in artificial intelligence (AI) and the Internet of Things (IoT) in particular, the use of wearable sensors including the accelerometer, gyroscope, and magnetometer [11], and the application of AI for FD have been the focus of several research studies [12]. Nevertheless, FD based on wearable sensors, especially pre-fall, is not an easy task due to the extreme rarity of the event compared to other ADLs. Indeed, it might take 100 000 activities to capture only 100 falls [13]. Consequently, acquiring enough spontaneous data to train an AI model is very difficult and previous works have focused on simulated data with carefully established scenarios. However, the datasets available on the internet [14] do not reflect reality since they capture falls and ADLs separately. Most studies on FD and pre-fall detection (PFD) are based on these datasets which is a major limitation despite resulting in high accuracy. On the other hand, slow fall detection (SFD) is another topic that has not been properly addressed yet. Many people, especially the elderly, experience slow falls

due to feeling faint from a sudden drop in the heart rate or blood pressure. Current FD solutions are not designed to detect slow falls due to their difference compared to hard falls in terms of data samples. As such, many people who experience slow falls end up losing their lives. In this paper, we propose a new method to detect falls, pre-falls, and slow falls in a realistic context based on different types of falls, walking, and running activities from a mixed dataset. The contributions are the following :

1. Generating a realistic context for FD by incorporating fall signals within ADLs through a novel approach.
2. Studying slow and hard FD within a realistic context and comparing our results to previously published results. Several ML algorithms were used and compared. FD does not seem as easy as shown in previously published results.
3. Evaluating the possibility of early fall (pre-fall) detection which may help to reduce fall impact.

The rest of this paper is organized as follows : Section 1.2. summarizes previous studies on FD, PFD, and SFD. Section 1.3. describes the proposed data generation method for both hard and slow falls as well as the experimental procedures. Section 1.4. shows the experimental results, and finally, section 1.5. provides the conclusion and discussion for future work.

## 1.2. Literature Review

Up until now, several studies have been conducted on FD. [15] used K-Nearest Neighbors (KNN) to classify daily and fall activities on data collected from a smartphone. [16] used the fuzzy entropy method to classify falls and ADL. On the other hand, [17] exploited smartphone accelerometer data to detect two types of falls, and [18] implemented an FD system based on smartphone accelerometer data. [19] built a mobile-edge framework for real-time FD based on a Long Short-Term Memory (LSTM) model. [20] built a real-time FD system using accelerometer and depth-sensor data to lower the probability of false detection. [21] proposed a real-time Federated Learning IoT solution for elderly healthcare.

Apart from FD, a few works were elaborated on PFD. For instance, in our previous work [22], we adopted a Bidirectional LSTM (Bi-LSTM) model to detect pre-falls. [23] experimented with several window sizes to reach an optimal trade-off between accuracy and lead time. They adopted Support Vector Machines (SVM) as their model. [24] experimented with a dynamic threshold model to find the trade-off between maximizing true positives and reducing the lead time. [25] suggested an alarm system based on a Hidden Markov model-based SVM model.

[26] proposed a multi-class FD system based on a Convolutional Neural Network (CNN) model and data collected from various sensors.

The major flaw of all previous studies on FD and PFD is that they did not detect falls in a realistic context knowing that the fall data did not contain ADL as it should in a real scenario. Hence the reason for our work in this article. To our knowledge, we are the first to study this aspect.

On the other hand, we found only one study on SFD. [27] used a four-staged deep neural network (DNN) that combines CNN and Bi-LSTM models with a fully connected layer based on accelerometer, gyroscope, and pressure data and reached an accuracy of 90.33%. However, they did not seek to detect a slow fall in an ADL signal. Despite the challenge of SFD in the ADL signal, we decided, in this paper, to detect slow falls in a realistic context and to evaluate its feasibility.

### 1.3. Approach

In this section, we first describe how we will generate fall data within a realistic context and then we study the possibility of fall detection having this realistic data. We are addressing 3 types of falls : hard fall (HF), pre-fall (PF), and slow fall (SF).

#### A. Generation of fall data in a realistic context

In order to generate context-aware data we used fall data from [8] and ADL data from [28]. Common ADLs such as walking and running were used. From the data provided in [8], we chose 13 types of HFs, including front-falls and back-falls, which are likely to happen during walking and running activities. Unlike signals that might be generated by other sensors such as gyroscopes and magnetometers for example, HF signals from accelerometers have a very significant variation in amplitude [29]. Therefore, to reduce the processing time, we decided to select only the data generated by the accelerometer positioned at the waist. This position is the most stable body position for FD [30, 15].

FIGURE 1.1 illustrates the X, Y, and Z axes of a walking ADL signal from the dataset collected by [28]. FIGURE 1.2 illustrates the X, Y, and Z axes of a fall signal from the dataset collected by [8]. As shown in FIGURE 1.2, the axes exhibit relatively constant parts before and after the fall signal, which has a significant variation. During the generation process, we first trimmed those parts by applying the differential and deleting the nearly-null parts. Then we applied the reverse differential to regenerate the signal. Second, we incorporated the main fall signals

within the ADL signals. Finally, we computed the variance of the vector representing the coordinates of the X, Y, and Z axes. Then, we applied a low-pass filter to eliminate the noise.

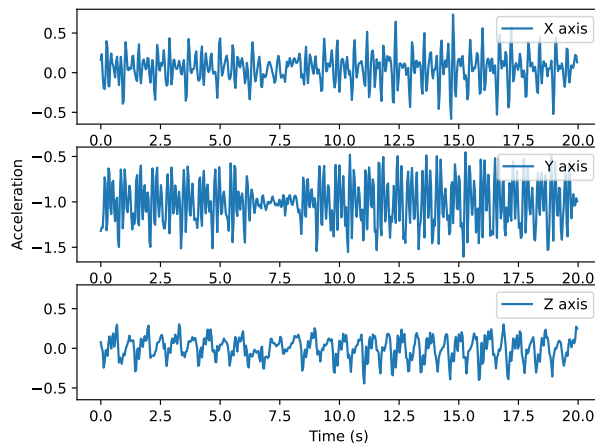


FIGURE 1.1 – A walking ADL signal from [28].

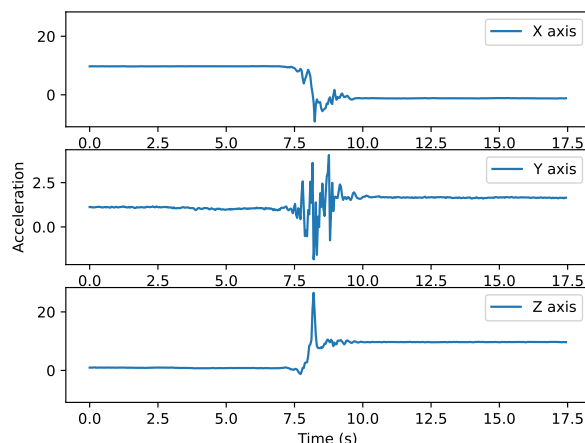


FIGURE 1.2 – A non-realistic-context HF signal from [8].

TABLE 1.1 summarizes the characteristics of the resulting dataset. FIGURE 1.3 illustrates a sample of the output data. The part before the peak is considered the area of pre-impact or PF, and the part following the pre-impact and reaching the peak is considered the area of impact. Finally, the part that follows the peak is considered the area of post-impact [22]. The generation of SF data will be detailed separately in subsection D.

TABLE 1.1 – A description of the dataset in consideration in this paper.

<b>Output Dataset</b>	
Number of samples	222 ADLs and 214 Realistic-Context Falls
Sample rate	25 Hz
Duration of sampling	5.2 seconds
Sensor positions	waist
Used sensors	Accelerometer
Activities	HF (fall after walking, fall after running), ADL (walk, run)

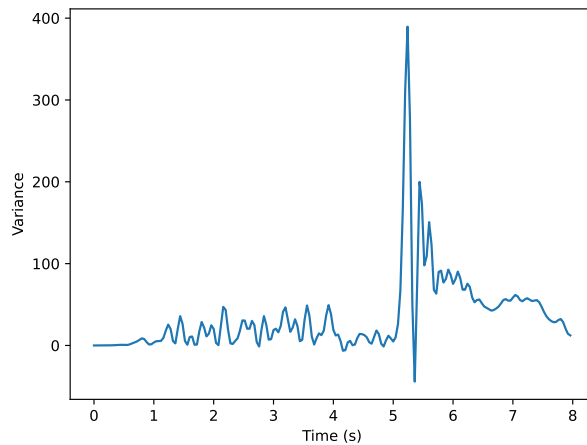


FIGURE 1.3 – An HF signal in a realistic context.

## B. Model selection

We conduct a series of experiments by applying a variety of ML algorithms to the generated realistic-context datasets and compare their performance. We chose tree-based ML algorithms; XGB, DT, RF, and CB; since they usually perform well even for large datasets and require less data pre-processing compared to other ML algorithms. TABLE 1.2 summarizes the reason behind choosing these specific tree-based algorithms.

TABLE 1.2 – Algorithms used and their characteristics.

Algorithm	Description
DT	Simple to interpret and visualize. does not require much data pre-processing such as data normalization.
XGB	Provides parallel tree boosting which makes it fast and accurate.
RF	An ensemble of decision trees. It helps improve test accuracy and reduces overfitting.
CB	Ensures superior performance compared to other gradient boost algorithms and has the best class prediction speed.

### C. Implementation of HF and PF detection

It is important to note that PF is very important for reducing the negative consequences of falls by activating a protection mechanism similar to airbags in vehicles. How this mechanism can be built is beyond the scope of this paper. Knowing the importance of early fall detection we trained our algorithms by having partial inclusions or observations of the fall signals. FIGURE 1.4 shows a representation of various percentages of inclusion from 0% (ADL only) to 100% (full fall signal inclusion), shown in the figure by a dashed line. Based on the fall duration [31], a window of 1.2 seconds was fed to the algorithms. To avoid confusion between HF and PF, we consider percentages of 10% to 20% as PFs and higher percentages as HFs.

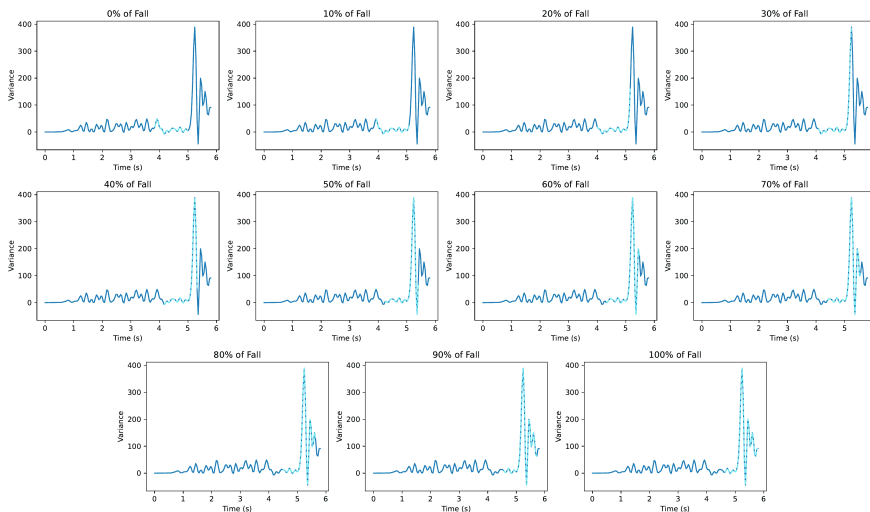


FIGURE 1.4 – Various views of partial representations of the HF shown in a dashed line.

After training our algorithms, we tried to detect HFs and PFs. FIGURE 1.5 shows a data signal of 38 seconds that contains 4 falls, 2 runs, and 3 walks. A window size of 1.2 seconds and an overlap of 50% were used during our trial [21].

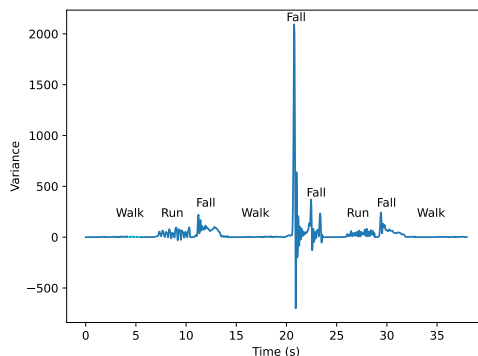
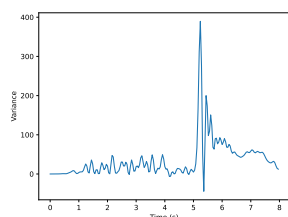


FIGURE 1.5 – An example of four fall signals within an ADL.

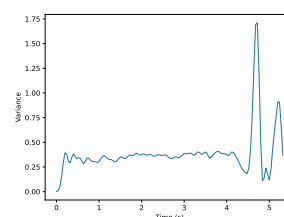
## D. Implementation of SF detection

SF is very important, however, it has very little attention in the literature as SF data is extremely scarce. In this study, we generate SF data based on the available HF data. This is feasible as long as we find similar patterns and key differences between SF and HF data. For this reason, we have collected in our laboratory a few SF data that we can compare with. Consequently, we adopted the following approach :

1. We collected SFs from two different subjects with a sample rate of 25 Hz. The accelerometer was placed on the waist.
2. We compared the SF data we collected in step 1 with HF data. FIGURE 1.6 shows both HF and SF signals.



(a) Hard fall signal.



(b) Collected slow fall sample.

FIGURE 1.6 – Samples of the HF and SF data.

As we may see from this figure, we can identify three similarities and two key differences between HF and SF signals. The similarities are :

1. The pre-impact in both signals has the same duration of approximately 4.5 seconds.
2. The impact zone : It is the part where the signal starts to rise dramatically until reaching the peak and then drops. SF and HF seem to have a similar shape and duration (approximately 0.5s).
3. Impact signal orientation : They are the same for HF and SF.

The key differences are :

1. The amplitude is extremely different. The maximum variance is 1.75 for SF and 400 for HF.
2. The post impacts do not have similar shapes.

Considering the similarities and differences, we generated the SF data from the HF data. First, we extracted the pre-impact zone. Then, we reduced the HF's amplitude to an approximate range of 1.5 to 2. A sample of the resulting data is shown in FIGURE 1.7.

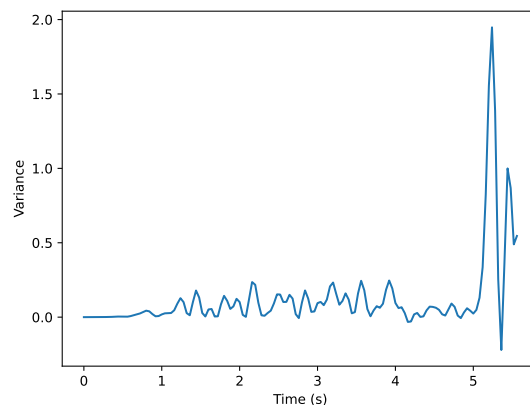


FIGURE 1.7 – A sample of the generated SF data.

For training, we used the same ML algorithms. Our classes are still fall and ADL, however, we added the SF data to the fall class. Furthermore, we attempted to detect SF in an ADL signal. FIGURE 1.8 shows a data signal including run, walk, and SF. The same window size and overlap were used as in the previous section.

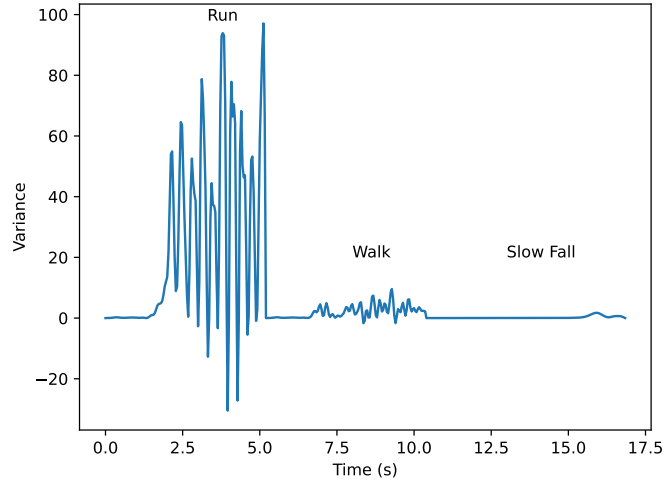


FIGURE 1.8 – An ADL signal including SF.

## 1.4. Experimental Results

This section presents our experimental results using our approach described in section 1.3.

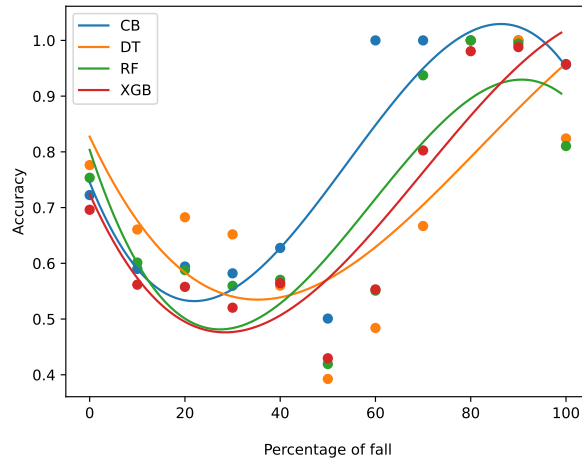


FIGURE 1.9 – The accuracy vs partial inclusion (noted as a percentage) of HF signals for several classification algorithms.

FIGURE 1.9 illustrates the accuracy of several ML algorithms as a function of partial inclusion of the HF signal. In this figure, all algorithms show an approximate V-shaped curve delimited by 0% and 100% of fall signal inclusion, which

corresponds to an average value of 75% and 90% accuracy respectively. Note that there is a variation in performance depending on the algorithm used. As the percentage of the inclusion increases, the performance of the algorithms decreases to reach a minimum at 50% and starts increasing again beyond this value. These results seem to be logical as the algorithms have a hard time to distinguish between fall and non-fall at 50% of inclusion.

Let's take the example of FIGURE 1.5 which includes four falls, two runs, and three walks, and evaluate the algorithms' performance for HF and PF detection presented in tables 1.3 and 1.4. TABLE 1.3 shows the ability of the algorithms used to detect the PF as defined before, i.e. 10-20% of observation or inclusion. From this table, we note that the success rate is 50%, 75%, 75%, and 100% for RF, DT, XGB, and CB respectively.

TABLE 1.3 – Algorithms' ability for PF detection.

	<b>DT</b>	<b>XGB</b>	<b>RF</b>	<b>CB</b>
PF 1	yes	no	no	<b>yes</b>
PF 2	no	yes	no	<b>yes</b>
PF 3	yes	yes	yes	<b>yes</b>
PF 4	yes	yes	yes	<b>yes</b>

TABLE 1.4 summarises the ability of the algorithms to detect both HF and PF. It is apparent that CB provides the best performance and RF is the worst. This is consistent with the results in FIGURE 1.9. The average performance for HF and PF detection is shown in column 3.

TABLE 1.4 – HF and PF detection for the algorithms used.

	<b>HF accuracy (%)</b>	<b>PF accuracy (%)</b>	<b>Average accuracy (%)</b>
RF	83.87	50.00	66.93
DT	80.64	75.00	77.82
XGB	85.48	75.00	80.24
<b>CB</b>	<b>88.70</b>	<b>100</b>	<b>94.35</b>

TABLE 1.5 shows the results for SF detection. The best performance 88.88% is given by DT. These results are much less than HF detection which shows the difficulties of detecting SF. Unfortunately, there is no significant SF study in the literature to compare with.

TABLE 1.5 – Accuracy of SF.

	<b>Detection accuracy (%)</b>
XGB	51.85
RF	81.48
CB	85.18
<b>DT</b>	<b>88.88</b>

Let us now compare our results to those already published in the literature. Note that our best results are taken from FIGURE 1.9. Tables 1.6 and 1.7 show the comparison results for HF and PF with two different works; [8] and [22]. Note that reference [8] does not provide PF detection. Even though we have lower performance than the other works, we believe that our results seem to be more realistic as we take into consideration a more realistic context.

TABLE 1.6 – HF detection accuracy compared to previous works.

<b>Study</b>	<b>Algorithm</b>	<b>Accuracy (%)</b>
<b>This study</b>	<b>CB</b>	<b>95.62</b>
[8]	KNN	99.91
[22]	Bi-LSTM	99.99

TABLE 1.7 – PF detection accuracy compared to previous works.

<b>Study</b>	<b>Algorithm</b>	<b>Accuracy (%)</b>
<b>This study</b>	<b>CB</b>	<b>60</b>
[22]	Bi-LSTM	99.95

As a general conclusion from all the results we obtained thus far, the low pass-filtered variance proved to be a very effective way to combine the accelerometer axes, especially in HF and PF detection. We believe that using such an approach may improve real-time implementation for FD.

## 1.5. Concluding Remarks and Future Works

In this paper, we put forward a realistic approach to detect HFs, PFs, and SFs within context-aware activities such as running and walking. Our results showed that it is not possible to obtain very high fall detection rates as reported in previously published works.

Though our approach presents rational results, further studies could be conducted based on collected data instead of generated ones. Additionally, other sensors such as the heart rate could be included in pursuit of improving the detection results. In future works, we will extend our research to mitigate these challenges and implement an approach using Federated Learning.

## **Acknowledgement**

This study is funded by the Natural Sciences and Engineering Research Council of Canada (DDG-2019-05756) as a grant to Dr. Jalal Almhana.

## Chapitre 2

# Generalization vs Personalization : A Trade-off for better Data Heterogeneity impact Mitigation in FL

# Abstract

Federated learning (FL) was introduced recently as a new machine learning (ML) paradigm. It is a distributed network of client nodes that train ML and deep learning (DL) models on their local data without sharing them to preserve data privacy (DP). However, these data are heterogeneous by nature as they are collected in different contexts using various sources such as IoT devices. Consequently, data heterogeneity (DH) in FL has brought new performance-related challenges. Few of these challenges have been addressed in the literature; moreover, context heterogeneity and balance rate were not explored at all. In this paper, we introduce an FL approach in which a trade-off between personalization and generalization is achieved to mitigate the impact of DH and obtain better performance. We focus on three DH challenges: context, non-independent and identically distributed (non-IID) data, and balance rate. For the implementation, fall detection (FD) data is used to demonstrate the potential of our approach in improving the FL system's performance. FD is an important subject and is particularly prevalent for the safety of elderly people. Hence, we collected fall data from two sensors: accelerometer (ACC) and heart rate (HR), then, we used two ML models to evaluate our approach. We utilized XGBoost (XGB) for balanced and unbalanced clients and One-Class Support Vector Machine (OC-SVM) for one-label clients. Our approach achieved an average F1-score of 88%. A comparative study was also conducted with previous works on FD. Our results showed a performance improvement which exceeded 94.30% on average.

## 2.1. Introduction

ML has attracted considerable attention as it leverages data from various sources such as IoT devices to train ML models [32]. These data are collected and stored on a central server. Centralized storage and processing may compromise DP, which is crucial in many IoT applications such as healthcare [33]. To mitigate this problem, FL has opened up new avenues for research [34]. FL is a distributed solution that allows data and knowledge to be distributed among IoT devices without compromising privacy. However, significant DH performance-related challenges were raised. Indeed, data are extremely versatile as they differ in many aspects, the three main ones being context, non-IID data, and balance rate. On the one hand, the context aspect translates into having different scenarios. These scenarios include a mixture of activities, and one activity may involve a variety of movements. For example, in a fall scenario, a person can fall forward while walking or slip and fall backward while running. On the other hand, the non-IID data aspect, also known as statistical heterogeneity, stems from the difference in physiological characteristics and habits amongst the client nodes. Lastly,

the balance rate aspect indicates the label distribution in a specific client node. For instance, some client nodes may have data classes with different balance rates, others could have both classes with the same balance rate, and other clients could have only one class. In this paper, we call these clients one-label clients. Existing FL studies that tackle the DH challenge tend to only research the non-IID aspect and omit, in the process, the context and the balance rate including one-label clients.

In light of these challenges, this research paper proposes a novel FL approach to address the three aspects of DH mentioned above through a trade-off between generalization and personalization while addressing the limitations of previous works. generalization is essential to build an inclusive and robust model that includes a maximum number of client nodes while weighing them according to their data calibers to reduce data and concept drift. And personalization serves to uphold fairness between high and low-caliber client nodes. Accordingly, we implement our approach in the case of FD. Indeed, the issue of falls among people, particularly the elderly, has emerged as a pressing concern in healthcare and aging societies. Falls not only cause immediate injuries but also lead to long-term physical and psychological consequences, diminishing the overall quality of life for affected individuals [35].

Extensive research has been conducted to develop effective FD systems that can promptly detect and alert caregivers or emergency services [36, 20]. For these reasons, we demonstrate the potential of our approach in FD. However, unlike prior research [37], we implement our approach in a realistic context, i.e., we incorporate various types of falls including hard falls (HF) and slow falls (SF) within walking and running activities to mimic real-life scenarios. For that, we use ACC and HR data we collected. As for ML models, we use XGB for balanced and unbalanced client nodes and OC-SVM for one-label ones. The primary contributions of this paper are the following :

1. We propose a robust FL approach named FedHSFD to mitigate the impact of DH through a trade-off between generalization and personalization. In this context, we address the three aspects of DH described above and include one-label clients who are perpetually excluded from the learning process. The generalization part consists in implementing a custom aggregation strategy that weighs FL clients based on their local contributions, whereas personalization compensates for the shortcomings of less-performing clients to ensure maximal results.
2. We implement this FL approach for HF and SF detection in a realistic context after collecting realistic context data from ACC and HR sensors.

3. We compare our approach to “vanilla” aggregation strategies such as federated averaging (FedAvg), and witness its superior performance in terms of F1-score.
4. We provide a comparative study with prior FD research to prove the effectiveness of our approach.

The remainder of this paper is organized as follows. The literature review is presented in section 2.2. In section 2.3., we describe our approach. Experimental settings and results are provided in section 2.4. Concluding remarks and future work are presented in section 2.5.

## 2.2. Literature Review

Here, we conduct a literature review mainly on FL and moderately on FD, since it is the application we have chosen.

Various research studies have been conducted on FD using simulated fall data collected from ACC. For instance, [15] and [17] implemented ML models to detect falls using smartphone ACC data while [19] and [20] implemented real-time FD systems based on ACC data collected from wearable sensors. Unfortunately, very few research works [38, 27] were carried out on SF detection, probably because of the lack of data and difficulties of implementation.

To the best of our knowledge, none of the above studies implemented FD in a realistic context, for example considering the fall within ADL [38]. Very few of them attempted to explore the correlation between HR and FD [39]. In this paper, we will use both HR and ACC data for SF and HF detection. We believe that falls impact HR and, as a result, should be taken into account in FD. Another problematic aspect of previous FD studies is that data are stored on a central shared node, which may affect DP. FL offers an interesting alternative where data are distributed among separate nodes. For this end, among others, FL was adopted for FD. However, FL implementation raised several challenges, such as DH and label scarcity. Several works were dedicated to solving these challenges.

[21] proposed a few-shot reinforcement learning-based (RL) framework to solve the problem of label scarcity, data unbalance, and statistical heterogeneity. In [40], they adopted transformers to solve statistical DH. [41] proposed a FL framework for elderly FD that uses extreme learning to solve the problem of inconsistent data distribution. [42] proposed a semi-supervised online personalized framework that tackles privacy preservation, label scarcity, real-timing, and DH. [43] adopted a

hybrid approach based on two existing personalized FL algorithms to solve statistical heterogeneity.

Unfortunately, these previous research studies have various limitations when it comes to solving the three DH aspects mentioned previously. Their primary focus is on non-IID distribution. They concentrate their efforts mainly on personalization and tend to ignore generalization, which is equally important for a robust global model (GM). Indeed, they use either FedAvg as an aggregation strategy or a custom strategy which is essentially similar to FedAvg, ignoring key factors such as the local model (LM) performance, the data balance rate, and the length of the train set. They also exclude one-label client nodes from their solutions. In addition, the majority of these previous studies adopt complex deep learning models, which increases the processing time and over-fitting. In this paper, we propose a novel approach that considers the three aspects mentioned above while balancing GM generalization and personalization to ensure maximal performance. We use lightweight ML models, which require much less processing time and improve the possibility of real-time implementation.

## 2.3. Approach

This section details our approach FedHSFD. We describe the theory behind achieving GM robustness and inclusiveness through a trade-off between generalization and personalization.

We adopt a centralized architecture since smartphones have limited memory, storage, and computation power to receive and aggregate the models of many client nodes. Our architecture is divided into 3 layers :

1. Cloud layer : It represents the server/central node. It is responsible for orchestrating the FL training sessions and aggregating the LM sent from the edge nodes into a GM.
2. Edge layer : It includes the client nodes ; each client node represents a smartphone that hosts the mobile application responsible for :
  - (a) Participating in the FL training sessions.
  - (b) Offline data acquisition and storage for LM training during a FL training session.
  - (c) Online data acquisition for real-time prediction. This phase uses the personalized GM resulting from the last FL training session.
3. IoT layer : It includes the wearable sensors. In our application, we use ACC and HR sensors.

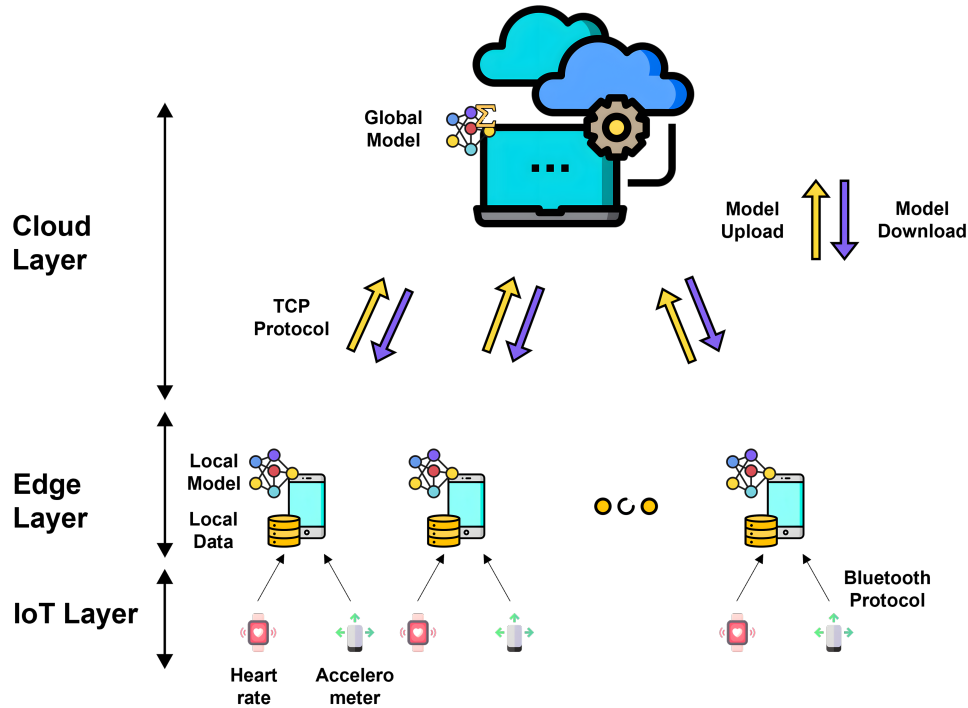


FIGURE 2.1 – The general architecture of FedHSFD.

Our FedHSFD approach offers a trade-off between model generalization and personalization to ensure maximum performance, particularly, the F1-score. To achieve this, we propose the following algorithm 2.1 which provides detailed steps for a FL training session. In this algorithm, we have three types of client nodes. They are classified based on their data distribution :

1. Unbalanced clients that have both classes (fall and non-fall) with a different number of data frames for each class. The XGB model is used as it is fast and accurate [38].
2. Balanced Clients which have both classes with the same number of data frames for each class. For the same reasons mentioned above, XGB is used as well.
3. One-label clients that have only one class (fall or non-fall). We use OC-SVM as the ML model as it is frequently used for anomaly detection where only one label is available [44].

---

**Algorithme 2.1** : FedHSFD training session

---

**Input :**  $N$  : Number of client nodes  
 $i$  : Client index  
 $D_i$  : Dataset of client  $i$   
 $Y_i$  : Label array of client  $i$   
 $f_i$  : Length of the fall class of client  $i$   
 $nf_i$  : Length of the Non-fall class of client  $i$   
 $\lambda_i$  : The personalization coefficient of client  $i$   
 $nb_r \in \mathbb{N}^*$  : Number of rounds per train session

**Output :**  $GM$  : The generalized GM  
 $PM_i$  : The personalized GM of client  $i$

**begin**

Begin of the training session by the central node.  
Broadcast of a notification to the  $N$  client nodes.  
Acceptation of a subset  $N_p$  of the  $N$  client nodes to participate in the training session.

**while** *Training* **do**

**if** *The central node has the GM from the last training session stored* **then**

    Send  $GM$  weights of the last training session to the  $N_p$  client nodes.

    The  $N_p$  client nodes initialize their LMs.

**else**

    Retrieve the initial weights from a random client node.

    Broadcast them to the  $N_p$  client nodes.

**end**

**for**  $round \in \{1, 2, \dots, nb_r\}$  **do**

**for**  $i \in \{1, 2, \dots, N_p\}$  *in parallel* **do**

$LM_i \leftarrow$  Local gradients using  $\{D_i, Y_i\}$

$l_i \leftarrow$  Log loss of the local gradients

$\alpha_i \leftarrow \phi(l_i, f_i, nf_i)$

      Upload  $LM_i$  and  $\alpha_i$  to the central node

**end**

$GM \leftarrow \sum_{i=1}^{N_p} \alpha_i LM_i$

**for**  $i \in \{1, 2, \dots, N_p\}$  *in parallel* **do**

      Download  $GM$

$PM_i \leftarrow \lambda_i LM_i + (1 - \lambda_i) GM$

**end**

**end**

**end**  
**return**  $\{PM_i\}_{i \in \{1, \dots, N_p\}}$

**end**

---

The formulas for the aggregation coefficient,  $\alpha_i$ , for unbalanced, balanced, and one-label client nodes are shown in equations (2.1), (2.2), and (2.3) respectively.

$$\alpha_i = \phi(l_i, f_i, nf_i) = \begin{cases} \frac{f_i + nf_i}{|f_i - nf_i|} \times \frac{1}{l_i} & f_i \neq nf_i & (2.1) \\ (f_i + nf_i) \times \frac{1}{l_i} & f_i = nf_i & (2.2) \\ \frac{1}{l_i} & f_i = 0 \text{ or } nf_i = 0 & (2.3) \end{cases}$$

$$\forall f_i, nf_i \in \mathbb{N}_+; \forall l_i \in \mathbb{R}_+^*; \forall i \in \{1, \dots, N\}$$

Let us use an example. Let  $i$  be a client node that has a dataset  $D_i$  associated with a set of labels  $Y_i$ . For instance, if client  $i$  collected 10 fall data frames and 40 non-fall data frames, then in this case,  $D_i$  would include 50 data frames that represent ACC axes (x, y, and z) and HR. Evidently, client node  $i$  is unbalanced. Thus, we use equation (2.1) to compute  $\alpha_i$ , which would be sent to the central node for aggregation after training an XGB model.  $f_i$  and  $nf_i$  respectively represent the number of fall and non-fall data frames in the train set, and they are defined by the sum :  $f_i + nf_i = 40$ , as we use 80% for training and 20% for testing. There are three motivations behind our equations :

1. Increasing the number of training data provides a more generalized and robust model [45]. This statement is expressed in the numerator  $(f_i + nf_i)$  which represents the train set length for client node  $i$ . Increasing the length will certainly increase  $\alpha_i$ .
2. The more unbalanced the data, the less performant the model, since it will add unwanted bias to the dominant class [46]. This statement is expressed in the denominator  $|f_i - nf_i|$  of equation (2.1) which represents the gap between the fall and non-fall classes in the  $i$ -th client node. Thus, if we increase the gap, the aggregation coefficient  $\alpha_i$  decreases, since they are inversely proportional.
3. Less performing LMs will negatively affect the generalization of the GM. This statement is expressed in the three equations in the term  $1/l_i$  which represents the local log loss of the  $i$ -th client node. Consequently, a smaller logarithmic loss (log loss) results in a greater aggregation coefficient  $\alpha_i$  since they are inversely proportional [47].

It is worthwhile noting that one-label clients are a particular case of unbalanced clients since one of the classes is non-existent. Equation (2.1) = (2.3) when  $f_i = 0$  or  $nf_i = 0$ .

To better understand the rationale behind client ponderations in our approach, let us consider 3 client nodes defined as follows : Client  $i$  is a balanced client, client  $j$  is an unbalanced client, and client  $k$  is a one-label client. Then, the aggregation coefficients of these clients are defined as follows :

$$\begin{cases} \alpha_i = (f_i + nf_i) \times \frac{1}{l_i} \\ \alpha_j = \frac{f_j + nf_j}{|f_j - nf_j|} \times \frac{1}{l_j} \\ \alpha_k = \frac{1}{l_k} \end{cases} \quad \forall i, j, k \in \{1, \dots, N\}$$

We demonstrate that  $f_i + nf_i > \frac{f_k + nf_k}{|f_k - nf_k|}$  and  $\frac{f_j + nf_j}{|f_j - nf_j|} > \frac{f_k + nf_k}{|f_k - nf_k|}$  in proofs 1 and 2 respectively.

**Proof 1.** Let  $f_i, nf_i \in \mathbb{N}_+^*$ ;  $f_k, nf_k \in \mathbb{N}^+$  where  $f_k \neq nf_k \forall i, k \in \{1, \dots, N\}$ .

As  $f_i + nf_i > 1$ ,

and since client  $k$  is one-labeled then,

$$f_k = 0 \text{ or } nf_k = 0$$

$$\Rightarrow \frac{f_k + nf_k}{|f_k - nf_k|} = 1$$

$$\Rightarrow f_i + nf_i > \frac{f_k + nf_k}{|f_k - nf_k|} = 1 \quad \forall i, k \in \{1, \dots, N\}$$

QED

**Proof 2.** Let  $f_j, nf_j \in \mathbb{N}_+^*$  where  $f_j \neq nf_j$ ;  $f_k, nf_k \in \mathbb{N}^+$  where  $f_k \neq nf_k \quad \forall j, k \in \{1, \dots, N\}$ .

As  $nf_j > -nf_j \quad (nf_j > 0)$ ,

$$\Rightarrow f_j + nf_j > f_j - nf_j \quad (1)$$

And as  $f_j > -f_j \quad (f_j > 0)$ ,

$$\Rightarrow f_j - nf_j > -f_j - nf_j \quad (2)$$

$$(1) + (2) \Rightarrow f_j + nf_j > f_j - nf_j > -f_j - nf_j$$

$$\Rightarrow f_j + nf_j > |f_j - nf_j|$$

$$\Rightarrow \frac{f_j + nf_j}{|f_j - nf_j|} > 1$$

and since  $\frac{f_k + nf_k}{|f_k - nf_k|} = 1$ ,

$$\Rightarrow \frac{f_j + nf_j}{|f_j - nf_j|} > \frac{f_k + nf_k}{|f_k - nf_k|} = 1 \quad \forall j, k \in \{1, \dots, N\}$$

QED

To summarize, we theoretically prove that our aggregation strategy logically ponders each client node. We ensure GM generalization and robustness by adding more weight to clients who contribute with larger, more balanced datasets (as shown in proofs 1 and 2), and better LM performance. We use linear aggregation since it is simpler and faster than a non-linear one. However, a non-linear approach can be studied in future works.

We compensate for the undermining of less performant client nodes with personalization. Thus, we adopt the model interpolation method defined in equation (2.4) :

$$PM_i \leftarrow \lambda_i LM_i + (1 - \lambda_i) GM \quad (2.4)$$

$$\forall \lambda_i \in [0, 1]; \forall i \in \{1, \dots, N\}$$

$GM$  represents the generalized GM resulting from the FedHSFD aggregation strategy.  $LM_i$  represents the local model of the  $i$ -th client node after training.  $\lambda_i$  represents the trade-off between generalization and personalization for client node  $i$ , defined as follows :

$$\begin{cases} \text{If } \lambda_i < 0.5, & GM \text{ has more weight than } LM_i \\ \text{Else if } \lambda_i = 0.5, & GM \text{ and } LM_i \text{ have the same weight} \\ \text{Else,} & LM_i \text{ has more weight than } GM \end{cases}$$

Hereafter, we will refer to  $\lambda_i$  as the personalization percentage.

## 2.4. Experimental Settings and Results

### A. Experimental settings

The experimental settings include the data collection scenario, pre-processing, and distribution across the client nodes.

#### a. Data collection

We established a comprehensive data collection scenario where we gathered realistic-context fall data from a group of 15 participants. The study was conducted in a controlled laboratory environment, ensuring safety and precision in data recording. Three distinct scenarios were designed to perform fall activities. The first scenario involved participants engaging in normal walking and then performing an HF. The second scenario included running as a preliminary activity and an HF. The third scenario featured an SF following a walking phase. Throughout

each scenario, the sensors [28] continuously recorded participants' HR and ACC data, yielding a comprehensive and valuable dataset. As for the non-fall data, we exploited the walking and running activities collected with the same sensors in our previous study [38].

## **b. Data preprocessing**

We took the following steps to pre-process our data :

- i. We over-sampled the HR signals in the fall class with linear interpolation to match the length of the ACC signals.
- ii. We performed pitch shifting and speed changing for data augmentation only on the fall data since fall signals are about 5 secs long, unlike the non-fall signals which last about 24 secs. This is a primal step to generate more windows for the client nodes.
- iii. We applied normalization on each data sample, since the ACC axes and HR don't have the same scale.
- iv. We applied the sliding window with a size of 1.2 secs. It is a reasonable duration to perform a fall, and 50% of overlap to extract fragments of each data sample [38]. These windows were then distributed to the client nodes.
- v. From each generated window, we extracted the 25 features in the time and frequency domains, namely, the Mean Absolute Value, Entropy, Spectral Energy, etc. [29]. Consequently, the input of our ML model is an array of  $25 \times 4$  values.

## **c. Data distribution across client nodes**

We defined 15 client nodes in total. To each client node, we associated a scenario with different activities, balance rates, data length, and data distribution which reflects a more plausible setting. TABLE 2.1 describes the scenario for each client node.

TABLE 2.1 – Client nodes’ description.

Client	Category	ADL	Length of the train set (data frames)
1	One-labeled	Run	38
2	One-labeled	Walk	38
3	One-labeled	HF	110
4	One-labeled	SF	110
5	One-labeled	Run and Walk	52 and 52
6	Unbalanced	Walk and HF	44 and 110
7	Unbalanced	Walk and SF	45 and 110
8	Unbalanced	Run and HF	15 and 110
9	Unbalanced	Run and SF	47 and 110
10	One-labeled	HF and SF	110 and 110
11	Balanced	Run, Walk and HF	44, 45 and 89
12	Unbalanced	Run, Walk and SF	43, 46 and 110
13	Unbalanced	Walk, HF and SF	48, 110 and 110
14	Unbalanced	Run, HF and SF	47, 110 and 110
15	Unbalanced	Walk, Run, HF and SF	44, 43, 110 and 110

## B. Experimental results

FIGURE 2.2 displays the average F1-score vs. the personalization percentage for the proposed FedHSFD approach. The average F1-score, which is often preferred over accuracy in case of data unbalance, is represented by the scatter points whereas its Gaussian fit is represented by the continuous curve, and its linear fit is represented by the dashed line.

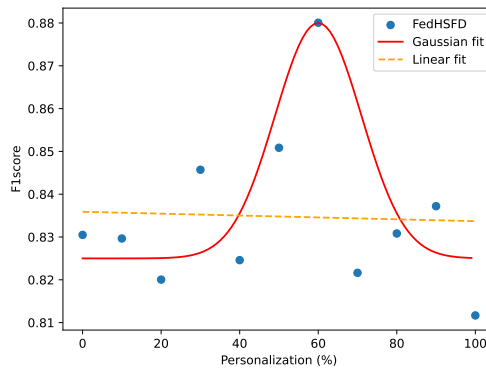


FIGURE 2.2 – F1-score vs. the personalization using our FedHSFD approach.

As we may see in this figure, the dashed line decreases slightly with the personalization increase, which means adding more ponderation, i.e. beyond 70%, to the LM reduces our approach performance. The continuous line shows that 60% of personalization provides the best results ; the F1-score is 88% which, we believe, is a good performance as it is difficult to detect HF and SF in a realistic context. If we choose lower personalization percentages, we fail to capture the personal traits specific to each client node. In contrast, if we choose higher percentages, we set aside the generalized GM.

Also, we compare the performance of our approach with other well-known aggregation strategies in the literature. FIGURE 2.3 displays the average F1-score vs. the personalization percentage of all the 15 client nodes for our FedHSFD approach compared with four other strategies, namely, FedAvg, QFedAvg, FedKrum, and FedAdam. It is evident that our approach provides the best performance with maximum and minimum F1-score values of 88% and 81.16%, at 60% and 100% of personalization, respectively.

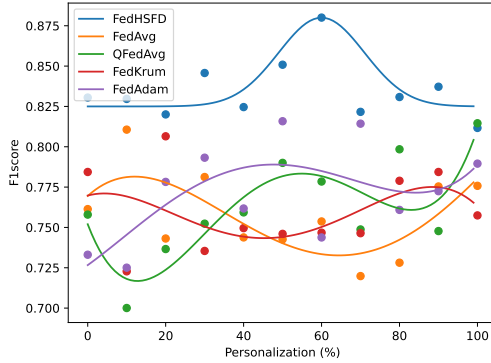


FIGURE 2.3 – F1-score for our approach and four other aggregation strategies.

Furthermore, we compared our approach with other suggested solutions in the literature using the same evaluation metrics. Several fall datasets are available (Sisfall, UP-Fall, HR-IMU, and Mobiact) and numerous research works are published [48, 49, 50, 51] using these datasets. In this comparison, we include cross-validation (CV) with 5 k-folds since it acts as a robustness indicator. Indeed, CV is often used to reduce over-fitting and drops the performance as a result.

As illustrated in TABLE 2.2, the comparison results are organized by dataset. Note that these datasets, the only ones available to us, don't use realistic fall context as we defined earlier in this paper. We compared the research works using the same datasets. The studies associated with Sisfall, UpFall, and Mobiact are based on ACC data only, whereas the study involving HR-IMU combines ACC and

TABLE 2.2 – Comparative study between the proposed approach and previous works.

Dataset	Paper	Approach	Model	F1score	AUC	Sensitivity	Specificity	Accuracy	Precision	Average performance
	[48]	FL	threshold-based	-	-	90.43	97.24	93.84	-	93.83
Sisfall	Our study (CV)	FL	XGB	-	-	92.44	99.17	96	-	95.87
	<b>Our study</b>	<b>FL</b>	<b>XGB</b>	-	-	<b>91.66</b>	<b>100</b>	<b>96.66</b>	-	<b>96.1</b>
	[49]	non-FL	KNN	-	-	98	98	97	-	97.66
UP Fall	Our study (CV)	FL	XGB	-	-	97.88	98.68	98.65	-	98.4
	<b>Our study</b>	<b>FL</b>	<b>XGB</b>	-	-	<b>99.66</b>	<b>98</b>	<b>98.78</b>	-	<b>98.81</b>
	[50]	non-FL	GMM	-	91.34	93.09	89.58	92.22	-	91.55
HR-IMU	Our study (CV)	FL	XGB	-	95.07	94.14	96.01	95.05	-	95.06
	<b>Our study</b>	<b>FL</b>	<b>XGB</b>	-	<b>97.94</b>	<b>95.88</b>	<b>100</b>	<b>98.18</b>	-	<b>98</b>
	[51]	FL	RL	~89	-	90	-	92.16	~88	~89.79
Mobiact	Our study (CV)	FL	XGB	89.29	-	91.66	-	93.03	90.4	91.09
	<b>Our study</b>	<b>FL</b>	<b>XGB</b>	<b>92.48</b>	-	<b>91.42</b>	-	<b>94.66</b>	<b>98.66</b>	<b>94.3</b>

HR data. The approach column indicates whether the research work uses FL or not, noted as FL and non-FL respectively. The dashes shown in the metrics columns indicate that the associated study did not use that metric. We use the average performance as the main criteria for comparison. It is clear that our approach outperforms the others, with and without CV.

## 2.5. Concluding Remarks and Future Works

In this paper, we leveraged FL as an alternative to traditional ML to enforce DP preservation. Hence, we introduced a novel FL approach to mitigate the impact of DH through a trade-off between generalization and personalization. In the process, we collected realistic fall data from ACC and HR sensors and used these data to compare our approach with “vanilla” aggregation strategies adopted by prior studies. Furthermore, we compared our approach with other FD research works using the same datasets and evaluation metrics. The comparison results confirmed the superiority of our approach. Though our contributions exhibit promising performance in the DH aspect, more effort could be put forward to improve the generalized and personalized GMs. A non-linear approach could be explored to aggregate the clients’ LMs and compute the personalized models. Indeed, non-linear approaches were proven to be more reliable when dealing with data imbued with outliers, and heteroscedasticity. Resolving these challenges can further enhance our approach’s capabilities for the future.

## Acknowledgement

This study is funded by the Natural Sciences and Engineering Research Council of Canada (DDG-2019-05756) as a grant to Dr. Jalal Almhana.

## Chapitre 3

# A Non-Linear Personalized Approach to Mitigate Poisoning Attack Coalitions in FL

# Abstract

Federated learning (FL) was introduced to mitigate security risks in traditional machine learning (ML) and deep learning (DL), particularly concerning data privacy (DP), through its decentralized architecture. However, FL remains susceptible to cyber-security threats. Such risks arise when several legitimate client nodes are transformed into a zombie network. These compromised nodes can launch large-scale attacks that affect the confidentiality, integrity, and availability (CIA) of the FL network, ultimately leading to a decline in prediction metrics for legitimate clients. While previous studies have focused on specific, small-scale attacks, none have addressed the impact of large-scale attacks, specifically, an `attack coalition`, where over 50% of the network is compromised, and multiple types of attacks occur simultaneously. Many existing studies rely on complex methods such as blockchain or malicious node detection, and only a few have considered the simpler alternative of `personalization`, though with limitations. In this paper, we introduce the concept of `attack coalition (AC)`, where malicious client nodes coordinate to undermine the FL system’s performance. We analyze the impact of this AC and propose a personalized FL approach to mitigate its effects. To validate our approach, we use a publicly available dataset measuring nurse stress levels, implemented through a deep neural network (DNN). Our results demonstrate that the AC severely reduces prediction metrics, with accuracy and F1-scores dropping to 18.34% and 18.68%, respectively. By applying our proposed method, these metrics improve significantly, reaching 99.36% and 98.46%, respectively.

## 3.1. Introduction

FL was introduced mainly to alleviate the security and DP concerns experienced in traditional ML/DL thanks to its distributed architecture. Indeed, traditional ML/DL solutions are beset with several cyber threats that affect privacy, integrity, and/or availability. The most widely recognized attacks are data breaches [52], adversarial attacks [53], and denial of service attacks (DoS) [54]. Although it doesn’t radically solve the aforementioned concerns, FL considerably reduces the risk of DP violation, poisoning, and DoS attacks thanks to its distributed structure [55]. Nonetheless, many loopholes remain unaddressed. Specifically, FL networks are susceptible to compromise by transforming a cluster of legitimate client nodes into a coalition of malicious ones capable of leading a large-scale raid featuring several types of cyber-attacks related to the CIA triad, thereby deteriorating the performance of the remaining legitimate client nodes. Prior research has been conducted in this regard [56, 57, 58]. However, these studies exhibited several limitations. First, their experiments depicted small-scale raids featuring one type of attack at

a time. Second, most previous works provided intricate solutions involving blockchain and malicious node detection. Few prior studies chose personalization as a straightforward yet effective approach [59, 60]. However, they used federated averaging (FedAvg) as an aggregation strategy. FedAvg may be a reliable alternative in most attack-free environments, however, it is unsuitable in attack-affected ones as it attributes similar weighting to all the client nodes including the malicious ones.

In this study, we introduce the concept of AC, which encompasses a coordinated effort by a coalition of malicious client nodes constituting over 50% of the FL network and launching a large-scale raid featuring multiple types of attacks simultaneously. In our case, the AC is achieved by converting multiple legitimate client nodes into malicious ones. These nodes launch a collective and diverse poisoning attack designed to corrupt the local models (LMs) and, consequently, the global model (GM), thereby affecting the local prediction metrics of the remaining legitimate client nodes. We analyze the impact of this AC on the network’s performance and propose a scalable personalized FL approach to mitigate its effect. We implement our approach for nurse stress level detection using a public dataset available on Kaggle. The main contributions of this paper are :

1. We introduce the general concept of AC, illustrating it through various case scenarios. For our approach, we focus on a particular scenario where we gradually increase the percentage of compromised client nodes from 14% to 57%. In this scenario, we launch a combination of data and model poisoning attacks involving label flipping, severe data imbalance, minimal data availability, and noisy LM gradients.
2. We analyze the impact of the AC and propose a personalized FL approach based on a non-linear aggregation strategy that neutralizes the effect of the malicious gradients on the server side and a GM/LM interpolation on the client side.
3. We implement our approach based on the nurse stress level detection dataset mentioned above. Our results demonstrate substantially diminished metrics following the AC, namely an average accuracy and F1-score of 18.34% and 18.68% respectively. Our approach, on the other hand, depicts a remarkable performance, specifically an average accuracy and F1-score of 99.36% and 98.46%, respectively.

The remainder of this paper is organized as follows. In Section 3.2., we review some previous studies conducted to mitigate data and model attacks in FL. Section 3.3. describes our proposed approach, followed by experimental results in Section 3.4. Finally, we conclude in Section 3.5.

## 3.2. Literature Review

In this section, we will focus on poisoning attacks, as they are employed in the AC studied in this paper. There are two types of poisoning attacks. Data poisoning involves tampering with data thus leading to model poisoning as well. Model poisoning, on the other hand, entails injecting perturbation into the GM and/or LM gradients. Both attacks instigate performance degradation in terms of prediction metrics. In this context, prior research works are divided into two categories : papers that only explored poisoning attacks and papers that proposed approaches to counter them. [56] and [61] fall into the first category, while the remaining papers cited in this section pertain to the latter. [56] introduced a support vector machine (SVM) approach to determine whether or not the dataset belonging to a client node, an edge device in this case, has been dirty-labeled. [61] proposed a novel data poisoning attack that created malicious gradients based on loss inversion. Those gradients were then used to generate poisoned labels which were later injected into the training dataset. [57] suggested a defense mechanism to mitigate the label-flipping problem in Sybil client nodes. [62] suggested a solution to counter adversarial attacks that involve corrupting the training records or injecting malicious noise-infested samples into the training dataset. [63] introduced four Byzantine-robust FL methods to reduce the impact of LM poisoning whereby an attacker compromises a group of client nodes and manipulates LM gradients during a training session. [64], on the other hand, generated several model poisoning attacks and demonstrated that the existing Byzantine-robust FL algorithms are not immune to them. [65] proposed an approach to counter model poisoning attacks by monitoring LM consistency over the training rounds. [58] addressed model poisoning whereby a byzantine client node crafts encrypted poisonous gradients using homomorphic encryption. [59] built a framework for personalized FL to address data and model poisoning attacks, namely, label poisoning, random updates, and model replacement. [66] used personalization to suppress model poisoning where client nodes send arbitrary LM gradients to the aggregator.

Although prior research studies have explored various attacks and occasionally suggested mitigations, there has been little to no attention to large-scale raids depicting significant sizes and executing multiple types of poisoning attacks simultaneously within a single training session. In addition, they relied on complex, costly technologies like blockchain and malicious node detection, which are challenging to deploy. Studies involving personalization used FedAvg, which is inadequate for handling data and model poisoning attacks. In this paper, we study the impact of a large-scale attack we call AC, where a coalition of malicious client nodes launch various data and model poisoning attacks. We analyze the AC’s effect by gradually increasing the percentage of malicious client nodes from 14% to 57%. We develop a

non-linear aggregation strategy that diminishes the impact of malicious gradients on the GM and enhance it with personalization on the client side.

### **3.3. Approach**

#### **A. Data**

We use the public “Nurse Stress Prediction Wearable Sensors” dataset. As the title indicates, this dataset leverages wearable sensor data including the acceleration, heart rate, electrodermal activity, and body temperature, all of which are organized in a dataframe of 11.5 million records. These records are collected over one week from 15 female nurses, aged 30 to 55 years, during their regular shifts at a hospital. The stress levels are categorized into three distinct labels : Level 0 reflects no stress, level 1 indicates mild stress, and level 2 implies severe stress.

#### **B. Selecting the appropriate DL model**

In this phase we implement a traditional DL configuration to select the appropriate DL model as well as the pre-processing and feature engineering techniques applied to the public dataset mentioned above, thereby ensuring optimal performance. We build a DNN that comprises two hidden layers encompassing 32 units and a Swish activation each. The output layer uses a Softmax activation for multi-label classification. For the training process, we adopt an Adam optimizer with a learning rate of 0.001. We reserve 80% of the dataset for training, 10% for validation, and 10% for testing. The number of epochs is 100. During training, we scale and prompt a batch of 10,000 samples for each iteration. This phase exhibits outstanding metrics : 99.99% of accuracy and F1-score. We add the F1-score since it provides a more accurate assessment of misclassifications compared to the accuracy.

#### **C. Establishing the FL architecture**

This phase serves as the reference for comparing metric variations in the subsequent phases. FL is a complex field with a vast research scope. Depending on the use case, three architecture types can be implemented : centralized, semi-decentralized, and fully decentralized [67]. Due to space limitations, we will focus on one architecture, specifically centralized FL, as illustrated in FIGURE 3.1. Other architectures will be explored in future work. In our simulation, we will include seven client nodes in the architecture. Note that our approach is scalable, and a higher number of nodes can be simulated without any difficulty.

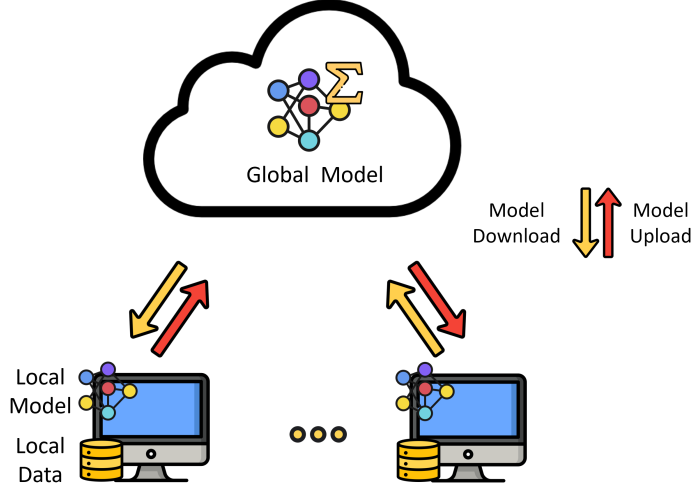


FIGURE 3.1 – FL architecture.

Accordingly, the FL training session is detailed in algorithm 3.1. The function *FedAvg* used in the algorithm is a well-known aggregation strategy in the literature [68]. As its name indicates, FedAvg calculates a pondered average of the received gradients as shown in equation (3.1) :

$$\Theta = FedAvg(n_i, \theta_i) = \sum_{i=1}^N \frac{n_i}{n} \theta_i \quad (3.1)$$

where  $n_i$  is the length of client  $i$ 's train set and  $n = \sum_{i=1}^N n_i$  is the length of all the clients train sets.

Ultimately, this phase, where there is no attack, shows remarkable metrics : an accuracy of 93.96% and an F1-score of 91.35%. These metrics are computed as the average across the seven client nodes and serve as a performance indicator for the subsequent phases.

---

**Algorithm 3.1** : FL training session

---

**Input :**     $N$  :                    Number of client nodes  
               $i \in \{1, \dots, N\}$  :    Client index  
               $n_i$  :                    Client  $i$ 's train set length  
               $r \in \mathbb{N}^*$  :                Number of rounds per training session  
**Output :**    $\Theta$  :                    The GM

**begin**

  Beginning of the training session by the server node.

**while** *Training do*

**if** *The central node has the GM  $\Theta$  from the last training session stored then*

      | Send  $\Theta$  to the client nodes.

**else**

      | Retrieve the initial gradients from a random client node.

      | Broadcast them to the client nodes.

**end**

**for**  $round \in \{1, \dots, r\}$  **do**

**for**  $i \in \{1, \dots, N\}$  *in parallel do*

        |  $\theta_i \leftarrow$  Local gradients after training.

        | Upload  $\theta_i$  and  $n_i$  to the server node.

**end**

$\Theta \leftarrow FedAvg(n_i, \theta_i)$

**for**  $i \in \{1, \dots, N\}$  *in parallel do*

        | Download  $\Theta$  for testing and prediction.

**end**

**end**

**end**

**return**  $\Theta$

**end**

---

## D. Implementing the AC

We define AC as a coordinated effort by a cluster of legitimate client nodes that have been converted into malicious ones by a hacker. These nodes work in unison by launching a cohort of attacks simultaneously to undermine the overall performance of the FL network in terms of confidentiality, integrity, and/or availability. Hence, the AC can manifest in three forms, each designed to disrupt the system's reliability :

- i. Form 1 : The malicious client nodes are divided into three clusters. Each cluster targets a distinct pillar of the CIA triad. The nodes of an individual cluster execute different types of attacks. For example, if a cluster aims to compromise confidentiality, a malicious client node can initiate a data breach

attack, while another could perform a LM inversion attack, etc.

- ii. Form 2 : In this form, the malicious client nodes are also divided into three clusters each targeting a distinct pillar of the CIA triad. The only difference is that the nodes of an individual cluster execute the same attack. For instance, if a cluster aims to undermine the integrity of the GM, all its nodes could perform a data poisoning attack.
- iii. Form 3 : In this form, all the malicious client nodes target the same aspect of the CIA triad but perform different types of attacks. For instance, if the objective is to disrupt the availability of legitimate client nodes, a malicious client node can initiate an Internet Control Message Protocol (ICMP) flood, another can execute a SYN flood, and another can carry out a SYN-ACK flood, etc [69].

In this paper, we limit our research to the third form of AC. Other forms can be studied in a similar way. Specifically, we target the integrity of the GM by carrying out the following data and model poisoning attacks :

- i. Label flipping : The malicious client nodes swap two of the training labels in their dataset and leave the testing labels intact.
- ii. Severe data imbalance : The malicious client nodes manipulate their training sets to disproportionately lower the number of samples in one of their labels. This action injects bias towards the dominant labels.
- iii. Minimal data availability : The malicious client nodes contribute with extremely small datasets that induce overfitting to their LM gradients.
- iv. Noisy LM gradients : The malicious client nodes inject Gaussian noise into the LM gradients resulting from training with legitimate data.

We analyze the AC’s impact on legitimate clients’ performance in terms of accuracy and F1-score by gradually increasing the size of attacks, from 14% to 57%, expressed here as the percentage of malicious client nodes. Note that the FL training session and architecture remain the same as defined in algorithm 3.1 and FIGURE 3.1 respectively.

TABLE 3.1 summarizes the status of client nodes across various levels of attack, ranging from 0% to 57%. As shown in the table, when the AC’s size is 0% i.e. no attack, each client node has the following label distribution : 309287 entries in level 0, 115062 entries in level 1, and 1219802 entries in level 2. As we gradually increase the percentage of malicious client nodes from 14% i.e 1/7 to 57% i.e 4/7, each one of them launches a distinct attack as shown in TABLE 3.1. In the label flipping attack, the label distribution remains unchanged, however, we swap level 0 and level 2. In the imbalance attack, the label distribution is the following :

123851 entries in level 0, 1153 entries in level 1, and 1219208 entries in level 2. In the minimal data availability attack, we lower the number of entries to 300 per label. In the model poisoning attack, we add random Gaussian noise to the LM gradients with a mean value of 1000 and a standard deviation of 10000.

TABLE 3.1 – Client nodes’ status at each level of attack

Size	Status of the client node				
	0%	14%	29%	43%	57%
Node No.					
1	Legitimate	Legitimate	Legitimate	Legitimate	Legitimate
2	Legitimate	Legitimate	Legitimate	Legitimate	Legitimate
3	Legitimate	Legitimate	Legitimate	Legitimate	Legitimate
4	Legitimate	Label flipping	Label flipping	Label flipping	Label flipping
5	Legitimate	Legitimate	Severe imbalance	Severe imbalance	Severe imbalance
6	Legitimate	Legitimate	Legitimate	Minimal data availability	Minimal data availability
7	Legitimate	Legitimate	Legitimate	Legitimate	Noisy LM gradients

## E. Implementing our solution to counter the effect of the AC

We develop a personalized FL solution to mitigate the impact of the aforementioned AC. The training session is depicted in algorithm 3.2. As shown in the algorithm, our approach consists of two primary components : The proposed aggregation strategy and personalization which are encapsulated by functions  $\Psi$  and  $\Gamma$  respectively. The function  $\Psi$  is a Gaussian transformation of a linear combination presented in equation (3.2) :

$$\Theta = \Psi(f_i, \theta_i, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\chi(f_i, \theta_i) - \mu}{\sigma}\right)^2} \quad (3.2)$$

where



gregation strategy leverages the advantages of non-linear transformation. Indeed, applying the appropriate transformation enhances both the linearity of relationships and the stability of variability [70]. The choice of the Gaussian distribution is based on its symmetric and bell-shaped curve which is dependent on the mean  $\mu$  and the standard deviation  $\sigma$ . One of the greatest advantages of the Gaussian distribution is the central limit theorem which, under certain circumstances, states that the sum of a large number of random variables -in our case, the LM gradients- will tend to follow a Gaussian distribution [71]. This makes it an evident choice for modeling a wide range of inquiries including ours. Furthermore, in the linear combination exhibited in equation (3.3), we weigh the LM gradients with the F1-score that we compute based on a portion of evaluation data stored on the server side to avoid the fraudulence of malicious client nodes. As a result, it demonstrates high values for legitimate clients and lower values for malicious ones.

We further enhance our proposed strategy with personalization on the client side. Hence, we use the model interpolation method we proposed in a previous study [72] and which is defined in equation (3.4) :

$$P_i = \Gamma(\theta_i, \Theta, \lambda) = \lambda\theta_i + (1 - \lambda)\Theta \quad (3.4)$$

The parameter  $\lambda$  is the personalization coefficient which encapsulates the trade-off between the LM and the GM. It is defined as follows :

$$\left\{ \begin{array}{l} \text{If } \lambda < 0.5, \quad \Theta \text{ has more weight than } \theta_i \\ \text{Else if } \lambda = 0.5, \quad \Theta \text{ and } \theta_i \text{ have the same weight} \\ \text{Else, } \quad \theta_i \text{ has more weight than } \Theta \end{array} \right.$$

Hereafter, we will refer to  $\lambda$  as the personalization percentage to avoid ambiguity in the experimental results.

### 3.4. Experimental Results

We present the experimental results based on the phases detailed in our approach. Since our client nodes are imbalanced, using only the accuracy for evaluation is insufficient, which is why we add the F1-score [73]. Ultimately, the evaluation is based on the test set.

#### A. Impact of the AC on the accuracy and F1-score

FIGURE 3.2 illustrates the impact of the AC as a function of the attack size expressed as a percentage of malicious client nodes. We can see that both metrics

gradually decrease with the increase of the AC's size. The impact is significant at 43% with an accuracy and F1-score dropping to 88% and 75% respectively. It is damaging for both metrics beyond 50% and it will certainly disrupt the functioning of the FL network.

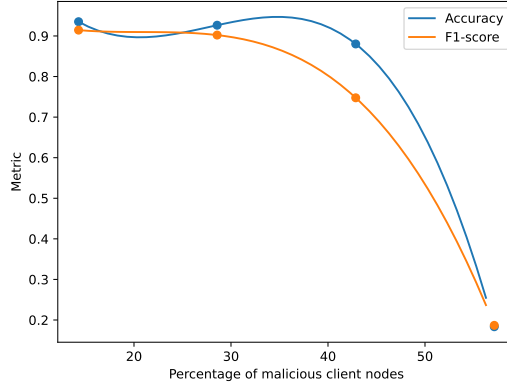


FIGURE 3.2 – Impact of the AC as a function of its size on the accuracy and F1-score.

## B. Proposed solution to counter the AC's damaging effect

Figures 3.3a and 3.3b illustrate how our solution enhances the accuracy and F1-score for legitimate client nodes as a function of the level of personalization. We evaluated five different values of the standard deviation  $\sigma$  within the aggregation strategy we proposed earlier.

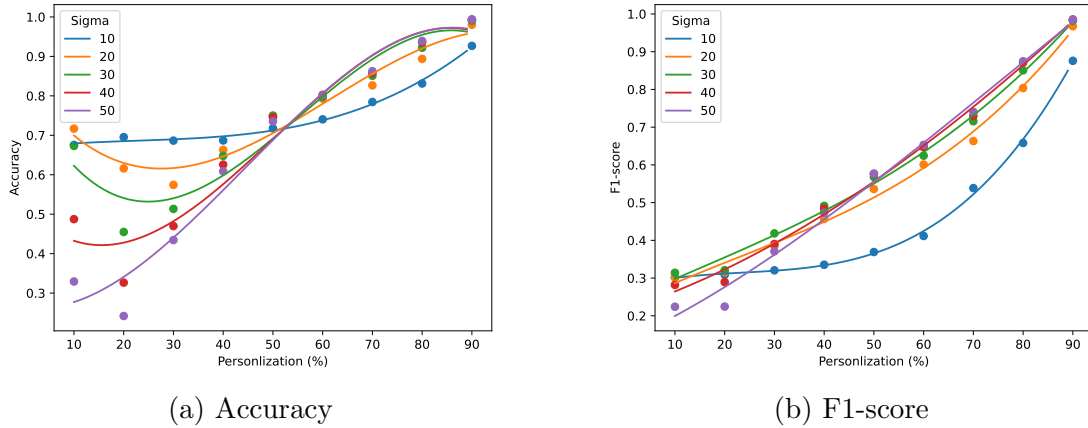


FIGURE 3.3 – Average accuracy and F1-score for legitimate client nodes vs. the personalization percentage  $\lambda$ .

As can be seen, the performance improves as the level of personalization increases, which is expected since the poison injected into the GM gradually diminishes. Our solution provides the best results when  $\sigma = 50$  and the personalization is at its maximum. At 90% personalization, the average values for the accuracy and F1-score are 99.36% and 98.46%, respectively. This represents a significant improvement compared to the results obtained in FIGURE 3.2.

### C. Comparison with the traditional aggregation strategy FedAvg

We compare the best results of our proposed aggregation strategy ( $\sigma = 50$ ) to FedAvg, a commonly used method in the literature. FIGURE 3.4 shows the performance in terms of accuracy and F1-score for both aggregation strategies.

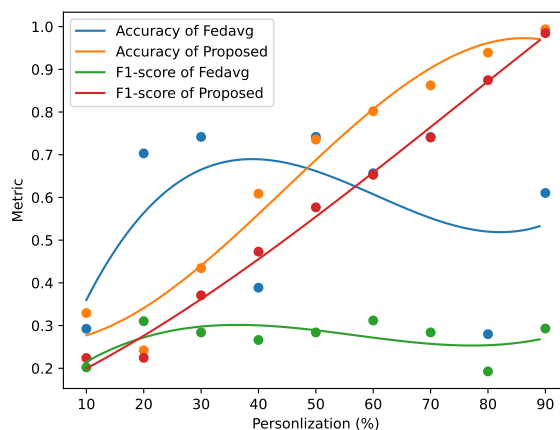


FIGURE 3.4 – Comparison of our aggregation strategy with FedAvg for both metrics : accuracy and F1-score.

It is evident that our aggregation strategy yields better results. Furthermore, both FedAvg metrics exhibit noticeable fluctuation with the increasing personalization, a trend that is not observed with our aggregation strategy.

### 3.5. Concluding Remarks and Future Works

In this paper, we studied the impact of a major security attack, which we defined as AC, on the accuracy and F1-score of the FL model. We proposed a new solution to mitigate the negative impact of this attack, which includes non-linear aggregation and the use of personalization. Our experimental results show the effectiveness of our proposed solution. In future work, we will study other

attack scenarios as well as different nonlinear aggregation techniques to improve performance metrics at low levels of personalization.

## **Acknowledgement**

This study is funded by the Natural Sciences and Engineering Research Council of Canada (DDG-2019-05756) as a grant to Dr. Jalal Almhana.

# Conclusion Générale

Dans cette thèse, nous avons adopté l'apprentissage fédéré comme une alternative plus efficace à l'apprentissage automatique traditionnel pour préserver la confidentialité des données. Nous avons ainsi exploré deux problématiques pressantes : l'hétérogénéité des données et la sécurité, dans le but de développer des applications médicales, notamment la détection des chutes et l'évaluation du niveau de stress chez les infirmières. Nos résultats ont montré l'efficacité de nos approches pour traiter ces problématiques.

Dans le chapitre 1, nous avons proposé une approche réaliste pour détecter les chutes brutales, les pré-chutes et les chutes lentes dans le cadre d'activités contextuelles telles que la course et la marche. Nos résultats ont montré qu'il n'est pas possible d'obtenir des taux de détection aussi élevés que ceux rapportés dans les travaux publiés dans la littérature.

Dans le chapitre 2, nous avons introduit une nouvelle approche d'apprentissage fédéré pour atténuer l'impact de l'hétérogénéité des données, en établissant un compromis entre généralisation et personnalisation, notamment dans le cadre de la détection des chutes. Nos résultats ont démontré l'efficacité de notre approche par rapport à celles publiées dans la littérature.

Dans le chapitre 3, nous avons introduit le concept de coalition d'attaques, où un groupe de nœuds clients malveillants lance simultanément plusieurs types d'attaques. Nous avons analysé l'impact de ces attaques et proposé une approche d'apprentissage fédéré personnalisée et non linéaire pour les atténuer.

Il est certain qu'il reste encore beaucoup de choses à explorer. Cependant, si nous nous limitons aux problèmes traités dans cette thèse, nous pensons qu'il existe certains aspects à approfondir :

1. Amélioration de la prédiction des chutes brutales et lentes, dans un contexte réaliste.
2. Expérimentation avec d'autres transformations non linéaires pour améliorer

la performance des stratégies d'agrégation, ainsi que l'exploration d'autres méthodes de personnalisation.

3. Exploration plus approfondie du concept de coalition d'attaques, y compris les différentes formes et types d'attaques qui affectent la confidentialité, la disponibilité et/ou l'intégrité.
4. Étude de la possibilité de déployer les solutions proposées dans cette thèse dans un cas réel.

# Sigles Abréviations

AI :	Artificial Intelligence
IoT :	Internet of Things
ML :	Machine Learning
DL :	Deep Learning
DP :	Data Privacy
FL :	Federated Learning
FD :	Fall Detection
ADL :	Activities of Daily Living
DH :	Data Heterogeneity
IID :	Independent and Identically Distributed
FedAvg :	Federated Averaging
UML :	Unified Modeling Language
CB :	CatBoost
DT :	Decision Tree
RF :	Random Forest
XGB :	XGBoost
WHO :	World Health Organisation
PFD :	Pre-Fall Detection
SFD :	Slow Fall Detection
KNN :	K-Nearest Neighbors
LSTM :	Long Short-Term Memory
Bi-LSTM :	Bidirectional LSTM
SVM :	Support Vector Machines

CNN :	Convolutional Neural Network
DNN :	Deep Neural Network
HF :	hard Fall
PF :	Pre-Fall
SF :	Slow Fall
ACC :	Accelerometer
HR :	Heart Rate
OC-SVM :	One-Class Support Vector Machines
RL :	Reinforcement Learning
GM :	Global Model
LM :	Local Model
CV :	Cross-Validation
CIA :	Confidentiality, Integrity, Availability
AC :	Attack Coalition
DNN :	Deep Neural Network
DoS :	Denial of Service
ICMP :	Internet Control Message Protocol

# Liste des tableaux

1	Bibliothèques et frameworks utilisés dans cette thèse. . . . .	15
1.1	A description of the dataset in consideration in this paper. . . . .	22
1.2	Algorithms used and their characteristics. . . . .	23
1.3	Algorithms' ability for PF detection. . . . .	27
1.4	HF and PF detection for the algorithms used. . . . .	27
1.5	Accuracy of SF. . . . .	28
1.6	HF detection accuracy compared to previous works. . . . .	28
1.7	PF detection accuracy compared to previous works. . . . .	28
2.1	Client nodes' description. . . . .	41
2.2	Comparative study between the proposed approach and previous works. . . . .	43
3.1	Client nodes' status at each level of attack . . . . .	53

# Table des figures

1	Topologie traditionnelle de l'apprentissage automatique. . . . .	10
2	Architecture générale de l'apprentissage fédéré. . . . .	11
1.1	A walking ADL signal from [28]. . . . .	21
1.2	A non-realistic-context HF signal from [8]. . . . .	21
1.3	An HF signal in a realistic context. . . . .	22
1.4	Various views of partial representations of the HF shown in a dashed line. . . . .	23
1.5	An example of four fall signals within an ADL. . . . .	24
1.6	Samples of the HF and SF data. . . . .	24
1.7	A sample of the generated SF data. . . . .	25
1.8	An ADL signal including SF. . . . .	26
1.9	The accuracy vs partial inclusion (noted as a percentage) of HF signals for several classification algorithms. . . . .	26
2.1	The general architecture of FedHSFD. . . . .	35
2.2	F1-score vs. the personalization using our FedHSFD approach. . . . .	41
2.3	F1-score for our approach and four other aggregation strategies. . . . .	42
3.1	FL architecture. . . . .	50
3.2	Impact of the AC as a function of its size on the accuracy and F1-score. . . . .	56
3.3	Average accuracy and F1-score for legitimate client nodes vs. the personalization percentage $\lambda$ . . . . .	56
3.4	Comparison of our aggregation strategy with FedAvg for both metrics : accuracy and F1-score. . . . .	57

# Liste des Algorithmes

2.1	FedHSFD training session . . . . .	36
3.1	FL training session . . . . .	51
3.2	Personalized FL training session . . . . .	54

# Bibliographie

- [1] Y. Chen, “Iot, cloud, big data and ai in interdisciplinary domains,” p. 102070, 2020.
- [2] S. Majumder, T. Mondal, and M. J. Deen, “Wearable sensors for remote health monitoring,” *Sensors*, vol. 17, no. 1, p. 130, 2017.
- [3] G. Drainakis, K. V. Katsaros, P. Pantazopoulos, V. Sourlas, and A. Amditis, “Federated vs. centralized machine learning under privacy-elastic users : A comparative analysis,” in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*. IEEE, 2020, pp. 1–8.
- [4] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, “A survey on federated learning,” *Knowledge-Based Systems*, 2021.
- [5] S. Besrouer, “Rapport Technique : Generalization vs Personalization : A Trade-off for Better Data Heterogeneity impact Mitigation in FL,” Tech. Rep.
- [6] “Heart rate monitors : Armband + chest strap — wahoo fitness canada,” <https://ca.wahoofitness.com/devices/heart-rate-monitors>, (Accessed on 09/10/2024).
- [7] “Metamotions – mbientlab,” <https://mbientlab.com/metamotions/>, (Accessed on 09/10/2024).
- [8] A. T. Özdemir and B. Barshan, “Detecting falls with wearable sensors using machine learning techniques,” *Sensors*, vol. 14, no. 6, pp. 10 691–10 708, 2014.
- [9] “Nurse stress prediction wearable sensors,” <https://www.kaggle.com/datasets/priyankraval/nurse-stress-prediction-wearable-sensors>, (Accessed on 09/10/2024).
- [10] “World health organization (who),” <https://www.who.int/en>, (Accessed on 03/24/2023).
- [11] T. de Quadros, A. E. Lazzaretti, and F. K. Schneider, “A movement decomposition and machine learning-based fall detection system using wrist wearable device,” *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5082–5089, 2018.
- [12] T. Vaiyapuri, E. L. Lydia, M. Y. Sikkandar, V. G. Díaz, I. V. Pustokhina, and D. A. Pustokhin, “Internet of things and deep learning enabled elderly

- fall detection model for smart homecare,” *IEEE Access*, vol. 9, pp. 113 879–113 888, 2021.
- [13] F. Bagala, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk, “Evaluation of accelerometer-based fall detection algorithms on real-world falls,” *PloS one*, vol. 7, no. 5, p. e37062, 2012.
- [14] R. Igual, C. Medrano, and I. Plaza, “A comparison of public datasets for acceleration-based fall detection,” *Medical engineering & physics*, vol. 37, no. 9, pp. 870–878, 2015.
- [15] P. Vallabh, R. Malekian, N. Ye, and D. C. Bogatinoska, “Fall detection using machine learning algorithms,” in *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2016, pp. 1–9.
- [16] A. Howedi, A. Lotfi, and A. Pourabdollah, “Accelerometer-based human fall detection using fuzzy entropy,” in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–7.
- [17] J.-S. Lee and H.-H. Tseng, “Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers,” *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8293–8302, 2019.
- [18] C. Chatzaki, M. Pediaditis, G. Vavoulas, and M. Tsiknakis, “Human daily activity and fall recognition using a smartphone’s acceleration sensor,” in *Information and Communication Technologies for Ageing Well and e-Health : Second International Conference, ICT4AWE 2016, Rome, Italy, April 21-22, 2016, Revised Selected Papers 2*. Springer, 2017, pp. 100–118.
- [19] D. Ajerla, S. Mahfuz, and F. Zulkernine, “A real-time patient monitoring framework for fall detection,” *Wireless Communications and Mobile Computing*, vol. 2019, pp. 1–13, 2019.
- [20] K. Kim, G. Yun, S.-K. Park, and D. H. Kim, “Fall detection for the elderly based on 3-axis accelerometer and depth sensor fusion with random forest classifier,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 4611–4614.
- [21] S. Ghosh and S. K. Ghosh, “Feel : Federated learning framework for elderly healthcare using edge-iomt,” *IEEE Transactions on Computational Social Systems*, 2023.
- [22] G. S. Mubibya, J. Almhana, and Z. Liu, “Efficient fall detection using bidirectional long short-term memory,” in *2023 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2023, pp. 983–988.
- [23] O. Aziz, C. M. Russell, E. J. Park, and S. N. Robinovitch, “The effect of window size and lead time on pre-impact fall detection accuracy using support vector machine analysis of waist mounted inertial sensor data,” in *2014 36th*

- Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 30–33.
- [24] N. Otanasap, “Pre-impact fall detection based on wearable device using dynamic threshold model,” in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. IEEE, 2016, pp. 362–365.
- [25] S. Liang, T. Chu, D. Lin, Y. Ning, H. Li, and G. Zhao, “Pre-impact alarm system for fall detection using mems sensors and hmm-based svm classifier,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 4401–4405.
- [26] L. Wang, M. Peng, and Q. Zhou, “Pre-impact fall detection based on multi-source cnn ensemble,” *IEEE Sensors Journal*, vol. 20, no. 10, pp. 5442–5451, 2020.
- [27] X. Chen, S. Jiang, and B. Lo, “Subject-independent slow fall detection with wearable sensors via deep learning,” in *2020 IEEE SENSORS*, 2020, pp. 1–4.
- [28] G. S. Mubibya, S. Besrou, and J. Almhana, “A real-time iot system and ml algorithms : A comparative study,” in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 5262–5267.
- [29] G. S. Mubibya and J. Almhana, “Improving human activity recognition using ml and wearable sensors,” in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 165–170.
- [30] F. A. S. Ferreira de Sousa, C. Escriba, E. G. Avina Bravo, V. Brossa, J.-Y. Fourniols, and C. Rossi, “Wearable pre-impact fall detection system based on 3d accelerometer and subject’s height,” *IEEE Sensors Journal*, vol. 22, no. 2, pp. 1738–1745, 2022.
- [31] S. Abbate, M. Avvenuti, P. Corsini, J. Light, and A. Vecchio, “Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network : a survey,” *Wireless Sensor Networks : Application-Centric Design*, vol. 1, 2010.
- [32] M. Merenda, C. Porcaro, and D. Iero, “Edge machine learning for ai-enabled iot devices : A review,” *Sensors*, vol. 20, no. 9, p. 2533, 2020.
- [33] S. P. Amaraweera and M. N. Halgamuge, “Internet of things in the healthcare sector : overview of security and privacy issues,” *Security, privacy and trust in the IoT environment*, pp. 153–179, 2019.
- [34] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, and B. Yoon, “A framework for privacy-preservation of iot healthcare data using federated learning and blockchain technology,” *Future Generation Computer Systems*, vol. 129, pp. 380–388, 2022.

- [35] J. A. Stevens, P. S. Corso, E. A. Finkelstein, and T. R. Miller, “The costs of fatal and non-fatal falls among older adults,” *Injury prevention*, 2006.
- [36] S. N., M. M. Islam, and F. A. Sharna, “An iot based device-type invariant fall detection system,” *Internet of Things*, 2020.
- [37] A. K. Bourke, J. O’Brien, and G. M. Lyons, “Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm,” *Gait & posture*, vol. 26, no. 2, pp. 194–199, 2007.
- [38] S. Besrou, G. S. Mubibya, Z. Liu, and J. Almhana, “Context-aware hard and slow fall detection,” in *2024 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2024, pp. 321–326.
- [39] M. P. Tan and R. A. Kenny, “Cardiovascular assessment of falls in older people,” *Clinical interventions in aging*, 2006.
- [40] H. Li, Z. Cai, J. Wang, J. Tang, W. Ding, C. T. Lin, and Y. Shi, “Fedtp : Federated learning by transformer personalization,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [41] Z. Yu, J. Liu, M. Yang, Y. Cheng, J. Hu, and X. Li, “An elderly fall detection method based on federated learning and extreme learning machine (fed-elm),” *IEEE Access*, 2022.
- [42] 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*, 2023.
- [43] K. Kirsten, B. Pfitzner, L. Löper, and B. Arnrich, “Sensor-based obsessive-compulsive disorder detection with personalised federated learning,” in *IEEE ICMLA*, 2021.
- [44] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, “Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks,” *IEEE Transactions on Information Forensics and Security*, 2010.
- [45] A. Rácz, D. Bajusz, and K. Héberger, “Effect of dataset size and train/test split ratios in qsar/qspr multiclass classification,” *Molecules*, 2021.
- [46] T. M. Padmaja, N. Dhulipalla, R. S. Bapi, and P. R. Krishna, “Unbalanced data classification using extreme outlier elimination and sampling techniques for fraud detection,” in *ICACCT*, 2007.
- [47] V. Vovk, *The Fundamental Nature of the Log Loss Function*, 2015.
- [48] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, “Sisfall : A fall and movement dataset,” *Sensors*, 2017.

- [49] M. J. A. Nahian, T. Ghosh, M. H. A. Banna, M. A. Aseeri, M. N. Uddin, M. R. Ahmed, M. Mahmud, and M. S. Kaiser, "Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features," *IEEE Access*, 2021.
- [50] Y. H. Nho, J. G. Lim, and D. S. Kwon, "Cluster-analysis-based user-adaptive fall detection using fusion of heart rate sensor and accelerometer in a wearable device," *IEEE Access*, 2020.
- [51] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome : Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, 2022.
- [52] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science : an overview from machine learning perspective," *Journal of Big data*, 2020.
- [53] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text : A review," *International Journal of Automation and Computing*, 2020.
- [54] M. Abdullahi, Y. Baashar, H. Alhussian, A. Alwadain, N. Aziz, L. F. Capretz, and S. J. Abdulkadir, "Detecting cybersecurity attacks in internet of things using artificial intelligence methods : A systematic literature review," *Electronics*, 2022.
- [55] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning : Attacks and defenses," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [56] R. Doku and D. B. Rawat, "Mitigating data poisoning attacks on a federated learning-edge computing network," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021.
- [57] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv :1808.04866*, 2018.
- [58] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "Shieldfl : Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, 2022.
- [59] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto : Fair and robust federated learning through personalization," in *International conference on machine learning*, 2021.
- [60] T. T. Thein, Y. Shiraishi, and M. Morii, "Personalized federated learning-based intrusion detection system : Poisoning attack and defense," *Future Generation Computer Systems*, 2024.

- [61] P. Gupta, K. Yadav, B. B. Gupta, M. Alazab, and T. R. Gadekallu, “A novel data poisoning attack in federated learning based on inverted loss function,” *Computers & Security*, 2023.
- [62] Y. Chang, S. Laridi, Z. Ren, G. Palmer, B. W. Schuller, and M. Fisichella, “Robust federated learning against adversarial attacks for speech emotion recognition,” *arXiv preprint arXiv :2203.04696*, 2022.
- [63] M. Fang, X. Cao, J. Jia, and N. Gong, “Local model poisoning attacks to {Byzantine-Robust} federated learning,” in *29th USENIX security symposium (USENIX Security 20)*, 2020.
- [64] V. Shejwalkar and A. Houmansadr, “Manipulating the byzantine : Optimizing model poisoning attacks and defenses for federated learning,” in *NDSS*, 2021.
- [65] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, “Fldetector : Defending federated learning against model poisoning attacks via detecting malicious clients,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.
- [66] R. Ye, Z. Ni, F. Wu, S. Chen, and Y. Wang, “Personalized federated learning with inferred collaboration graphs,” in *International Conference on Machine Learning*, 2023.
- [67] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, “Decentralized federated learning : Fundamentals, state of the art, frameworks, trends, and challenges,” *IEEE Communications Surveys & Tutorials*, 2023.
- [68] T. Sun, D. Li, and B. Wang, “Decentralized federated averaging,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [69] S. R. Ghanti and G. Naik, “Design of system on chip for generating syn flood attack to test the performance of the security system,” 2015.
- [70] N. Shachar, A. Mitelpunkt, T. Kozlovski, T. Galili, T. Frostig, B. Brill, M. Marcus-Kalish, Y. Benjamini *et al.*, “The importance of nonlinear transformations use in medical data analysis,” *JMIR medical informatics*, 2018.
- [71] G. A. Brosamler, “An almost everywhere central limit theorem,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, 1988.
- [72] S. Besrou, G. S. Mubibya, C. Ben Abdeljelil, and J. Almhana, “Generalization vs Personalization : A Trade-off for better Data Heterogeneity impact Mitigation in FL,” in *IEEE Global Communications Conference (GLOBECOM)*, Cape Town, South Africa, 2024.
- [73] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, S. Huang, M. Brooks, M. J. Lee, and H. Asadi, “Peering into the black box of artificial intelligence : evaluation metrics of machine learning methods,” *American Journal of Roentgenology*, 2019.