Resilient and Communication-Efficient Federated Learning for Scalable Heterogeneous IoT Networks on Grassmann Manifolds

Mr. Tung Anh Nguyen
B.Sc.(Vietnam), M.Sc.(South Korea)

A thesis submitted to fulfil requirements for the degree of Doctor of Philosophy

Doctor of Philosophy (PhD)

Supervisor:
Associate Professor. Nguyen Hoang Tran

The University of Sydney
School of Computer Science
Faculty of Engineering

2025

Statement of Originality

This thesis is submitted to the School of Computer Science, Faculty of Engineering, the University of Sydney in fulfillment of the requirements for the degree of Doctor of Philosophy.

I certify that to the best of my knowledge and belief, the content of this thesis is original except as acknowledged in the text. The work presented in this thesis has not been submitted to any institution for a degree or other purposes.

Tung-Anh Nguyen
School of Computer Science
Faculty of Engineering
The University of Sydney

Acknowledgements

I would like to express my deepest gratitude to my principal supervisor, Associate Professor Nguyen H. Tran, for his unwavering support, thoughtful guidance, and inspiring encouragement throughout my research journey. It has been an honor to learn from such an exceptional mentor whose wisdom, insight, and dedication have shaped my growth as a researcher and as an individual. His belief in my abilities has been a constant source of motivation, and I am truly thankful for his kindness and mentorship.

I am also sincerely grateful to my co-supervisor, Associate Professor Bing Zhou, for his invaluable advice, patient guidance, and continuous encouragement. My heartfelt thanks go to Dr. Wei Bao for his thoughtful insights, engaging discussions, and generous support. Additionally, I would like to express my deep gratitude to Dr. Ta Thi Kim Hue, whose guidance and mentorship have been pivotal in shaping my career and personal growth.

To my colleagues, I extend my warmest thanks for their friendship and support. Their encouragement, shared knowledge, and teamwork have made this challenging journey not only manageable but also rewarding and memorable.

I am profoundly thankful for the financial support provided by Engineering Research Scholarship (ERS) and the Post-Graduate Research Support Scheme (PRSS) from The University of Sydney. This generosity has allowed me to fully dedicate myself to my research and achieve my goals.

This thesis is dedicated to my family: my parents, my parents-in-law, my younger brother, my wife, and my little daughter, Diep Anh. Their unconditional love, unwavering belief in me, and endless encouragement have been my greatest source of strength and inspiration. They have motivated me to pursue my research and contribute, in my own small way, to the betterment of the world.

Abstract

The accelerated expansion of mobile devices and Internet of Things (IoT) technologies has led to an unprecedented surge in data generation at the network edge. This shift has transformed the paradigm of "big data" creation and storage, transitioning from centralized data centers to decentralized edge networks. As a result, there is a growing trend toward edge intelligence, where artificial intelligence and machine learning algorithms are deployed directly on user equipment (UE) and localized data sources. Unlike traditional distributed ML techniques, which require data to be collected and processed centrally, a novel framework known as Federated Learning (FL) empowers UEs to conduct local learning on their data, driving the evolution of edge intelligence.

The conventional FL framework consists of a network of clients, such as users, edge devices, or workers, connected to a central server. Its primary objective is to collaboratively construct a global model by integrating contributions from distributed datasets and computational resources. Instead of transferring raw data, clients transmit locally trained models to the server, where they are aggregated to update the global model. This process not only reduces network bandwidth consumption but also safeguards user privacy by keeping sensitive data localized. However, while FL offers significant benefits, it also faces major challenges in IoT settings due to the dynamic and heterogeneous nature of IoT data, which often appears as spatio-temporal multivariate time series. Unlike conventional FL applications that focus on image or text data, IoT data streams are continuous, high-dimensional, and subject to non-stationary patterns caused by changing device behaviors, network conditions, and attack surfaces. These challenges are particularly pronounced in IoT networks, where devices are constrained by limited resources. This thesis addresses these interconnected challenges by focusing on three critical areas: improving communication efficiency, enhancing the effectiveness of intrusion detection, and developing models capable of capturing the dynamic nature of IoT data.

The first contribution of this thesis is the development of a computationally and communication-efficient intrusion detection framework tailored for IoT networks. To achieve this, we introduce the Federated PCA framework, which is designed to profile both normal and abnormal behaviors of IoT devices while maintaining data privacy. Central to this framework is the application of gradient-based optimization on Grassmann manifolds, integrated within the consensus Alternating Direction Method of Multipliers framework. This methodological integration not only ensures computationally efficient model training but also enables real-time anomaly detection, making the framework particularly well-suited for privacy-conscious IoT environments characterized by resource constraints. Moreover, the proposed approaches are supported by theoretical analysis, including under a sub-sampling scheme—an innovative

contribution that extends their applicability to real-world distributed environments. Additionally, the computational complexity of the Grassmann manifold-based approach has been thoroughly analyzed, confirming its feasibility for deployment on hardware-constrained IoT devices. Empirical evaluations further validate the theoretical findings, showing that the Federated PCA framework achieves anomaly detection performance comparable to advanced non-linear methods. Notably, the framework delivers significant enhancements in both communication and memory efficiency, underscoring its potential as a robust and practical solution for securing IoT networks against adversarial threats while effectively addressing the challenges of limited resources.

Building on the spatial compression capabilities of Grassmann manifolds, the second contribution of this thesis addresses the dynamic and evolving nature of IoT data by integrating Koopman operator theory. This approach bridges spatial insights with temporal dynamics, offering a unified framework to analyze spatio-temporal dependencies in non-stationary multivariate time series. To achieve this, we propose Federated Koopman Learning (`FedKoop`), a novel methodology that leverages Koopman operators to model the intricate dynamics of IoT data. By incorporating gradient-based optimization on Grassmann manifolds, `FedKoop` achieves high forecasting accuracy while significantly reducing computational and memory demands. This design makes it particularly well-suited for large-scale cyber-physical systems. In addition, the thesis introduces Federated Koopman Learning for Wireless Traffic Prediction (`FedKooL`), which applies the Koopman-based framework to resolve a critical challenge of IoT networks. Through optimized low-rank modeling on Grassmann manifolds and the integration of advanced techniques such as Rockafellar's envelope, `FedKooL` demonstrates notable improvements in predictive accuracy and operational efficiency. Empirical evaluations on real-world datasets validate the effectiveness of this approach, highlighting its potential for managing wireless traffic in dynamic and resource-constrained IoT environments.

Statement of Authorship Attribution

This thesis contains several chapters with material that was previously published during my Ph.D. candidature under the supervision of Associate Professor Nguyen Hoang Tran. The ideas and the preparation behind each publication is primarily my own work, with all assistance that I received stated and acknowledged below:

Journal Articles

1. **Tung-Anh Nguyen**, Long Tan Le, Tuan Dung Nguyen, Wei Bao, Suranga Seneviratne, Choong Seon Hong, Nguyen H. Tran: "Federated PCA on Grassmann Manifold for IoT Anomaly Detection," in *IEEE/ACM Transactions on Networking*, vol. 32, no. 5, pp. 4456-4471, Oct. 2024 [1]. This research is reported in Chapter 3. I designed the research, conducted the research, did experiments, and wrote the paper.

Conference Papers

1. **Tung-Anh Nguyen**, Jiayu He, Long Tan Le, Wei Bao, Nguyen H. Tran: "Federated PCA on Grassmann Manifold for Anomaly Detection in IoT Networks," *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1-10 [2]. This research is reported in Chapter 3. I designed the research, conducted the research, did experiments, and wrote the paper.

Preprints

1. **Tung-Anh Nguyen**, Tuan Dung Nguyen, Long Tan Le, Canh T Dinh, Nguyen H. Tran: "On the Generalization of Wasserstein Robust Federated Learning," arXiv preprint arXiv:2206.01432, 2022 [3]. I co-designed the research, co-conducted the research, did experiments, and wrote the paper.

2. **Tung-Anh Nguyen**, Quan Van La, Phuong Tran, Long Tan Le, Wei Bao, Choong Seon Hong, Nguyen H. Tran: "Federated Koopman Learning for Wireless Traffic Prediction," under review at *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications* at the time of writing this thesis. This research is reported in Chapter 4.2. I designed the research, conducted the research, did experiments, and wrote the paper.

3. **Tung-Anh Nguyen**, Long Tan Le, Quan La Van, Phuong Tran Nguyen Hoang, Robert Kohn, Ngoc Tran Minh, Wei Bao and Nguyen H. Tran: "Spatio-Temporal

Federated Koopman Learning for Non-Stationary Multivariate Time Series Prediction, " under review at *SIAM International Conference on Data Mining (SDM25)* at the time of writing this thesis. This research is reported in Chapter 4. I designed the research, conducted the research, did experiments, and wrote the paper.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

December 15, 2024

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

December 15, 2024

Contents

Statement of Originality ii

Acknowledgements iii

Abstract iv

Relevant Publications vi

List of Figures x

List of Tables xii

Introduction 1

1 Introduction 1
- 1.1 Background and Motivation . . . 1
  - 1.1.1 Federated Learning for the Internet of Things . . . 1
  - 1.1.2 A Typical Federated Training Process . . . 2
  - 1.1.3 Advantages and Challenges of FL in IoT . . . 3
    - 1.1.3.1 Non-IID Problem . . . 7
    - 1.1.3.2 Communication Challenges . . . 8
    - 1.1.3.3 Intrusion Detection Challenges in IoT-FL Systems . 9
    - 1.1.3.4 Learning Challenges from the Dynamic Nature of IoT Data . . . 10
  - 1.1.4 Leveraging Low-rank Learning for FL in IoT Networks . . . 11
- 1.2 Thesis Organisation and Contributions . . . 12

2 Literature Review 15
- 2.1 Federated Learning . . . 15
- 2.2 Federated Learning for IoT Anomaly Detection . . . 16
- 2.3 Federated Learning for Non-stationary Time Series Modeling . . . 18
- 2.4 Manifold Hypothesis . . . 19
- 2.5 Learning on Grassmann Manifolds . . . 19
- 2.6 Distributed Optimization via the Alternating Direction Method of Multipliers . . . 22

3 Federated PCA on Grassmann Manifolds for Anomaly Detection in IoT Networks **24**
3.1 Introduction . . . 24
3.2 System Model . . . 26
3.2.1 Principal Component Analysis Revisiting . . . 26
3.2.2 Federated PCA via ADMM Consensus Optimization . . . 28
3.2.3 FedPCA using ADMM on Grassmann Manifolds . . . 30
3.2.4 Computational Complexity . . . 33
3.3 FedPCA: Convergence Analysis . . . 34
3.3.1 Bound on the Successive Difference of Dual Variables . . . 35
3.3.2 Bound for the Augmented Lagrangian . . . 36
3.3.3 Convergence of the augmented Lagrangian . . . 38
3.3.4 Convergence to stationary point . . . 38
3.3.5 Convergence rate of the augmented Lagrangian . . . 39
3.3.6 Convergence rates of the proposed methods . . . 41
3.4 Experimental Results . . . 41
3.4.1 Experimental settings . . . 41
3.4.1.1 Datasets and Metrics . . . 42
3.4.1.2 Baselines . . . 43
3.4.1.3 Performance Metrics . . . 43
3.4.1.4 Training Details . . . 44
3.4.2 Experimental Results . . . 44
3.4.2.1 Efficiency of Federated PCA on Grassmann Manifold 44
3.4.2.2 Effects of number clients on performance . . . 46
3.4.2.3 Performance of Network Anomaly Detection Tasks . 47
3.4.2.4 Communication Efficiency . . . 50
3.4.2.5 Memory and Time Complexity . . . 51

4 Communication-Efficient Federated Koopman Learning for Non-stationary IoT Data on Grassmann Manifolds **53**
4.1 FedKoop: Communication-Efficient Federated Koopman Learning for Non-stationary IoT Time Series on Grassmann Manifolds . . . 53
4.1.1 Introduction . . . 53
4.1.2 Multivariate Time Series as Dynamics . . . 56
4.1.2.1 Koopman Theory . . . 56
4.1.2.2 Optimal Mode Decomposition . . . 56
4.1.2.3 Multivariate Time Series as Dynamics . . . 57
4.1.3 Federated Koopman Methodology . . . 57
4.1.3.1 Problem Statement . . . 57
4.1.3.2 ADMM for Federated Koopman Optimization . . . 57
4.1.3.3 Federated Koopman on Grassmann Manifold . . . 60
4.1.3.4 Stationary Component Separation . . . 61
4.1.3.5 Long-Term Prediction with FedKoop . . . 62
4.1.4 Experimental Results . . . 64
4.1.4.1 Experimental Setup . . . 64
4.1.4.2 Experimental Results . . . 65

4.2 FedKooL: Communication-Efficient Federated Koopman Learning for Wireless Traffic Prediction on Grassmann Manifolds 69
4.2.1 Introduction 69
4.2.2 Wireless Traffic Time Series as Dynamics 71
4.2.3 Federated Koopman Methodology 72
4.2.3.1 System Model 72
4.2.3.2 Centralized Koopman Learning 73
4.2.3.3 Problem Formulation 73
4.2.3.4 Federated Koopman Learning 74
4.2.3.5 FedKooL: Computational Complexity Analysis 75
4.2.3.6 Stationary Component Separation 76
4.2.3.7 Long-term Prediction with FedKooL 78
4.2.4 Experimental Results 78
4.2.4.1 Experimental Settings 79
4.2.4.2 Experimental Results 81

5 Conclusion and Future Work **85**
5.1 Summary 85
5.2 Future Work 86

List of Figures

1.1 Illustration of a Federated Learning system for IoT. IoT devices are connected to a central server via the Internet. User data is generated locally over time and remains on the device, ensuring privacy. Each device trains a local model using its own data and periodically sends model updates to the server, where they are aggregated into a global model. 2

2.1 Illustration of a geodesic update on the Grassmann manifold. Given a point $A \in G(n,k)$ and a tangent direction $\Delta \in \mathcal{T}_A G$, the exponential map $\exp_A(\Delta)$ yields a new point $B \in G(n,k)$. The manifold contains all matrices $A \in \mathbb{R}^{n\times k}$ such that $A^\top A = I_k$. 20

3.1 Overview of a host-based IoT anomaly detection system utilizing Federated PCA. 27
3.2 Illustration of movement on a Grassmann manifold, represented by a sphere. Given a point U_i^k on Grassmann manifold G and a vector Δ on the tangent space denoted by $\mathcal{T}_{U_i}G$ at U_i^k, a point U_i^{k+1} is identified by exponential mapping $\exp_{U_i^k}(\Delta)$. In our context, the manifold G contains all matrix U_i satisfy the condition $U_i^\top U_i = I$. 31
3.3 Class distribution of test sets for UNSW-NB15 and TON-IoT 42
3.4 (a,b) Effects of ρ on the performance of `FedPG` and (c,d) Training loss over global iterations 46
3.5 Effects of rank-k on the performance of `FedPE` and `FedPG` 47
3.6 Test accuracy and training time over global iterations of `FedPE` and `FedPG` on UNSW-NB15 (a,b) and TON-IoT (c,d) 48
3.7 Effects of number of clients on training time and performance of `FedPE` and `FedPG` 49
3.8 (a,b) Receiver Operating Characteristic (ROC) curve and (c,d) Precision-Recall curve 50
3.9 Visualization of UNSW-NB15 traffic after reconstruction using different methods 51
3.10 The size of model exchanged in each communication round. 52
3.11 The memory consumption and inference time of all methods. 52

4.1 Koopman theory removes non-stationary components from non-stationary MTS for electricity dataset [4] in FL, producing stationary data with a zero-centered mean and revealing stationary patterns. The non-stationarity arises from distributional shifts, changing trends, and dominant seasonality, which affect the mean and variance of the time series. 55
4.2 Koopman transformation of a nonlinear dynamics $x_{t+1} = \varphi(x_t)$ to a linear dynamics $\xi(x_{t+1}) = K_\varphi \xi(x_t)$, where ξ with its inverse ξ^{-1} and the Koopman operator K_φ are approximated using finite-rank matrices P^T, P, and Q, respectively. 56
4.3 Federated Koopman learning for non-stationary MTS prediction . . . 58
4.4 FedKoop for non-stationary MTS prediction 62
4.5 Large Scale Prediction Performance 67
4.6 Convergence Analysis 68
4.7 Effect of γ and r on `FedKoop` Performance 68
4.8 Koopman theory is applied to the Telecom Italia dataset [5] to separate non-stationary components from stationary dynamics by projecting the time series into Koopman-invariant subspaces. This yields zero-mean stationary data, aligns patterns to a common range, and uncovers spatial correlations between traffic series. Stationarity is confirmed using the Augmented Dickey-Fuller test with a p-value of 0.005 [6]. . 71
4.9 System model. 72
4.10 Non-stationary components removal with Koopman 76
4.11 Trend and seasonalities deconstruction with Koopman 77
4.12 Seasonality in Milano and Trentino datasets. Trend can be verified due to large values of ACF for small lags and followed by a slow decrease, while the "scalloped" shape is due to the seasonality. Traffic data from a GW covering 100 base stations shows non-stationarity, with trends indicated by significant ACF values and seasonality reflected by "scalloped" shapes. 80
4.13 Impact of missing data on forecasting performance 82
4.14 Convergence rate comparison: FedKooL vs FedDA 83
4.15 Impact of gateway coverage on forecasting performance 84

List of Tables

1.1 Fundamental differences between Federated Learning and data-center distributed learning. . . . 4
1.2 Key challenges associated with FL. . . . 6

3.1 UNSW-NB15 Dataset . . . 45
3.2 TON-IoT Dataset . . . 45
3.3 Performance of network anomaly detection tasks on UNSW-NB15 and TON-IoT datasets. . . . 49

4.1 Dataset Statistics . . . 64
4.2 Baseline Details . . . 65
4.3 MTS forecasting results with $H \in \{24, 48, 96, 168, 336\}$. . . . 66
4.4 Training Time Record for 48-hour Forecast (s/global-iter) . . . 67
4.5 Synthetic data . . . 78
4.6 Dataset Statistics . . . 79
4.7 Italian Wireless Traffic Prediction Performance . . . 81
4.8 `FedKooL` testing on Never-before-seen-data . . . 81
4.9 Wall clock training time with 100 global iterations . . . 83
4.10 Average wall clock training/inference time (seconds) across clients per global iterations . . . 84

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Federated Learning for the Internet of Things

With the rapid proliferation of mobile and Internet-of-Things (IoT) devices, the volume of data generated at the network edge has grown exponentially. By 2025, predictions suggest the number of connected devices will exceed 80 billion, generating 180 trillion gigabytes of data annually [7]. Traditionally, these devices collect diverse data ranging from text and video to geolocation information which is aggregated and processed in centralized, cloud-based data centers to derive insights or train machine learning models [8]. However, this cloud-centric approach is increasingly unsustainable due to key challenges.

First, privacy concerns, legal restrictions, and the sensitive nature of collected data pose significant barriers. IoT data often contains personal information, such as health records or location history, making centralized storage risky. Unauthorized access or data breaches can have severe consequences. For instance, Strava, a fitness tracking app, collects real-time location data to map user routes and foster social engagement through leaderboards and challenges. However, Strava's Global Heatmap incident exposed the locations of military bases and other sensitive areas, revealing the risks of inadequate data management [9]. This incident underscores the need for stringent privacy measures to protect user data. Governments worldwide are also responding to these concerns by enacting stricter privacy laws. The California Consumer Privacy Act (CCPA) [10], implemented in 2020, grants residents the right to access data collected about them and restricts its sale. Similarly, Brazil's General Data Protection Law (LGPD) [11] enhances transparency and control over personal data use. These regulations reflect a growing trend toward empowering consumers and limiting the collection and storage of non-essential data.

Second, the communication cost of transferring massive amounts of data from edge devices to a central server poses a significant challenge. For example, industrial IoT systems in smart manufacturing rely on continuous data streams from sensors for real-time monitoring and optimization. This process not only incurs high operational costs but also depends on stable network connectivity. Additionally, latency issues can impede time-sensitive tasks, such as early fault detection. The strain on core networks becomes even more severe when handling large volumes of unstructured data.

Cloud-based training makes these challenges worse, as wireless networks cause delays and increase communication overhead, reducing overall performance [12, 13].

Since proposed in 2016, FedAvg [14], Federated Learning (FL) has emerged as a promising solution for distributed machine learning, facilitating the training and updating of models across thousands of decentralized devices while minimizing the risk of data leakage [15]. Unlike traditional centralized approaches that necessitate transferring raw data to a central server, FL allows data to remain on local devices, as depicted in Fig. 1.1. This design not only alleviates privacy concerns but also enhances data security. Within an FL framework, a global model is initially shared with client devices, where it is locally trained using the data stored on these devices. The locally trained models are subsequently sent back to a central server, where they are aggregated—typically by averaging the model weights [14]—to produce an updated global model. This iterative process continues until convergence. The communication-efficient and privacy-preserving characteristics of FL make it particularly well-suited for IoT networks, where devices generate vast amounts of sensitive data while operating under constrained communication bandwidth.

1.1.2 A Typical Federated Training Process

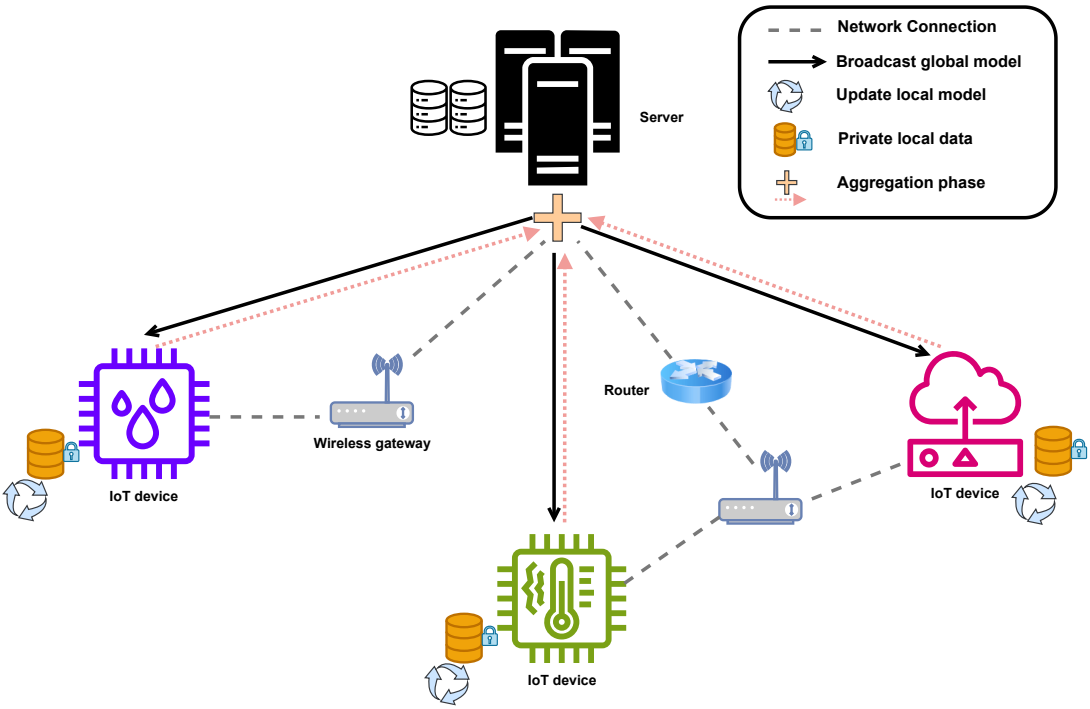

Figure 1.1: Illustration of a Federated Learning system for IoT. IoT devices are connected to a central server via the Internet. User data is generated locally over time and remains on the device, ensuring privacy. Each device trains a local model using its own data and periodically sends model updates to the server, where they are aggregated into a global model.

Since the introduction of the widely adopted `FedAvg` algorithm [14], numerous FL variants have been developed to enhance the performance of IoT systems across

diverse applications, including smart homes [16], healthcare monitoring [17], and autonomous vehicles [18]. These advancements seek to address challenges such as network heterogeneity [19], device reliability [20], and resource limitations [21], thereby enabling FL to facilitate intelligent decision-making in distributed IoT environments.

The conventional setting of FL in Fig.1.1 involves a federation of distributed users, such as mobile phones, user equipment (UEs), and IoT devices, each possessing a private local dataset, along with a central server that maintains a shared global model. The training process in FL generally follows these steps until convergence:

1. *Broadcasting and User Selection*: The server samples eligible users, recognizing that UEs or mobile devices may not always be online due to battery-saving idle modes. Once connected, the selected users download the current global model weights.

2. *Local Training*: Selected users perform local updates on the model, utilizing various optimization techniques. For instance, in the case of `FedAvg`, users may employ Stochastic Gradient Descent (SGD) on their local datasets [15].

3. *Aggregation*: The server collects and aggregates the updated models from participating users into the global model. Techniques such as secure aggregation, differential privacy, and compression may be employed to enhance communication efficiency. In `FedAvg`, aggregation typically involves averaging the local model updates.

4. *Model Update*: The server updates the shared model based on the aggregated results from the current round of user participation.

The whole process of FL includes user computation, model aggregation, and updates, allowing users considerable freedom in how they update their local models and transmit these models to the server. This flexibility facilitates the development of various algorithms and supports multiple research directions. Key areas of focus include improvements in compression techniques, communication efficiency, robustness, differential privacy, and secure multi-party computation. The following section will explore the ongoing challenges in FL, highlighting various existing research directions.

1.1.3 Advantages and Challenges of FL in IoT

Advances in technology have equipped modern edge devices, such as smartphones, IoT nodes, and wearables, with high-performance processors, including CPUs, GPUs, and dedicated AI hardware like neural processing units (NPUs) (e.g., Snapdragon 8 Gen 2, Apple M2, or Google Tensor G3). These devices enable local model training, significantly reducing the need to transmit raw data to centralized servers, thereby improving privacy and lowering network congestion [15]. In addition, the enhanced storage capacity and diverse sensors (e.g., LiDAR, accelerometers, microphones, and GPS) embedded in these devices allow for continuous data collection at the edge. This enables large-scale, distributed model training without compromising user privacy. With improvements in edge computing frameworks, such as TensorFlow Lite and PyTorch Mobile, FL can now be deployed seamlessly in real-world scenarios. For example, smartphones can collect usage data throughout the day and update the global

Table 1.1: Fundamental differences between Federated Learning and data-center distributed learning.

	Data Center Distributed Learning	**Federated Learning**
Architecture	Utilizes robust, centralized computing clusters that can tackle extensive and complex training tasks with high throughput. Nodes are consistently online and ready for data processing.	Comprises a decentralized network of diverse devices, such as smartphones, wearables, and IoT nodes, handling smaller localized tasks. User participation varies, with only a subset active during training sessions.
Scalability	Typically supports 1 to a few thousand nodes, enabling large-scale batch processing of data.	Scales massively, with potential to include billions of devices (10^{10} or more), allowing for extensive user engagement.
Data Management	Data is stored centrally, allowing for manipulation and shuffling, which can lead to homogenized training sets. All nodes can access the entire dataset.	Data remains on individual devices, enhancing privacy by preventing cross-device access. Users' data is non-IID, reflecting individual preferences and behaviors, leading to diverse training experiences.
Server Functionality	The centralized server manages all data, redistributing it as needed for model training and optimization.	The server aggregates only model updates from devices, ensuring that sensitive user data is never transmitted or stored centrally, enhancing privacy and security.
Main Limitation	Computational resources are often the main limitation due to heavy reliance on high-speed data transfers and processing capabilities.	Communication overhead frequently poses challenges, particularly with variable network conditions; often relying on Wi-Fi or cellular connections that can impact efficiency.

model during off-peak hours, such as overnight, improving efficiency and ensuring models remain up-to-date. State-of-the-art models align well with FL frameworks. Vision transformers (ViTs) [22] and lightweight CNN architectures, such as MobileNetV3 [23], are suitable for image classification on mobile devices. Similarly, large language models like BERT [24] or federated adaptations of GPT-4o [25] can handle personalized text prediction tasks. This versatility makes FL a promising solution for various applications, including personalized healthcare monitoring, predictive text input, and privacy-preserving recommendations, ensuring models are both scalable and privacy-centric.

Advantages: FL offers several significant benefits for IoT applications:

- *Privacy*: FL ensures user privacy by transmitting only model parameters rather than raw data to central servers, thus enhancing data security and encouraging more users to engage in collaborative learning.
- *Low Latency*: Local model training allows for real-time decision-making at edge devices, significantly reducing latency, which is vital for applications such as autonomous vehicles.
- *Hardware Efficiency*: The simpler hardware requirements of edge devices compared to central servers facilitate easier deployment and operation.
- *Efficient Bandwidth Usage*: By minimizing the data sent to the cloud, FL reduces network congestion, as only updated model parameters are communicated.

Applications: The advantages of FL lead to transformative real-world applications:

- *Mobile Applications*: FL enhances user experiences in applications like Google's Gboard and Apple's Siri by allowing personalized learning while maintaining privacy, facilitating advancements in smart assistants and health monitoring apps.
- *Healthcare*: FL enables the development of predictive models across decentralized data sources, allowing healthcare institutions to collaboratively diagnose diseases while ensuring patient confidentiality, thus driving innovations in telemedicine and epidemic predictions.
- *Autonomous Vehicles*: FL is crucial for connected autonomous vehicles (CAVs), allowing them to learn from shared experiences in real time and improve safety and navigation without compromising user data.
- *Manufacturing and Predictive Maintenance*: FL aids in building predictive maintenance models that enhance operational efficiency across distributed manufacturing sites while adhering to data privacy regulations, also supporting energy optimization in smart grids.

These applications demonstrate that FL is revolutionizing IoT by enabling privacy-conscious, efficient, and collaborative technologies across various domains.

Challenges: Despite the considerable potential of FL, several critical challenges must be navigated within distributed frameworks:

Table 1.2: Key challenges associated with FL.

System Diversity	FL operates across a wide range of devices, each varying in computational power (CPU, GPU), battery life, and network capabilities. This diversity complicates the training process, as the system must adapt to varying performance levels and conditions.
Non-IID Data	User-generated data often stems from different sources and distributions, leading to a lack of independence and identical distribution (non-IID) among datasets. This variability can hinder the model's ability to generalize effectively across all users.
Communication Overhead	Frequent communication rounds are necessary to synchronize model updates between users and the central server. The reliance on wireless networks can lead to increased latency and bottlenecks, especially in environments with limited connectivity.
Security Vulnerabilities	Limited data visibility and device heterogeneity make it difficult to distinguish intrusions from normal anomalies. Constrained resources and distributed model updates further increase vulnerability to evolving threats and adversarial manipulation.
Privacy Risks	While FL aims to enhance user privacy by transmitting only model updates (e.g., gradients), these updates can inadvertently leak sensitive information. This risk necessitates robust mechanisms to ensure data confidentiality against adversaries.

- *System Heterogeneity*: The diverse ecosystem of user devices in FL presents significant challenges due to varying hardware specifications (such as CPU, GPU, and battery capacity) and unique data distributions. This heterogeneity complicates the implementation of a scalable and unified global model, as the inherent statistical diversity can obstruct effective aggregation.

- *Non-IID Data*: Data generated by users often originates from distinct distributions, leading to non-independent and identically distributed (non-IID) scenarios. In conventional FL paradigms, a singular global model is developed, which may inadequately represent local data characteristics, resulting in user drift and suboptimal model performance on localized datasets.

- *Communication Overhead*: The dependence on regular communication between user devices and the central server creates a substantial overhead. Elevated communication costs can introduce bottlenecks, particularly in wireless environments, potentially impeding the training process and diminishing overall system

throughput.

- *Security Vulnerabilities*: in FL for IoT intrusion and anomaly detection stem from limited data visibility and heterogeneous devices, which make distinguishing malicious behaviors from normal anomalies difficult. Resource constraints and distributed updates further expose the system to evolving threats and potential adversarial manipulation.

- *Privacy Implications*: While FL endeavors to bolster user privacy by exchanging model updates rather than raw data, there are still inherent risks associated with the transmission of these updates. Sensitive information may inadvertently be disclosed to external entities or the central server, necessitating robust security measures to safeguard user confidentiality.

Key obstacles to the practical implementation of FL, as indicated in Table 1.2, include communication bottlenecks and vulnerabilities to adversarial attacks, alongside privacy concerns. Effectively addressing these issues requires the development of robust algorithmic solutions that ensure convergence while overcoming these limitations. Thus, the challenge remains to develop algorithms that improve convergence rates and mitigate communication limitations. The following section explores representative scenarios highlighting these challenges in FL, including non-IID data, communication constraints, adversarial threats, and the modeling dynamic nature of IoT data .

1.1.3.1 Non-IID Problem

FL encounters significant challenges due to the non-IID problem [26], particularly in the context of the IoT. In IoT systems, data generated by various user equipments is influenced by individual behaviors, environmental factors, and temporal variations, leading to heterogeneous datasets across devices [27]. Each IoT device i maintains a local dataset represented as $\{\mathbf{x}_j, y_j\}_{j=1}^{D_i}$, where $\mathbf{x}_j \in \mathbb{R}^d$ and D_i is the size of the dataset. During training, an aggregation server samples device i from an overall distribution $\mathcal{D}$. A critical feature of non-IID data in this context is the divergence in data distributions among devices, expressed as $\mathcal{D}_i \neq \mathcal{D}_m$ for any two devices m and i. The non-IID issue manifests in various forms:

- *Feature Distribution Skew*: Devices may generate data with similar labels but different feature distributions. For example, smart home devices may monitor temperature under different environmental conditions, leading to variations in how the same temperature readings are represented.

- *Label Distribution Skew*: Although the conditional feature distribution $\mathcal{D}(\mathbf{x}|y)$ might be stable, the label distributions $\mathcal{D}_i(y)$ can vary widely. In healthcare IoT applications, different devices might primarily track distinct health parameters, resulting in uneven distributions of health-related labels among users.

- *Heterogeneity and Imbalance*: The size of datasets across devices often varies significantly, with some devices contributing minimal or no data initially, complicating the aggregation process. For instance, in a smart city context, some sensors may report frequent data, while others may only report intermittently, creating imbalances in data volume.

The implications of non-IID data in IoT are profound, as they can result in suboptimal model performance due to inadequate representation of specific classes or features in the global model. This occurs because the global model aggregates data that may insufficiently represent certain classes or features, which is especially challenging when devices generate highly skewed data distributions or rare but important events [28, 29]. As a result, the model's ability to generalize across diverse clients is compromised, emphasizing the necessity for approaches that specifically address heterogeneity in data. Additionally, traditional federated learning algorithms often suffer from slower convergence or even divergence when handling non-IID data. The inconsistent gradient directions arising from heterogeneous client data impede the global model's progress toward optimal solutions [29, 30]. This problem necessitates algorithmic adaptations designed to stabilize training dynamics despite data disparities.

To tackle these challenges, several advanced solutions have been proposed. Adaptive sampling and client selection mechanisms focus on choosing clients with more representative or diverse data to reduce bias and enhance convergence [31, 32]. Personalized model architectures allow individual clients to retain specific local parameters alongside a shared global backbone, effectively balancing local specialization and global knowledge sharing to improve accuracy under heterogeneous conditions [33, 34]. Moreover, robust aggregation techniques have been developed to replace simple averaging with methods resistant to outliers and the adverse effects of non-IID data, including median-based, trimmed mean, and Byzantine fault-tolerant strategies [35, 36].

Despite these promising advancements, several challenges persist. There remains a delicate balance between personalization and generalization, as excessive personalization can undermine the benefits of collaborative learning [26]. The increased computational and communication costs introduced by sophisticated methods also pose practical constraints [37]. Furthermore, privacy concerns emerge when adaptive or personalized approaches require additional metadata exchange [29]. Many existing studies focus primarily on benchmark datasets, leaving open questions about scalability and robustness in real-world IoT deployments.

1.1.3.2 Communication Challenges

As detailed in Section 1.1.2, the FL framework facilitates the distribution of a global model to a vast number of user devices within the IoT ecosystem via wireless networks. This user base can number in the millions, often contending with limited and unstable bandwidth [38]. Consequently, implementing FL in IoT environments poses significant challenges. The training process requires multiple communication rounds to exchange global and local models between the central server and user devices, with potential delays stemming from bandwidth restrictions and energy limitations on user equipment [39]. Although communication costs for sharing model updates are typically lower than those for transmitting extensive user data to a central server, optimizing bandwidth and minimizing communication rounds are essential for improving efficiency [40]. Key challenges contributing to communication bottlenecks within the FL network for IoT are outlied as follows:

- *Large User Base:* A key advantage of FL is its ability to train machine learning models on extensive datasets contributed by numerous IoT devices. However, the simultaneous involvement of many devices can create communication bottlenecks,

escalating overall computational costs [39, 41].

- *Network Bandwidth Limitations:* FL is inherently dependent on reliable communication for the exchange of model updates between distributed devices and the central server. However, in bandwidth-constrained environments, particularly within IoT networks, unstable or low-speed connections introduce significant communication bottlenecks [41]. These limitations delay the transmission of model parameters, impede timely synchronization, and prolong the training process, thereby adversely affecting the convergence rate and overall performance of the FL system.

- *User Computation Constraints:* In FL, computational tasks are offloaded to IoT devices, which may have varying processing capabilities. Limitations in computational power, storage, and bandwidth can adversely affect performance. For example, training an image classification model with millions of parameters on a powerful cloud GPU can achieve high throughput, while IoT devices may require significantly more time due to lower processing speeds [42].

- *Statistical Heterogeneity:* Non-IID data across IoT devices can also lead to communication bottlenecks, as discrepancies in data sizes affect model update efficiency [43]. Devices with larger datasets may experience longer update times, delaying the aggregation of the global model.

- *System Heterogeneity:* The diversity in wireless channel conditions and hardware capabilities among a multitude of IoT devices complicates both model training and communication processes. This variation results in increased communication costs during model aggregation.

The discussed limitations underscore the challenges of communication bottlenecks in FL networks within IoT environments. Evaluating the complexity of FL algorithms extends beyond merely counting update rounds, as wall-clock duration during deployment serves as a crucial metric. This duration is determined by the interplay of multiple factors, including the number of participating devices and their heterogeneous characteristics, such as computational capabilities and energy constraints. The overall training time is influenced by user-side computation, local dataset sizes, and collective communication times, which are further impacted by channel conditions and data transmission volumes. To optimize wall-clock training time, resource allocation strategies must account for both algorithmic parameters and user-specific constraints, particularly under dynamic and unreliable wireless conditions.

1.1.3.3 Intrusion Detection Challenges in IoT-FL Systems

Intrusion detection in federated IoT systems presents substantial challenges due to the decentralized and large-scale nature of IoT systems. In FL-enabled IoT networks, millions of edge devices collaboratively train models by sharing local updates over bandwidth-constrained and potentially unstable wireless connections [39]. This distributed setup, while preserving data privacy, limits the central server's visibility into local data and computations, making it difficult to detect irregular or malicious behaviors. Adversarial devices can exploit this limited oversight to introduce abnormal

patterns, disrupt convergence, or manipulate the global model without immediate detection [44].

A significant challenge arises from the scale of participation. Even a small number of compromised or malfunctioning devices can substantially degrade the performance of the global model, particularly when their updates are disproportionately influential or when they hold critical segments of the data distribution. The situation is further complicated by the inherent heterogeneity in IoT systems. Non-IID data distributions and variations in sensing environments can cause legitimate client updates to differ significantly, making it difficult to distinguish between benign anomalies and truly malicious behavior [45].

System-level constraints also hinder effective intrusion detection. Many IoT devices operate under strict limitations in terms of computation, memory, and energy resources. These constraints restrict the use of advanced defense mechanisms locally, shifting the burden of detection and mitigation to the central server. However, centralized strategies must function under limited observability and still maintain communication and computational efficiency [46]. Furthermore, existing detection methods often assume certain statistical properties of client updates that may not hold in real-world, dynamic IoT environments.

Together, these factors create a highly complex landscape for intrusion detection in federated IoT networks. Addressing this requires the design of FL algorithms and defense mechanisms that can tolerate statistical variability, operate efficiently under resource constraints, and reliably identify and suppress malicious behavior without sacrificing performance or scalability.

1.1.3.4 Learning Challenges from the Dynamic Nature of IoT Data

The dynamic nature of data in IoT systems presents profound challenges for FL. In contrast to conventional machine learning pipelines, which typically rely on curated, stationary datasets, IoT devices continuously generate streaming data under ever-changing operational, environmental, and contextual conditions. These non-stationary and heterogeneous characteristics violate many of the statistical assumptions underlying standard FL algorithms and significantly complicate model convergence, robustness, and long-term generalization [47, 48].

One of the primary challenges stems from the non-stationarity of data streams. IoT devices often operate in environments where the data distribution drifts over time—a phenomenon known as concept drift. For instance, human activity patterns captured by wearables may change with seasons or daily routines, while sensors in smart agriculture systems respond to weather or soil changes. Without continuous model adaptation, FL systems become susceptible to performance degradation, as static models fail to keep pace with evolving input distributions [49]. Recent advances in online and continual learning within FL frameworks have attempted to mitigate this issue, yet most solutions remain limited by communication costs and instability under asynchronous updates [50].

In addition to temporal drift, spatio-temporal dependencies are a distinguishing feature of many IoT applications. Sensor data from geographically proximate or functionally related devices such as traffic flow sensors on adjacent road segments or synchronized traffic lights often exhibit structured correlations in congestion patterns

and environmental conditions. Ignoring these relationships can lead to models that miss crucial contextual cues necessary for accurate prediction or anomaly detection. While centralized learning settings have begun to address this challenge using techniques such as graph neural networks (GNNs) and temporal attention mechanisms, their application within FL systems remains limited. These approaches are often computationally intensive and introduce additional communication overhead, making them difficult to scale across resource-constrained edge devices [51, 52].

Event-driven variability introduces additional volatility. Many IoT deployments remain dormant for extended periods until triggered by specific events, such as medical emergencies in health monitoring or security breaches in surveillance systems. These events often produce data bursts with drastically different distributions, requiring the global model to adapt quickly. However, current FL aggregation strategies typically use uniform sampling or static client participation rates, failing to exploit contextual signals that could prioritize critical updates [53]. Designing event-aware client selection or adaptive aggregation remains an open and underexplored research frontier.

The heterogeneity of data sources also poses a persistent problem. Devices differ in hardware capabilities, sampling frequencies, communication patterns, and sensing modalities, resulting in vastly diverse local datasets. These disparities hinder convergence and undermine fairness across clients. Personalization techniques such as meta-learning, multi-task FL, or clustered aggregation have been proposed to tailor global models to individual clients, but achieving scalability and fairness without sacrificing privacy remains a delicate trade-off [54, 55].

Finally, IoT data is often governed by long-tail distributions, where common events dominate training data while rare, safety-critical events (e.g., faults, intrusions) appear infrequently. Standard FL training schemes risk overfitting to frequent patterns and underperforming in rare but vital scenarios. Memory-based replay buffers, reweighting schemes, and selective update filtering have shown promise, but these methods are typically designed for centralized architectures and often require assumptions incompatible with real-world IoT deployments [56].

Taken together, these challenges underscore the need for FL algorithms that are not only privacy-preserving and communication-efficient, but also adaptable, context-aware, and capable of long-term learning in the presence of non-stationarity and diversity. While existing techniques offer partial solutions, several directions remain underexplored. There is growing interest in understanding how federated models can distinguish between benign distributional shifts and malicious behavior, particularly in scenarios where changes occur abruptly and without warning. At the same time, developing architectures that can model temporal and spatial dependencies efficiently, without imposing significant resource overhead on constrained edge devices, remains a pressing concern. Ultimately, advancing toward a truly lifelong and resilient FL paradigm for IoT applications will require addressing these multifaceted challenges in a principled and scalable manner.

1.1.4 Leveraging Low-rank Learning for FL in IoT Networks

The manifold hypothesis, which asserts that high-dimensional data generated from natural processes typically resides on or near a lower-dimensional manifold embedded within the ambient space, underpins many advances in representation learning [57, 58].

In the context of FL, this assumption has profound implications for both communication efficiency and system robustness. By exploiting the intrinsic geometry of data manifolds, FL systems can encode high-dimensional model updates or inputs into compact latent representations, thereby substantially reducing communication overhead—a critical requirement for IoT environments characterized by limited bandwidth, energy, and computational capacity [59–61].

Beyond compression, the manifold structure provides a geometric framework for the detection and mitigation of irregularities such as anomalies, corrupted updates, and adversarial manipulations [62, 63]. Data points or gradient updates that deviate significantly from the learned manifold can be effectively identified through techniques such as local intrinsic dimensionality estimation, manifold projection, or regularization methods [64, 65]. These capabilities enable FL systems to isolate and suppress unreliable or malicious contributions from distributed clients, enhancing overall robustness and reliability. Thus, leveraging the manifold hypothesis not only facilitates efficient distributed optimization but also strengthens the resilience of FL against adversarial and non-standard data behaviors [66].

Building on this foundation, Grassmann manifolds [67] offer a mathematical framework for handling low-rank representations of data. Projections onto Grassmann manifolds allow FL systems to preserve the intrinsic structure of model updates while optimizing their transmission [68]. This technique is particularly advantageous in IoT environments, where maintaining the integrity of transmitted updates is critical despite the constraints of limited communication bandwidth and computational power. Additionally, Grassmannian optimization provides a robust mechanism for ensuring that updates align with the expected low-dimensional manifold, effectively filtering out noisy or adversarial data inputs [69]. This not only improves the system's communication efficiency but also enhances its robustness against malicious attacks and environmental noise.

While the manifold hypothesis and Grassmann manifolds address the spatial structure of data, IoT systems are inherently dynamic, with data patterns evolving over time. Koopman theory introduces a powerful tool for capturing these temporal dynamics by offering a linear framework to represent the evolution of nonlinear systems [70]. Through Koopman operators, FL systems can model and predict the temporal behavior of IoT data by lifting it into a high-dimensional space where linear transformations describe complex dynamics. This complements the spatial insights provided by the manifold hypothesis, as the invariant subspaces identified by Koopman theory often correspond to the intrinsic low-dimensional structures of the data [71]. By integrating temporal modeling with spatial compression, Koopman theory enhances the adaptability of FL systems to changing network conditions and evolving data distributions.

1.2 Thesis Organisation and Contributions

The overarching goal of this thesis is to enable robust and communication-efficient FL in dynamic, non-IID IoT environments. Achieving this requires addressing two fundamentally distinct yet interconnected challenges, each demanding specialized focus due to their unique characteristics and complexities. The first challenge pertains to intrusion detection—developing effective methods to identify and mitigate adversarial

or anomalous behavior that can compromise the integrity of the federated model. This aspect is critical because intrusion threats manifest in diverse and evolving forms, requiring tailored geometric and statistical tools to ensure system robustness without excessive communication overhead.

The second challenge involves modeling the intrinsic temporal dynamics and statistical heterogeneity of IoT data under non-IID conditions. Accurately capturing these dynamics is essential for adapting the global model to reflect real-world variations, which in turn enhances learning efficiency and predictive performance. Due to the fundamentally different nature of dynamic data modeling compared to intrusion detection, this challenge necessitates leveraging advanced operator-theoretic frameworks and low-dimensional manifold representations to balance expressiveness and resource constraints.

By dividing the research into these two focused sections, the thesis can develop dedicated solutions that address the unique demands of each problem space. The first section concentrates on designing communication-efficient, geometry-driven intrusion detection mechanisms that safeguard model integrity. The second section advances novel methodologies for dynamic, non-IID data modeling using Koopman operator theory and Grassmann manifolds to enable scalable and adaptive FL. Together, these complementary lines of inquiry establish a comprehensive framework for secure, resilient, and efficient federated learning in IoT systems. The remainder of this thesis is organized into five chapters, with the main contributions of each chapter summarized as follows.

Chapter 2. Literature Review. This chapter reviews state-of-the-art research on FL for non-IID and heterogeneous data, methodologies for intrusion detection and anomaly detection in federated IoT systems, and approaches for modeling dynamic IoT time series. Moreover, we review the manifold hypothesis and Grassmann manifolds, which are foundational to the thesis and key to enabling communication efficiency in the proposed methods.

Chapter 3. Federated PCA on Grassmann Manifolds for Anomaly Detection in IoT Networks. In this chapter, we present a framework for detecting intrusions and anomalies in IoT systems to enhance security and mitigate potential threats. We propose a novel federated unsupervised anomaly detection framework – FedPCA – that leverages Principal Component Analysis (PCA) and the Alternating Directions Method Multipliers (ADMM) to learn common representations of distributed non-i.i.d. datasets. Building on the FedPCA framework, we propose two algorithms, `FedPE` in Euclidean space and `FedPG` on Grassmann manifolds. Our approach enables real-time threat detection and mitigation at the device level, enhancing network resilience while ensuring privacy. Moreover, the proposed algorithms are accompanied by theoretical convergence rates even under a sub-sampling scheme, a novel result. Experimental results on the UNSW-NB15 and TON-IoT datasets show that our proposed methods offer performance in anomaly detection comparable to non-linear baselines, while providing significant improvements in communication and memory efficiency, underscoring their potential for securing IoT networks.

Chapter 4. Communication-Efficient Federated Koopman Learning for Non-Stationary IoT Data on Grassmann Manifolds. Building upon the communication-efficient framework developed in Chapter 3, this chapter delves into

FL methodologies tailored for IoT data. The focus lies on addressing the dynamic and non-stationary characteristics of IoT data while ensuring communication efficiency. We introduce **Federated Koopman** (`FedKoop`) learning, a privacy-preserving FL framework designed for non-stationary multivariate time series, leveraging Koopman theory to capture spatio-temporal dependencies and tackle non-stationarity. `FedKoop` formulates a novel optimization problem to learn the Koopman operator over distributed time series data, utilizing gradient movement on Grassmann manifolds to accelerate training. Experimental results confirm that `FedKoop` outperforms baseline methods in non-stationary time series forecasting, achieving reduced computational overhead and memory usage, making it well-suited for large-scale cyber-physical systems.

In the second part of this chapter, we extend the advancements introduced by `FedKoop` to develop **Federated Koopman Learning (`FedKooL`)** for wireless traffic prediction, a critical challenge in managing IoT system traffic for next-generation networks. Building on the Koopman-based approach, `FedKooL` captures inter-spatial dependencies and non-stationarity, further refining the modeling of distributed time series data through low-rank matrices on low-dimensional linear subspaces, significantly reducing computational complexity. To optimize efficiency, `FedKooL` incorporates Rockafellar's envelope for managing low-rank parameters and computes gradient projections on Grassmann manifolds to accelerate the learning process. Empirical evaluations using the Telecom Italia dataset demonstrate that `FedKooL` achieves a 38% improvement in RMSE and a 56% improvement in MAE over five benchmark methods, underscoring its effectiveness in wireless traffic prediction for IoT networks.

Chapter 5. Conclusion and Future Work. This chapter provides a summary of the key contributions of this research and outlines promising avenues for future exploration.

Chapter 2

Literature Review

In Chapter 1, we articulated the primary objective of this thesis: addressing the challenges of intrusion detection, non-stationarity, and communication efficiency within IoT environments. Although significant advancements have been made in these domains, the distinctive properties of IoT systems demand further in-depth investigation. This chapter commences with a comprehensive review of anomaly detection methodologies tailored for federated IoT settings, with a particular focus on strategies that enhance intrusion detection capabilities. Subsequently, we survey state-of-the-art approaches for handling non-stationary time series data, highlighting adaptive techniques designed to accommodate evolving data distributions. Thereafter, we introduce the manifold hypothesis, a foundational principle for characterizing the intrinsic low-dimensional structure underlying high-dimensional data. Building upon this, we discuss learning algorithms formulated on Grassmann manifolds, which are instrumental in improving communication efficiency in federated learning frameworks. Finally, we examine optimization techniques based on the Alternating Direction Method of Multipliers (ADMM), emphasizing their applicability to decentralized optimization problems and their potential to provide robust, scalable solutions by harnessing distributed computational resources in IoT networks.

2.1 Federated Learning

FL has emerged as a pivotal distributed learning paradigm that enables collaboration among thousands to millions of edge devices to jointly train machine learning models while preserving data privacy and reducing communication overhead [72, 73]. Unlike traditional centralized learning, where raw data is collected on a central server, FL keeps user data localized on client devices. Each device downloads a global model, performs training using its private data, and transmits only model updates (such as gradients or weights) to a central server for aggregation. The updated global model is then redistributed to clients for subsequent training rounds [14].

The foundational FedAvg algorithm [14] introduced an effective approach by iteratively aggregating client models through weighted averaging, balancing simplicity and communication efficiency. This approach has seen extensive adoption in IoT settings, where privacy concerns and limited bandwidth are paramount [18, 74]. FL inherently supports privacy preservation by avoiding raw data transfers, while enabling intelligent processing across geographically distributed and resource-constrained IoT

devices [75, 76].

Since FedAvg, numerous variants have been developed to address real-world challenges including statistical heterogeneity (non-IID data), system heterogeneity, communication constraints, and security threats [77]. Personalized FL methods tailor global models to fit diverse local data distributions, improving performance in heterogeneous environments [78, 79]. Communication-efficient strategies such as model compression, quantization, and sparsification have been introduced to alleviate bandwidth bottlenecks [80, 81]. Robust aggregation techniques further enhance system resilience by mitigating the impact of adversarial or faulty client updates [82, 83].

Emerging research explores integrating FL with advanced learning frameworks to better capture complex data characteristics. Federated continual learning addresses evolving data distributions over time, a crucial aspect in dynamic IoT ecosystems [84]. Federated graph neural networks have been proposed to model intricate spatio-temporal dependencies within sensor networks and interconnected devices [85, 86].

Despite these advances, significant open challenges remain. These include effectively handling non-IID and dynamically changing data distributions, achieving scalable and communication-efficient training under constrained device resources, and ensuring strong privacy and security guarantees in practical deployments [74, 87]. These challenges are particularly acute in IoT due to device heterogeneity, dynamic operational environments, and limited computational capabilities.

A pressing concern lies in intrusion detection and the management of evolving, non-stationary data streams. FL frameworks must distinguish between benign anomalies stemming from natural data variability and malicious behaviors, often under limited observability and strict communication budgets. Moreover, adapting FL algorithms to continuously changing data while maintaining computational and communication efficiency is essential. Addressing these intertwined challenges calls for novel methodologies that harmonize robustness, adaptability, efficiency, and privacy.

2.2 Federated Learning for IoT Anomaly Detection

The increasing reliance on IoT systems has brought their inherent vulnerabilities into sharp focus, underscoring the urgent need for robust anomaly detection mechanisms to safeguard these complex and widely distributed networks [88]. Operating in often uncontrolled and heterogeneous environments, IoT devices face a diverse range of cyber threats, including denial-of-service attacks, stealthy intrusions, and data breaches [89]. Consequently, anomaly-based Intrusion Detection Systems (IDS) have emerged as a pivotal defense strategy, leveraging machine learning to identify deviations from expected network or device behaviors [90].

Early research efforts predominantly relied on centralized anomaly detection architectures, wherein raw data from distributed IoT devices is aggregated at a central server for analysis and model training [91–96]. These approaches benefit from access to large, centralized datasets that facilitate training of complex models, such as deep neural networks and ensemble classifiers [91]. For instance, Meidan et al. [91] developed a deep learning-based IDS demonstrating high detection accuracy through centralized data aggregation. However, despite their effectiveness, centralized solutions pose significant challenges in large-scale, privacy-sensitive IoT deployments. The transfer and storage of raw user data at a central location introduce substantial privacy risks and increase

exposure to cyberattacks [94]. Additionally, continuous data transmission exacerbates network congestion, particularly in bandwidth-constrained environments [95], while scalability becomes impractical as IoT-generated data volumes grow exponentially [96].

To overcome these limitations, FL has emerged as a promising decentralized paradigm that enables collaborative model training without aggregating raw data [72, 73]. In FL, IoT devices or edge nodes train models locally on private data and transmit only model updates—such as gradients or parameters—to a coordinating server, which aggregates these to update a global model [14]. This process inherently preserves privacy by retaining raw data on-device and reduces communication overhead by exchanging compressed model updates instead of raw datasets [75]. The foundational FedAvg algorithm [14] introduced weighted averaging of client updates to iteratively refine a global model. Its simplicity and communication efficiency have driven widespread adoption in IoT scenarios where privacy and constrained network resources are critical [18, 74]. FL aligns naturally with the distributed topology of IoT systems, enabling scalable, privacy-aware anomaly detection across heterogeneous devices [76].

Several seminal works have demonstrated FL's viability for IoT anomaly detection. Nguyen et al. [97] proposed a federated intrusion detection system leveraging Recurrent Neural Networks (RNNs) trained locally on edge gateways. Their approach preserved user privacy by confining data to local domains while achieving high detection accuracy through global model aggregation. Similarly, Wang et al. [98] applied feed-forward artificial neural networks in a federated setting, showing improved detection across heterogeneous devices and varied network traffic. Mothukuri et al. [99] extended these findings by employing Gated Recurrent Units (GRUs) within FL to capture temporal dependencies in IoT streams, maintaining strict data locality and privacy.

Despite these advances, several challenges remain. A primary difficulty is the severe class imbalance characteristic of IoT traffic, where normal behavior vastly outweighs anomalous events, complicating supervised FL model training and impairing rare or novel attack detection [100]. Moreover, heterogeneity in device capabilities, sensing modalities, and local data distributions leads to non-IID conditions that hinder model convergence and degrade generalization [77, 78]. Tackling this requires personalized or clustered FL strategies [79, 101], which often increase computational and communication costs.

Additionally, IoT devices typically face constraints in computation, memory, energy, and network connectivity, mandating the design of lightweight, communication-efficient FL algorithms that uphold detection accuracy and privacy [76, 80].

While FL offers a compelling framework for privacy-preserving and communication-efficient anomaly detection in IoT, addressing issues related to data imbalance, heterogeneity, resource limitations, and dynamic network conditions remains an open challenge. The development of adaptive, scalable FL frameworks capable of sustaining robustness, efficiency, and generalization across diverse real-world deployments is essential for advancing anomaly detection in IoT ecosystems.

2.3 Federated Learning for Non-stationary Time Series Modeling

IoT data is inherently dynamic, evolving continuously over time and influenced by complex interdependencies. This ever-changing nature of IoT systems, such as those employed in traffic control, climate forecasting, and energy management, introduces significant challenges for predictive modeling. Among these, the most critical challenge arises from the non-stationarity of IoT data, where statistical properties shift over time due to factors such as environmental changes, seasonal patterns, and long-term trends [102]. These shifts obscure underlying temporal dynamics and hinder the stability and accuracy of predictive models.

To address non-stationarity in time series forecasting, various methods aim to mitigate trends, seasonality, and cyclic patterns. A common approach involves decomposing the time series into non-stationary and stationary components, enhancing model predictions. N-BEATS [103] identifies patterns without feature engineering, while traditional methods like ARIMA effectively capture linear trends. Integrating traditional models (e.g., ARIMA [104]) with neural networks (e.g., LSTM [105]) leverages both statistical and machine learning strengths for complex MVTS analysis. Attention-based mechanisms, inspired by NLP applications, such as the Transformer and its adaptations (e.g., Informer [106], Autoformer [107]), show promising results in forecasting. Additionally, data normalization techniques like RevIN [108], Dish-TS [109] and Non-Stationary Transformer (NST) [110] significantly improve model performance by mitigating non-stationarity's adverse effects. Recent advancements have integrated Koopman theory with deep learning techniques, such as KAE [111] and eDMD [112], to simultaneously learn measurement functions and operators. Very recently, techniques like KNF [113] and Koopa [114] have leveraged Koopman theory to address changing temporal distributions in time series. Despite significant advancements, mentioned approaches relied on centralized settings, neglecting the inherently decentralized nature of IoT data. IoT environments are characterized by distributed data sources with diverse and often non-IID statistical properties, necessitating methods designed specifically for decentralized and heterogeneous data streams. FL frameworks have emerged as a promising solution for these challenges in IoT settings. For instance, Fog-based Federated Time Series Forecasting (FFTF) [115] integrates FL and fog computing to address the non-IID characteristics of IoT data while preserving privacy and reducing communication overhead. Similarly, MOENS-FTS [116] employs embedding transformations and non-stationary fuzzy time series modeling to manage high-dimensional streaming data in IoT applications.

However, these approaches, like their centralized counterparts, often rely on increasing model complexity to achieve better performance. This strategy disregards the resource-constrained nature of many IoT devices, emphasizing the critical need for FL algorithms that optimize computational and communication efficiency. Additionally, the dynamic and evolving nature of IoT data demands FL models capable of adapting to temporal changes in data distributions. This adaptation requires frequent retraining to address concept drift, which significantly increases bandwidth requirements.

Therefore, a robust FL framework is essential one that models IoT data effectively under non-stationary conditions, considers inter-client interactions, preserves data de-

centralization, and operates within the computational and communication constraints typical of IoT environments.

2.4 Manifold Hypothesis

The manifold hypothesis posits that high-dimensional data observed in real-world scenarios, such as images, speech, and text, lie on or near a low-dimensional manifold embedded within the ambient high-dimensional space [117, 118]. This assumption has been central to the development of algorithms for dimensionality reduction, data representation, and deep learning.

Empirical studies support the hypothesis by demonstrating that data such as natural images can often be well-approximated by a lower-dimensional structure [59]. Methods like Isometric Mapping (Isomap) [117] and Locally Linear Embedding (LLE) [118] aim to uncover these intrinsic structures by preserving global and local relationships, respectively. These techniques have facilitated the understanding and visualization of complex datasets.

The manifold hypothesis has also influenced the design of deep learning architectures. Neural networks leverage the hypothesis by structuring data representations to exploit its low-dimensional nature. For example, generative models, such as autoencoders and GANs, implicitly rely on the hypothesis to learn compact representations [119, 120]. Recent works explore generative models that explicitly model data as lying on manifolds, improving robustness and interpretability [121].

In FL, the manifold hypothesis provides a compelling inductive bias for addressing the challenges posed by data heterogeneity and limited communication. This hypothesis suggests that, despite the high dimensionality of observed data, the underlying structure of client data distributions often resides on or near a shared low-dimensional manifold. Leveraging this assumption enables FL algorithms to learn compact, transferable representations that generalize across non-IID clients. Empirical studies have shown that encoding data along intrinsic low-dimensional subspaces facilitates both representation alignment and communication efficiency [122]. For instance, manifold-aware techniques—such as federated representation learning and low-rank model updates—exploit this structure by transmitting only the most informative components, thus reducing communication overhead while preserving critical semantic information [34]. Furthermore, regularization methods that encourage consistency among client-specific manifolds have been shown to enhance model robustness and prevent overfitting, particularly in settings with sparse, personalized, or privacy-constrained data. By aligning learning objectives with the underlying geometric structure of the data, the manifold hypothesis offers a principled and practical foundation for developing scalable, communication-efficient, and generalizable FL algorithms.

2.5 Learning on Grassmann Manifolds

Grounded in the manifold hypothesis, the Grassmann manifold provides a robust geometric framework for modeling data that reside in linear subspaces—a structural assumption prevalent across high-dimensional learning tasks such as dimensionality reduction, low-rank approximation, and geometry-aware representation learning. Formally, for integers $n \geq k > 0$, the Grassmann manifold $G(n, k)$ denotes the set of all k-dimensional linear subspaces in the Euclidean space $\mathbb{R}^n$. Each point on $G(n, k)$ cor-

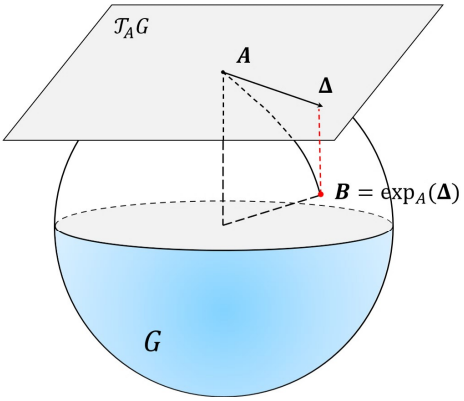

Figure 2.1: Illustration of a geodesic update on the Grassmann manifold. Given a point $A \in G(n,k)$ and a tangent direction $\Delta \in \mathcal{T}_A G$, the exponential map $\exp_A(\Delta)$ yields a new point $B \in G(n,k)$. The manifold contains all matrices $A \in \mathbb{R}^{n\times k}$ such that $A^\top A = I_k$.

responds to the subspace spanned by the columns of an orthonormal matrix $A \in \mathbb{R}^{n\times k}$, where $A^\top A = I_k$. This leads to the canonical definition:

$$G(n,k) = \{\text{span}(A) : A \in \mathbb{R}^{n\times k},\ A^\top A = I_k\},$$

highlighting that the Grassmann manifold parameterizes subspaces rather than specific bases. Consequently, A and AQ, for any orthogonal matrix $Q \in \mathcal{O}(k)$, represent the same point in $G(n,k)$. This property gives rise to the quotient manifold structure:

$$G(n,k) = \mathcal{V}(n,k)/\mathcal{O}(k),$$

where $\mathcal{V}(n,k)$ is the Stiefel manifold of all $n \times k$ matrices with orthonormal columns, and $\mathcal{O}(k)$ is the orthogonal group. This quotient formally encodes the invariance of $G(n,k)$ under right multiplication by orthogonal transformations [68].

For optimization over $G(n,k)$, the differential geometric structure is essential. At a point $A \in G(n,k)$, the tangent space $\mathcal{T}_A G \subset \mathbb{R}^{n\times k}$ is defined as the set of matrices Δ satisfying the orthogonality constraint

$$A^\top \Delta = 0,$$

ensuring that Δ is perpendicular to the current subspace span(A). These tangent vectors describe feasible infinitesimal directions for movement on the manifold. This concept is visually illustrated in Figure 2.1, where a tangent vector at point A on the manifold defines a direction for movement along the geodesic.

Gradient-based optimization on the Grassmann manifold involves computing the gradient of a function $F : G(n,k) \to \mathbb{R}$ by projecting the Euclidean gradient $F_A = \frac{\partial F}{\partial A}$ onto the tangent space at A via the orthogonal projection

$$F_A \longrightarrow \nabla_A F,$$

where $\nabla_A F$ denotes the projected gradient given by

$$\nabla_A F = (I_k - AA^\top)F_A,$$

with the matrix $(I_k - AA^\top)$ serving as the orthogonal projector onto the orthogonal complement of A. Since $A^\top \nabla_A F = 0$, this projection guarantees that $\nabla_A F$ is a valid tangent vector. To ensure that each update remains on the Grassmann manifold, the tangent vector must be mapped back onto $G(n, k)$. One approach is the *exponential map*, which defines geodesically exact updates along the manifold:

$$\exp_A(\Delta) = \text{geodesic path from } A \text{ in the direction of } \Delta.$$

However, the exponential map is computationally expensive for practical use. As an efficient alternative, *retractions* are employed to approximate these updates. Among retraction methods, QR decomposition is widely adopted because it provides a simple and reliable way to map updates back onto the manifold. Given a step size $\eta > 0$, the updated point is computed as

$$B = R(A - \eta \nabla_A F),$$

where $R(\cdot)$ denotes the operation that orthonormalizes its argument, typically via QR decomposition. This guarantees $B \in G(n, k)$, preserving the manifold constraint. This framework enables efficient and principled optimization directly on the Grassmann manifold and has been widely applied in tasks involving subspace tracking, domain adaptation, dictionary learning, and robust low-rank estimation [123, 124].

The concept of Grassmann manifolds has also emerged as a versatile mathematical tool with applications spanning multiple disciplines, including computer vision, machine learning, wireless communication systems, and natural language processing. In the field of computer vision, Grassmann manifolds are employed for tasks like object recognition and categorization by capturing spaces invariant to changes in lighting or orientation [125, 126]. Within statistical learning, methods based on Grassmannian geometry have been introduced for applications such as discriminant analysis [127–129] and clustering [130, 131], as well as for optimizing learning algorithms. These methods leverage the Grassmann manifold to handle data structures specific to these spaces or to use Grassmannian-based optimization for enhancing algorithmic efficiency. For instance, in recommendation systems, challenges like matrix completion under constraints such as sparsity or low rank have been successfully addressed with Grassmannian approaches [132, 133]. Similarly, in wireless communications, Grassmannian packing techniques are utilized for designing space-time code constellations [134–136]. These applications exploit the geometric properties of Grassmann manifolds to improve communication efficiency. Additionally, in natural language processing, the manifold framework has been used to model affine subspaces within language models tailored to specific documents [137], enabling structured dimensionality reduction and basis-invariant optimization. Recent progress includes leveraging Grassmann manifolds in deep learning for tasks such as shape retrieval and alignment [138], geodesic convolutional networks [139], and incorporating Riemannian geometry into neural network designs [140]. New neural architectures tailored for data on Grassmann manifolds have also been proposed [141, 142]. The key advantage

of Grassmannian methods lies in their capacity to exploit the intrinsic structure of problems, leading to greater computational efficiency and enhanced outcomes across a wide range of scenarios.

Despite the growing interest in geometric methods for machine learning, the integration of Grassmannian geometry into FL remains largely unexplored, presenting a promising direction for addressing key challenges inherent to decentralized optimization. One of the most fundamental obstacles in FL is statistical heterogeneity: client data are typically drawn from non-identically distributed sources due to variations in user behavior, device characteristics, and local environments [28–30]. This non-IID nature leads to divergent local updates that often span distinct subspaces [143, 144], making global aggregation difficult and reducing convergence stability. Grassmann manifolds, which parameterize the space of fixed-dimensional linear subspaces, offer a natural geometric framework for modeling such variability. By interpreting client updates or feature representations as points on a Grassmann manifold, one can perform geometry-aware operations such as geodesic interpolation, subspace alignment, and distance-preserving averaging, which reduce inter-client discrepancy while respecting the underlying structure of the data [145–147]. Moreover, the low-rank nature of Grassmannian representations facilitates compact encoding of model updates [148, 149], enabling efficient communication—an essential requirement in resource-constrained FL scenarios. These techniques are especially relevant in cross-device and personalized FL, where models must adapt to user-specific data distributions without excessive computation or communication overhead. Taken together, these insights suggest that Grassmannian learning provides both a mathematically principled and practically scalable approach to mitigating heterogeneity, improving generalization, and enhancing the robustness of federated optimization.

2.6 Distributed Optimization via the Alternating Direction Method of Multipliers

The Alternating Direction Method of Multipliers (ADMM) has emerged as a foundational algorithm in distributed optimization, particularly valued for its ability to decompose large-scale problems into subproblems that can be solved in parallel [150]. By combining dual decomposition with the method of multipliers, ADMM enables coordination among distributed agents through local updates and global variable synchronization, thus offering scalability and computational efficiency. Its core formulation is especially attractive in scenarios where data or computation is inherently decentralized, such as sensor networks, FL and multi-agent systems.

In convex settings, ADMM has been rigorously analyzed, with strong convergence guarantees under relatively mild assumptions [151–153]. The modular nature of ADMM allows it to be applied across a wide spectrum of machine learning and signal processing problems, including LASSO, support vector machines, and sparse coding [154, 155]. However, despite this maturity in convex analysis, the behavior of ADMM in more realistic, nonconvex environments remains a challenging and less understood area of research.

Recent efforts have extended ADMM to nonconvex formulations, motivated by applications such as matrix factorization, tensor decomposition, and distributed learning

with deep models. For instance, Wang et al. [156] studied multi-convex problems where each subproblem is convex but the overall objective is not, and established convergence to stationary points. Further extensions have addressed nonconvex and nonsmooth objective functions [157], nonlinearly coupled constraints [158], and quadratic penalties [159], all contributing to a growing body of theoretical insights. Nonetheless, convergence guarantees in these settings are typically weaker than in the convex case, and often restricted to local optimality under restrictive assumptions.

The use of ADMM for training deep neural networks in distributed environments adds another layer of complexity. Although empirical studies have reported performance improvements over SGD in certain regimes [160, 161], the nonconvexity and nonlinearity of deep models exacerbate convergence issues. In such cases, ADMM may converge to suboptimal or unstable solutions, particularly when model updates are noisy or communication is constrained.

In the context of FL, ADMM has garnered attention for its natural alignment with the requirements of decentralized training. FL systems aim to collaboratively train models across a network of clients, each holding private, possibly non-IID data. ADMM-based methods address this by decomposing the global optimization objective into local subproblems solved independently, followed by synchronization through dual variable updates [162, 163]. This decomposition not only reduces communication overhead but also allows for flexible update schemes, including asynchronous and semi-synchronous protocols.

To improve ADMM's practical utility in FL, several variants have been developed. These include strategies such as gradient compression and quantization [164], adaptive penalty parameter tuning [165], and hierarchical aggregation for multi-level systems [166]. Nevertheless, the interplay between statistical heterogeneity, client dropouts, and nonconvexity continues to pose theoretical and algorithmic challenges. In particular, the absence of strong convexity in local objectives and limited participation across rounds hinder convergence analysis and performance guarantees.

In response, ongoing research has explored hybrid methods that integrate ADMM with momentum acceleration [167], variance reduction [168], and gradient tracking [169] to stabilize updates and improve convergence in heterogeneous, nonconvex environments. Moreover, recent works have started to establish convergence rates under weaker assumptions and have begun to design privacy-preserving extensions of ADMM tailored for FL scenarios [170]. These directions highlight a growing recognition of ADMM's potential in handling the dual challenges of distributed optimization and nonconvexity in FL systems.

Chapter 3

Federated PCA on Grassmann Manifolds for Anomaly Detection in IoT Networks

3.1 Introduction

The proliferation of IoT technologies has enabled smart devices to share information and coordinate actions with minimal human intervention, leading to a significant improvement in the quality of life. These intelligent devices have become integral across various sectors, including climate systems [171], smart city infrastructures [172], and energy systems [173]. In recent years, the number of IoT devices has grown exponentially, resulting in an increasingly interconnected and complex ecosystem. However, this rapid expansion has also heightened the importance of robust network security, as many IoT devices possess inherent security vulnerabilities due to their limited computational resources [100]. As a result, the need for effective methods to detect and mitigate network attacks has become a critical challenge.

In the intricate domain of IoT security, machine learning-based intrusion detection systems (ML-IDS) play a vital role in safeguarding networks from potential threats. The effectiveness of ML-IDS largely depends on their ability to model normal behavioral patterns and detect anomalies as deviations from these learned profiles. This capability spans a spectrum of methodologies, ranging from conventional algorithms to advanced deep learning approaches [174]. Despite their promise, supervised ML-IDS encounter significant limitations, notably the requirement for extensive labeled datasets, which are resource-intensive and time-consuming to generate, as well as challenges in managing high-dimensional data. Moreover, these models often struggle with detecting novel anomalies that fall outside the scope of the training data.

To cope with these problems, unsupervised Machine Learning-based Intrusion Detection Systems (ML-IDS), including methods like AutoEncoders [175] and GAN-based models [176, 177], have been introduced. These approaches enable learning representations of normal behavior without requiring labeled data, thereby enhancing their adaptability for anomaly detection tasks. Nevertheless, the substantial computational requirements and the limited interpretability of these models pose challenges to their deployment in resource-constrained IoT environments [178]. Among various unsu-

pervised methods, PCA has emerged as a widely adopted approach [179, 180]. PCA effectively captures the underlying relationships within data, allowing for anomaly detection by identifying deviations in new observations from established patterns. Moreover, PCA offers enhanced detection speed compared to many supervised models [100]. However, applying PCA in a centralized manner for IoT data introduces challenges, particularly due to the high computational costs of data aggregation and the critical requirement of preserving data privacy in IoT networks [181]. Therefore, developing effective anomaly detection frameworks for IoT demands a careful balance between computational efficiency and privacy preservation, emphasizing local data processing while maintaining robust detection capabilities.

In response to these constraints, traditional intrusion detection systems (IDS) offer an alternative through hierarchical architectures for detecting attacks and anomalies [95, 97, 182]. This architecture involves local IoT devices connecting to a local access gateway, which transmits real-time data to a global security gateway. The global gateway subsequently distributes suitable anomaly detection models [183] to local gateways, enabling continuous monitoring of communications and detection of abnormal behaviors. This distributed approach not only improves the identification of security threats but also addresses privacy concerns by localizing data processing, thereby reducing the need for extensive data transmission. Building on this, FL-based IDS has been introduced to further address privacy challenges in anomaly detection [97, 99, 184]. In a federated IDS, multiple local gateways serve as FL clients, participating collaboratively in the anomaly detection process. During training, local gateways retain their data locally, sharing only the learned models with the global security gateway. While many existing federated IDS approaches and traditional IDS methods rely on *supervised learning methods* for intrusion detection, they encounter conventional challenges such as the need for labeled network datasets and the difficulty of identifying unseen anomalies [182, 184]. Additionally, the training of complex models is often computationally intensive, demanding substantial time, memory resources and bandwidth for effective deployment in IoT networks.

In this study, we propose a novel FL framework for ML-IDS in IoT networks using PCA, an effective *unsupervised anomaly detection* approach, to address above challenges. To build up an ML-IDS on the FL framework, we first formulate a novel federated PCA optimization problem, namely `FedPCA`, to learn the profile of normal behavior in distributed network datasets. We then design an ADMM-based algorithm `FedPE` to solve the proposed `FedPCA` problem. Specifically, in `FedPE`, each local gateway aims to find a local low-rank representation matrix on Euclidean space to capture the most variability in its own data such that the reconstruction error, averaged over all clients, is minimized in a distributed setting. The optimal representation matrix is treated as the hidden profile of normal behaviors of the IoT network, which deviates significantly in anomalous observations. A key distinction of `FedPCA` is the incorporation of an orthogonal matrix constraint, which resides on a manifold. Specifically, the unique manifold structure of the `FedPCA` constraint necessitates further exploration of Grassmann manifolds [185]. In response, we propose an ADMM-based learning algorithm on Grassmann manifolds, termed `FedPG`. The Grassmann manifold is particularly well-suited for identifying low-dimensional subspaces, allowing it to effectively leverage the structural information inherent to the problem, leading to

faster convergence and improved anomaly detection performance.

In this work, each IoT device functions as an autonomous anomaly detection unit, actively contributing to the overall security of the network. These devices continuously monitor their communication patterns, employing anomaly detection models to identify deviations that may indicate potential threats. By implementing machine learning-based ML-IDS at the IoT device level, we create a distributed network of vigilant nodes, each capable of real-time threat detection. This decentralized approach enables IoT devices to participate in the learning process without sharing sensitive local data, which is crucial in maintaining privacy in IoT environments where user data security is a primary concern. Consequently, this framework enhances the resilience and security of the IoT network while offering faster detection times compared to traditional unsupervised methods. In this section, we bridge the gap by the following contributions:

- We propose an unsupervised federated PCA framework for efficient host-based IoT anomaly detection, which is formulated by a consensus optimization problem for privacy-preserving and communication-efficient.
- We introduce a novel algorithm for the `FedPE` and `FedPG` frameworks, developed using ADMM-based procedures integrated with client sub-sampling techniques to improve robustness and reduce communication overheads.
- We provide a rigorous theoretical convergence analysis for the proposed ADMM-based algorithms, establishing that `FedPE` and `FedPG` converge under a sub-sampling scheme, which constitutes a novel contribution.
- We conduct extensive experiments on the UNSW-NB15 [186], and TON-IoT [187] network datasets to demonstrate that `FedPE` and `FedPG` not only achieve competitive performance compared to non-linear methods in anomaly detection but also deliver significant improvements in communication and memory efficiency. This comprehensive evaluation underscores the robustness of our framework in handling non-i.i.d. data distributions, rendering it highly applicable for IoT networks.

3.2 System Model

We consider an IoT network comprised of a central global coordinator CO and a set of N IoT devices, each denoted by a unique identifier ID_i holding a local data set X_i, where $i \in \{1, 2, ..., N\}$. This network interacts within a client-server topology, as depicted in Fig. 3.1. Our framework employs a host-based IDS strategy, where each IoT device operates as a sentinel within the network, examining its network traffic and identifying potential security threats. This distributed approach significantly increases the network's resilience by facilitating real-time threat detection and mitigation. Furthermore, since the learning is local to each device, it allows for a reduction in communication overhead and strengthens privacy by keeping the data on the device.

3.2.1 Principal Component Analysis Revisiting

We first revisit the basics of PCA to formally define our method. PCA projects a given dataset X onto principal components ordered by the amount of variance they capture in

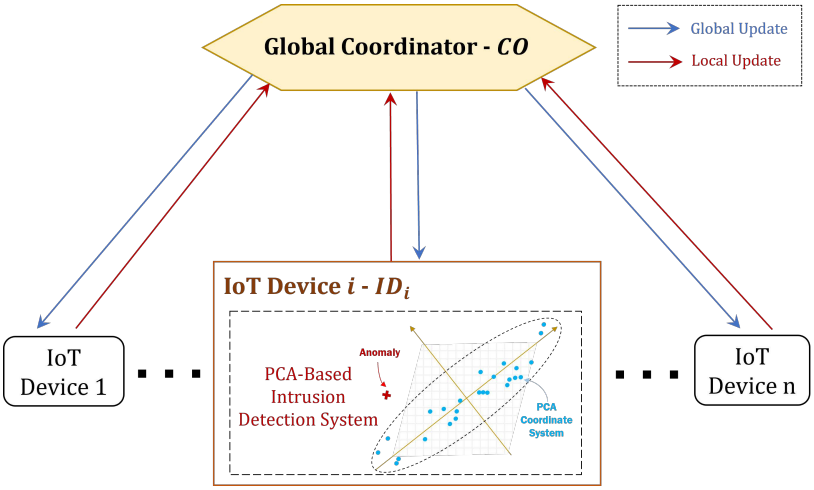

Figure 3.1: Overview of a host-based IoT anomaly detection system utilizing Federated PCA.

the data [188]. Specifically, it seeks an optimal rank-k ($k < d$) matrix $\tilde{X}$ that minimizes the reconstruction error $\|X - \tilde{X}\|_F$, where $||\cdot||_F$ denotes the Frobenius norm [189]. Typically, $\tilde{X}$ is derived based on the singular value decomposition (SVD) of X, i.e., $X = U\Sigma V^\top$, where $U \in \mathbb{R}^{d\times d}$ is an orthonormal matrix containing the left singular vectors of X, $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_d)$ contains the singular values ($\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d$), and $V \in \mathbb{R}^{D\times D}$ contains the right singular vectors of X. Let $U^{(k)} \in \mathbb{R}^{d\times k}$ be the first k columns of U and $\Sigma_k = \text{diag}(\sigma_1, \ldots, \sigma_k)$. We can rewrite the solution for $\tilde{X}$ as

$$\tilde{X} = U^{(k)}U^{(k)\top}X, \tag{3.1}$$

which can be interpreted as the projection of data matrix X onto the column space of $U^{(k)}$. The PCA problem can thus be formulated as an optimization problem as follows:

$$\begin{aligned} \min_{U\in\mathbb{R}^{d\times k}} \quad & \|(I - UU^\top)X\|_F^2 \\ \text{s.t.} \quad & U^\top U = I. \end{aligned} \tag{3.2}$$

The essence of this problem is to locate an orthogonal matrix U that minimizes the reconstruction error of X when projected onto the column space of U, resulting in a matrix $Z = U^\top X \in \mathbb{R}^{k\times D}$ provides a lower-dimensional representation of X.

In the context of developing ML-IDS for IoT networks, solving this PCA problem typically requires IoT devices to transmit their local data to a centralized server. While this approach offers a streamlined method for anomaly detection, it is hampered by two main obstacles. First, data transferring within the resource-limited and bandwidth-constrained landscape of IoT can lead to significant *communication bottlenecks*, which in turn might escalate operational costs, amplify latency, and induce network congestion. Second, the act of centralizing such data amplifies *privacy concerns* as it risks exposing sensitive information from personal environments, posing serious threats to user confidentiality. Hence, an alternative would be to seek distributed methods that address these issues while still effectively detecting anomalies.

3.2.2 Federated PCA via ADMM Consensus Optimization

To address these concerns, we propose a federated approach for solving PCA on distributed datasets, namely `FedPCA`, which aims to collectively learn a *common representation* from all local datasets, $X_1, X_2, ..., X_N$, without the CO directly accessing these datasets. We first introduce a set of local variables $U_1, \ldots, U_N$, where U_i is the local PCA matrix learned by client i. Then, we formulate a consensus optimization problem as follows:

$$\begin{aligned} \min_{U_1,\ldots,U_N} \quad & \sum_{i=1}^{N} \|(I - U_i U_i^\top) X_i\|_F^2 \\ \text{s.t.} \quad & U_i = Z, \ \forall i = 1, \ldots, N. \\ & U_i^\top U_i = I, \ \forall i = 1, \ldots, N, \end{aligned} \tag{3.3}$$

where the first linear constraint enforces the "consensus" to a *common low-rank matrix* Z [190] for all local U_i. The second constraint ensures that the matrix U_i is orthonormal for every client keeping the distributed problem consistent with the classical centralized PCA. However, addressing the non-linearity introduced by this orthonormality constraint is a challenging task due to the non-convex nature of the problem. Here, we employ a surrogate technique, as introduced in [151], to handle the above non-linear constraint as follows.

$$h_i(U_i) = \max\{0, U_i^\top U_i - I\}^2, \tag{3.4}$$

where the operators max and squared are component-wise. By reformulating the problem to the $h_i(U_i)$ function, we gently steer the optimization towards matrices U_i that approach orthonormality, making the landscape more tractable for iterative frameworks. We then have the problem (3.3) equivalent to the following:

$$\begin{aligned} \min_{U_1,\ldots,U_N} \quad & \sum_{i=1}^{N} \|(I - U_i U_i^\top) X_i\|_F^2 \\ \text{s.t.} \quad & U_i = Z, \ \forall i = 1, \ldots, N \\ & h_i(U_i) \leq 0, \ \forall i = 1, \ldots, N. \end{aligned} \tag{3.5}$$

Given the nature of problem (3.5) as a consensus optimization with linear constraints, we adopt the *alternating direction method of multipliers* (ADMM) [150] to address it. ADMM is an iterative optimization framework that decomposes complex problems into simpler subproblems, which are then coordinated through dual variable updates. It has found widespread application in distributed and large-scale optimization tasks, including non-convex settings. The essence of this iterative framework is the alternating updates of primal and dual variables to achieve two main objectives: the U_i across clients converge towards a common Z and the objective function is iteratively minimized. Applying this to our problem, we first construct the augmented Lagrangian function associated with the problem (3.5) as follows.

$$\begin{aligned} \mathcal{L}(\{U_i\}, Z, \{Y_i\}, \{T_i\}) &= \sum_{i=1}^{N} f_i(U_i) + \sum_{i=1}^{N} \langle Y_i, U_i - Z \rangle_F \\ &+ \sum_{i=1}^{N} \langle T_i, h_i(U_i) \rangle_F + \frac{\rho}{2} \sum_{i=1}^{N} \|U_i - Z\|_F^2 + \frac{\rho}{2} \sum_{i=1}^{N} \|h_i(U_i)\|_F^2, \end{aligned} \tag{3.6}$$

Algorithm 1 FedPCA in Euclidean Space (`FedPE`)

Randomly initialize Z^0 and $U_i^0, \quad \forall i = 1, \ldots, N$
for $k = 0, \ldots, T-1$ **do**
  Sample subset clients $\mathcal{S}^k$
  for each client $i \in \mathcal{S}^k$ in parallel **do**
    $U_i^{k+1} = \arg\min_{U_i} \Big\{ f_i(U_i) + \langle Y_i^k, U_i - Z^k \rangle_F + \langle T_i^k, h_i(U_i) \rangle_F + \frac{\rho}{2} \|U_i - Z^k\|_F^2 + \frac{\rho}{2} \|h_i(U_i)\|_F^2 \Big\}$
  end for
  CO updates $Z^{k+1} = \frac{1}{|\mathcal{S}^k|} \sum_{i \in \mathcal{S}^k} (U_i^{k+1} + \frac{1}{\rho} Y_i^k)$
  CO broadcasts Z^{k+1} to all clients
  for each client $i \in \mathcal{S}^k$ in parallel **do**
    $Y_i^{k+1} = Y_i^k + \rho \Big(U_i^{k+1} - Z^{k+1} \Big)$
    $T_i^{k+1} = T_i^k + \rho \, h_i(U_i^{k+1})$
  end for
end for

where $f_i(U_i) = \|(I - U_i U_i^\top) X_i\|_F^2$ is the objective function, $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product between two matrices.

This augmented Lagrangian makes the optimization problem more amenable to iterative solution techniques while ensuring that constraints are respected. The terms with Lagrange multipliers (or dual variables) – $\sum_{i=1}^N \langle Y_i, U_i - Z \rangle_F$ and $\sum_{i=1}^N \langle T_i, h_i(U_i) \rangle_F$ – quantify the influence of the constraints on achieving a common PCA representation across distributed datasets. They ensure that each local PCA matrix U_i converges to a shared matrix Z and adheres to the non-linear constraints dictated by $h_i(U_i)$. On the other hand, the penalty terms – $(\rho/2) \sum_{i=1}^N \|U_i - Z\|_F^2$ and $(\rho/2) \sum_{i=1}^N \|h_i(U_i)\|_F^2$ – are governed by the penalty parameter ρ to penalize any deviation of the local PCA matrices U_i from the global consensus matrix Z or from satisfying the orthogonality conditions $h_i(U_i)$.

Since, the Frobenius inner product for matrices is analogous to the standard dot products for vectors, we can obtain the gradients of the augmented Lagrangian (4.7) w.r.t. dual variables Y_i and T_i as follows.

$$\nabla_{Y_i} \mathcal{L}_\rho(U, Z, Y, T) = U_i - Z$$

$$\nabla_{T_i} \mathcal{L}_\rho(U, Z, Y, T) = h_i(U_i)$$

To solve the proposed problem, we employ an ADMM-based procedure in which a subset of clients, denoted as $\mathcal{S}^k$, performs iterative updates using a step size ρ during the k-th communication round as follows.

Primal Update (Local Update)

$$U_i^{k+1} = \operatorname*{argmin}_{U_i} \Big\{ f_i(U_i) + \langle Y_i^k, U_i - Z^k \rangle_F + \langle T_i^k, h_i(U_i) \rangle_F + \frac{\rho}{2} \|U_i - Z^k\|_F^2 + \frac{\rho}{2} \|h_i(U_i)\|_F^2 \Big\} \tag{3.7}$$

Consensus Update (Global Update)

$$Z^{k+1} = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \Big(U_i^{k+1} + \frac{1}{\rho} Y_i^k \Big) \tag{3.8}$$

Dual Update (Local Updates)

$$Y_i^{k+1} = Y_i^k + \rho \Big(U_i^{k+1} - Z^{k+1} \Big) \tag{3.9}$$

$$T_i^{k+1} = T_i^k + \rho \, h_i(U_i^{k+1}). \tag{3.10}$$

When $|\mathcal{S}_k| = N$, substituting Z^{k+1} to Y_i^{k+1} results in $\frac{1}{N} \sum_{i=1}^{N} Y_i^{k+1} = 0$. This which means Z update can be rewritten as [190]:

$$Z^{k+1} = \frac{1}{N} \sum_{i=1}^{N} U_i^{k+1}. \tag{3.11}$$

These steps are repeated iteratively until convergence, at which point the local PCA matrix U_i and the consensus variable Z agree, resulting in a common matrix learned from the distributed data. This ADMM-based framework collaboratively drives towards a consensus PCA solution while ensuring that the unique characteristics of each dataset are preserved.

The above procedure is denoted as `FedPCA` in Euclidean space, abbreviated as `FedPE`. The full procedure is detailed in Alg. 1. Specially, in each round, a distinct subset of clients $\mathcal{S}^k$ is selected to perform the update (Alg. 1, line 3). Each selected client is then involved in solving the primal problem (3.7) on Euclidean space locally using gradient-based methods (Alg. 1, line 5). Once finishing training, selected clients send their local PCA matrix to the CO for updating the common PCA matrix Z (Alg. 1, line 7). Finally, each selected client i update its dual variable Y_i and T_i based on the latest U_i and Z (Alg. 1, lines 10-11). One main difference between `FedPE` in this work and the one presented in [191] is that we employ a client subsampling strategy during the federated training process. This offers significant benefits, including reduced communication overhead, improved scalability for large-scale IoT networks, and enhanced resilience to stragglers or non-responsive devices.

3.2.3 FedPCA using ADMM on Grassmann Manifolds

Although `FedPE` effectively addresses the proposed problem, we further extend its capacity by formulating `FedPCA` on Grassmann manifolds, denoted as `FedPG`, to accelerate convergence. The motivation for adopting Grassmannians lies in the geometric structure of the PCA problem: since we are interested in identifying a low-dimensional subspace rather than specific basis vectors, it is natural to perform optimization on the Grassmann manifold, which treats all orthonormal bases of the same subspace as equivalent. In fact, PCA solutions are inherently non-unique—any orthonormal basis that spans the same subspace is an equally valid solution, due to the rotational invariance of the objective. This ambiguity makes the Grassmann manifold, which represents equivalence classes of such bases, a more appropriate optimization domain than the Stiefel manifold, which distinguishes between different bases of the

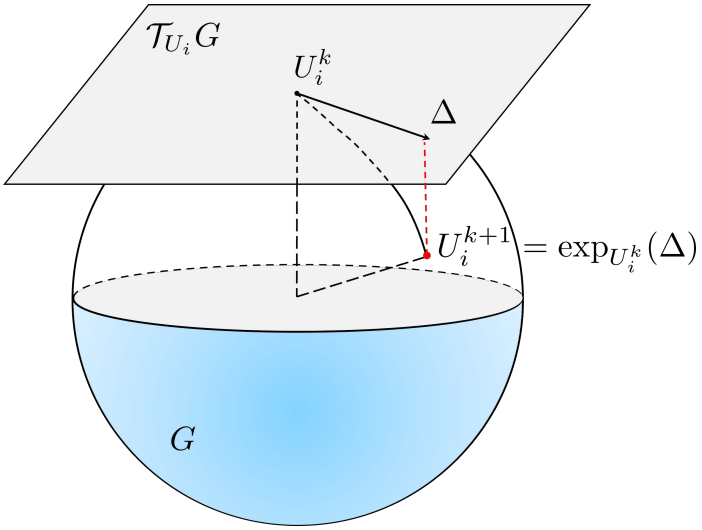

Figure 3.2: Illustration of movement on a Grassmann manifold, represented by a sphere. Given a point U_i^k on Grassmann manifold G and a vector Δ on the tangent space denoted by $\mathcal{T}_{U_i}G$ at U_i^k, a point U_i^{k+1} is identified by exponential mapping $\exp_{U_i^k}(\Delta)$. In our context, the manifold G contains all matrix U_i satisfy the condition $U_i^\top U_i = I$.

same subspace. Notably, the orthonormality constraint utilized in (3.3) implies that the parameters updated in (3.7) can be naturally projected onto a Grassmann manifold, a structure widely adopted in subspace optimization problems such as PCA [185].

Orthonormality meets the Grassmannian

The Grassmann manifold, denoted as $G(n, d)$, represents a geometric space that encapsulates sets of d-dimensional subspaces within an overaching n-dimensional real or complex Euclidean space, subject to $n \geq d > 0$ [185]. The distinct of this manifold lies in its representation capacity: Any point on $G(n, d)$ can be identified by an $n \times d$ orthogonal matrix U whose column spans the corresponding subspace to ensure orthonormality $U^\top U = I_d$. Hence, $G(n, d)$ is defined as follows.

$$G(n, d) = \{\text{span}(U) : U \in \mathbb{R}^{n\times d}, U^\top U = I_d\} \tag{3.12}$$

Fig. 3.2 shows an example of the Grassmann manifold as the surface of a sphere. Here, the tangent space $\mathcal{T}_U G$ at U captures the set of all directions in which the subspace can "move" but still stay very close to the manifold. By capturing the space of all orthonormal subspaces, the Grassmann manifold offers an elegant framework for addressing optimization problems with orthonormality constraints (e.g., problem (3.3)) since these constraints are automatically maintained when operations are performed within this manifold.

FedPCA via Grassmann Manifold Optimization

By framing problem (3.3) within the Grassmann manifold, the augmented Lagrangian (4.7) evolves as follows.

$$\mathcal{L}(\{U_i\}, Z, \{Y_i\}) = \sum_{i=1}^{N} f_i(U_i) + \sum_{i=1}^{N} Y_i^\top (U_i - Z) + \frac{\rho}{2} \sum_{i=1}^{N} \|U_i - Z\|_F^2. \tag{3.13}$$

Algorithm 2 FedPCA on Grassmann Manifold (`FedPG`)

1: Randomly initialize Z^0 and $U_i^0, \quad \forall i = 1, \ldots, N$
2: **for** $k = 1, \ldots, T$ **do**
3: Sample subset clients $\mathcal{S}^k$
4: **for** each client $i \in \mathcal{S}^k$ in parallel **do**
5: $U_i^{k,0} = U_i^k$
6: **for** $c = 1, \ldots, C$ **do**
7: $\nabla_U F_i^k(U_i^{k,c}) = (I_d - U_i^{k,c} U_i^{k,c\top}) F_i^k(U_i^{k,c})$
8: $U_i^{k,c+1} = R(U_i^{k,c} - \eta \nabla_U F_i^k(U_i^{k,c}))$
9: **end for**
10: $U_i^{k+1} = U_i^{k,C}$
11: **end for**
12: CO updates $Z^{k+1} = \frac{1}{|\mathcal{S}^k|} \sum_{i \in \mathcal{S}^k} (U_i^{k+1} + \frac{1}{\rho} Y_i^k)$
13: CO broadcasts Z^{k+1} to all clients
14: **for** each client $i \in \mathcal{S}^k$ in parallel **do**
15: $Y_i^{k+1} = Y_i^k + \rho\Big(U_i^{k+1} - Z^{k+1}\Big)$
16: **end for**
17: **end for**

Here, the terms associated with $h_i(U_i)$ are eliminated because the inherent orthonormality constraints are satisfied.

Denoting the local update for each client i at the k-th iteration by following:

$$F_i^k(U_i) := f_i(U_i) + Y_i^{k\top}(U_i - Z^k) + \frac{\rho}{2}\|U_i - Z^k\|_F^2 \tag{3.14}$$

We then recast the update rules for U_i and Z on the Grassmann manifold as follows.

$$U_i^{k+1} = \arg\min_{U_i} \Big\{ F_i^k(U_i) \Big\} \tag{3.15}$$

As detailed in Alg. 5 (lines 5-8), projected gradient descent on the Grassmann manifold is applied in every local iteration in clients to find the update of U_i^{k+1} in (4.16). Considering the definition of the Grassmann manifold in (3.12), we enforce the second constraint $U_i^T U_i = I$ by using the projected gradient descent on the Grassmann manifold (Alg. 5, line 8) as follows

$$U_i^{k+1} = R(U_i^k - \eta \nabla_U F_i^k(U_i^k)), \tag{3.16}$$

Here, η is the step size and $\nabla_U F_i^k(U_i^k)$ is the gradient of $F_i^k(U_i^k)$ at the point U_i^k, and the retraction function $R(\cdot)$ is the projection operation performed by QR decomposition [185]. The projection process for U_i^k and U_i^{k+1} is graphically illustrated in Fig. 3.2. We compute the updates by projecting the Euclidean gradient onto the tangent space of the manifold using orthogonal projection $F_i^k(U_i^k) \to \nabla_U F_i^k(U_i^k)$ (Alg. 5, line 7) as follows.

$$\nabla_U F_i^k(U_i^k) = (I_d - U_i^k U_i^{k\top}) F_i^k(U_i^k). \tag{3.17}$$

Note that Equations (4.31) and (4.30) are repeatedly updated in C local rounds to solve Equation (4.16) (Alg. 5, lines 6-9).

The use of Grassmannian gradients in `FedPG` enables more precise and effective steps toward the optimal solution, thus accelerating convergence. Furthermore, PCA problems often include orthogonality constraints, which are complex to handle in traditional Euclidean spaces and slow the convergence rate. `FedPG` elegantly deals with these constraints by projecting the problem onto the Grassmann manifold, where orthogonality constraints are naturally satisfied. By doing so, it eliminates the need for extra steps to manage these constraints, reducing computational overhead and facilitating faster convergence.

Moreover, `FedPG` ensures an exact solution due to its unique feature of projection onto the Grassmann manifold. This process maintains the inherent orthogonality conditions that PCA requires, effectively preserving the constraints of the original problem. Therefore, the solutions derived from `FedPG` are precise and exact as they are always in the feasible region. On the other hand, `FedPE`'s solutions are approximations. While `FedPE` works effectively in solving the problem, it doesn't inherently uphold the orthogonality conditions. The resulting solutions, therefore, might be outside the feasible region, and additional steps are necessary to bring the solution back into the feasible space. This could lead to approximation errors, and therefore the solutions are not always exact.

3.2.4 Computational Complexity

Here, we study the strength of computational complexity of the proposed algorithm `FedPG`. We recall that d and k denote the input data dimension and the number of PCA components, respectively. C and T denote the number of local and global rounds, respectively.

In `FedPG`, the major local-client computational costs in each global communication round come from two parts, calculating the gradient updates for the matrix U_i^k (Lines 7-8 in Algorithm 5) and performing consensus updates for the matrices Y and T (Lines 15-16 in Algorithm 5). Especially, computing the orthogonal projection to obtain the gradient costs at most $O(2d^2k + 2dk)$ flops for two matrix-matrix multiplications (Line 7 in Algorithm 5). Once the gradient is calculated, the retraction of the update step on the Grassmann manifold is done by applying QR decomposition with complexity $O(dk + dk^2)$ (Line 8 in Algorithm 5). The Y and T update phase needs simple matrix-matrix multiplication requiring $O(d^2k)$ flops (Line 16 in Algorithm 5). Suppose the algorithm is terminated after C rounds, then the overall computational complexity for each local security gateway is given by $O(C(d^2k + dk + dk^2))$. Therefore, FedPG's complexity is (i) independent of the data side D_i, and (ii) dominated by two factors d and k. In IDS applications, the network data is often not too high-dimensional (i.e, the value of d is not too large [192] and $k < d$, as confirmed in the experiments in next section) making this computational complexity acceptable for limited-resource computing IoT devices.

At each global communication round, the global coordinator mainly updates the matrix Z by averaging the sum of U matrices from a subset $S = |10\%N|$ client number, hence costing $O(Sdk)$ flops. This leads to the overall computational complexity of the global coordinator for T rounds as $O(TSdk)$ flops.

3.3 FedPCA: Convergence Analysis

In this section, we present a theoretical analysis to understand the convergence properties of `FedPG` and `FedPE` within the `FedPCA` framework. To the best of our knowledge, this is the first convergence analysis of a federated PCA algorithm conducted on both the Euclidean and Grassmann manifolds since 2024. While prior works have studied convergence for federated PCA under certain settings, our work provides novel formal guarantees that simultaneously address several practical and theoretical challenges. Specifically, our analysis differs from prior works in the following key aspects:

- It is conducted with matrix-valued variables, capturing the natural structure of PCA rather than relying on vectorized simplifications.
- It leverages the consensus ADMM framework to address the decentralized optimization inherent in federated learning.
- It establishes convergence guarantees under a sub-sampling scheme, reflecting practical scenarios where clients use partial data in each iteration.

To begin with, we consider an FL environment with N clients, indexed by $i = 1, 2, \ldots, N$, and introduce the following notations for the analyses:

1. $\mathcal{S}^k \subset \{1, 2, \ldots, N\}$: denotes the subset of clients selected in the k-th iteration.
2. $k(i)$: denotes the latest iteration at which the client i has its model updated.

The orthonormality constraint in (3.3) is a critical component for our proposed method. While `FedPE` handles this by introducing a surrogate linear constraint in (4.5), `FedPG` relaxes this by leveraging projected gradient descent on Grassmann manifolds. Hence, to ensure the generality of our framework, we employ the augmented Lagrangian (4.15) and rewrite it to adapt for the convergence analysis of both algorithms as follows.

$$\begin{aligned} \mathcal{L}(\{U_i\}, Z, \{Y_i\}) = &\sum_{i=1}^{N} f_i(U_i) + \sum_{i=1}^{N} Y_i^\top (U_i - Z) + \frac{\rho_i}{2} \sum_{i=1}^{N} \|U_i - Z\|_F^2 \\ \text{s.t.} \quad &U_i^\top U_i = I. \end{aligned} \tag{3.18}$$

In addition, we modify the global update (4.10) by following:

$$Z^{k+1} = \frac{1}{|S_k|} \sum_{i \in S_k} (U_i^{k+1} + \frac{1}{\rho_i} Y_i^k) \tag{3.19}$$

Accordingly, the local updates for U_i^{k+1} in (3.7) and Y_i^{k+1} in (4.11) at a selected client i will be changed as listed below.

$$\begin{aligned} U_i^{k+1} = \arg\min_{U_i} &\Big\{ f_i(U_i) + \langle Y_i^k, U_i - Z^k \rangle_F + \frac{\rho_i}{2} \sum_{i=1}^{N} \|U_i - Z^k\|_F^2 \Big\} \\ \text{s.t.} \quad &U_i^\top U_i = I \end{aligned} \tag{3.20}$$

$$Y_i^{k+1} = Y_i^k + \rho_i \Big(U_i^{k+1} - Z^{k+1} \Big) \tag{3.21}$$

It is worth noting that this modification does not affect the performance of `FedPE` and `FedPG` since the update results will be the same after one iteration.

Assumption. We first make the following assumptions:

- A1: For each client's objective function $f_i, \forall i = 1, \ldots, N$ there exists a constant $L_i \geq 0$ such that for any two points U_i, Z_i, the gradient of f_i satisfies the Lipschitz condition:

$$\|\nabla f_i(U_i) - \nabla f_i(Z_i)\|_F \leq L_i \|U_i - Z_i\|_F, \forall i$$

- A2: For all i, the penalty parameter ρ_i is chosen large enough such that the U_i sub-problem (3.20) is strongly convex with modulus $\mu_i(\rho_i)$.

- A3: $\mathcal{L}(x)$ is bounded from below, i.e.,

$$\underline{\mathcal{L}} := \min_X \mathcal{L}(X) > -\infty$$

Remark 1. Assumptions A1 and A3 are standard in analyzing convergence properties for ADMM. Under Assumption A2, as ρ_i increases, the subproblem (3.20) will be eventually strongly convex with respect to U_i. The associated strong convexity modulus $\mu_i(\rho_i)$ is a monotonic increasing function of ρ_i.

To establish a convergence guarantee of `FedPE` and `FedPG`, we aim to show the following. Start with the variables $(\{U_i^k\}, Z^k, \{Y_i^k\})$. After T rounds (i.e., after all clients have participated at least once since iteration k), the new variables $(\{U_i^{k+T}\}, Z^{k+T}, \{Y_i^{k+T}\})$ lead to decrease in the global augmented Lagrangian $\mathcal{L}$. Then we show these variables will converge to a local stationary point of $\mathcal{L}$. Notably, our analysis employs matrix variables, in contrast to other ADMM-based methods using vector variables [151–153, 193]

3.3.1 Bound on the Successive Difference of Dual Variables

We establish an upper bound for the successive difference of the dual variables Y_i in terms of the primal variables U_i.

Lemma 1. *Suppose Assumptions A1 hold, then we have*

$$\left\|Y_i^{k+1} - Y_i^k\right\|_F^2 \leq L_i^2 \left\|U_i^{k+1} - U_i^k\right\|_F^2, \forall i, \forall k \tag{3.22}$$

Proof. In the case of $i \notin S^k$, (3.22) is true because both sides of (3.22) are equal to zero. When $i \in S^k$, we have below equation from optimal condition for (3.20).

$$\nabla f_i\left(U_i^{k+1}\right) + Y_i^k + \rho_i(U_i^{k+1} - Z^{k+1}) = 0 \tag{3.23}$$

Substituting (3.23) to the dual variable update (3.21), the following equation is derived.

$$\nabla f_i\left(U_i^{k+1}\right) = -Y_i^{k+1} \tag{3.24}$$

Combining with Assumption $A1$, and noting that for any given i, U_i and Y_i are updated in the same iteration, we obtain for all $i \in S^k/\{0\}$:

$$\begin{aligned}\left\|Y_i^{k+1}-Y_i^k\right\|_F &= \left\|\nabla f_i\left(U_i^{k+1}\right)-\nabla f_i\left(U_i^{k(i)}\right)\right\|_F \\ &= \left\|\nabla f_i\left(U_i^{k+1}\right)-\nabla f_i\left(U_i^{k(i)}\right)\right\|_F \\ &\leq L_i\left\|U_i^{k+1}-U_i^{k(i)}\right\|_F = L_i\left\|U_i^{k+1}-U_i^k\right\|_F\end{aligned}$$

3.3.2 Bound for the Augmented Lagrangian

Next, we show that the augmented Lagrangian (3.18) can be decreased sufficiently and bounded below.

Lemma 2. *Suppose Assumption A1, A2 and A3 hold, then the augmented Lagrangian is bounded as follows.*

$$\begin{aligned}&\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^{k+1}\right\}\right)-\mathcal{L}\left(\left\{U_i^k\right\}, Z^k, \left\{Y_i^k\right\}\right) \\ \leq &\sum_{i\in S^k}\left(\frac{L_i^2}{\rho_i}-\frac{\mu_i}{2}\right)\left\|U_i^{k+1}-U_i^k\right\|_F^2-\frac{\gamma}{2}\left\|Z^{k+1}-Z^k\right\|^2\end{aligned}$$

Proof. For each client $i \in S^k$, we can split the successive difference of the augmented Lagrangian by following.

$$\begin{aligned}&\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^{k+1}\right\}\right)-\mathcal{L}\left(\left\{U_i^k\right\}, Z^k, \left\{Y_i^k\right\}\right) \\ &=\underbrace{\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^{k+1}\right\}\right)-\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^k\right\}\right)}_{A} \\ &+\underbrace{\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^k\right\}\right)-\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^k, \left\{Y_i^k\right\}\right)}_{B} \\ &+\underbrace{\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^k, \left\{Y_i^k\right\}\right)-\mathcal{L}\left(\left\{U_i^k\right\}, Z^k, \left\{Y_i^k\right\}\right)}_{C}\end{aligned} \tag{3.25}$$

Bounding Term A. The first term in (3.25) can be bounded as follows.

$$\begin{aligned}&\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^{k+1}\right\}\right)-\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^k\right\}\right) \\ &=\sum_{i=1}^N\left\langle Y_i^{k+1}-Y_i^k, U_i^{k+1}-Z^{k+1}\right\rangle_F \\ &\overset{(a)}{=}\sum_{i\in S^k}\frac{1}{\rho_i}\left\|Y_i^{k+1}-Y_i^k\right\|_F^2\end{aligned} \tag{3.26}$$

where (a) is achieved by using (3.21).

Bounding Term B. The second term in (3.25) can be bounded as follows.

$$\begin{aligned}&\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, Y^k\right)-\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^k, \left\{Y_i^k\right\}\right) \\ &\overset{(A2)}{\leq}\sum_{i=1}^N\left(\left\langle\nabla_{U_i}\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, Y^k\right), U_i^{k+1}-U_i^k\right\rangle-\frac{\mu_i}{2}\left\|U_i^{k+1}-U_i^k\right\|_F^2\right) \\ &\overset{(b)}{=}-\sum_{i\in S^+}\frac{\mu_i}{2}\left\|U_i^{k+1}-U_i^k\right\|_F^2.\end{aligned} \tag{3.27}$$

where (b) is trivially true in the case of $i \notin S^k$ as $U_i^{k+1} = U_i^k$. In the case of $i \in S^k$, the $\nabla_{U_i} \mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \{Y_i^k\}\right) = 0$ because of optimal condition of (3.20).

Bounding Term C. The third term in (3.25) can be bounded as follows.

$$\begin{aligned}
&\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^k, \{Y_i^k\}\right) - \mathcal{L}\left(\left\{U_i^k\right\}, Z^k, \{Y_i^k\}\right) \\
&\overset{(c)}{\leq} \sum_{i=1}^{N} \left(\left\langle \nabla_Z \mathcal{L}\left(\left\{U_i^k\right\}, Z^{k+1}, \{Y_i^k\}\right), Z^{k+1} - Z^k \right\rangle - \frac{\gamma}{2} \left\| Z^{k+1} - Z^k \right\|_F^2 \right) \\
&\overset{(d)}{=} \sum_{i \in S^k} -\frac{\gamma}{2} \left\| Z^{k+1} - Z^k \right\|_F^2 .
\end{aligned} \tag{3.28}$$

where in (c) we used the fact that $\mathcal{L}\left(\{U_i\}, Z, \{Y_i\}\right)$ is strongly convex w.r.t. Z.

For (d), it is trivially true if $i \notin S^k$ as $U_i^{k+1} = U_i^k$. When $i \in S^k$, the $\nabla_Z \mathcal{L}\left(\left\{U_i^k\right\}, Z^{k+1}, \{Y_i^k\}\right) = 0$ because of the optimal condition of (3.20).

Sufficient decrease. Combining the inequalities Equation (3.26), Equation (3.27) and Equation (3.28) together, we have

$$\begin{aligned}
&A + B + C \\
&= \mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^{k+1}\right\}\right) - \mathcal{L}\left(\left\{U_i^k\right\}, Z^k, \left\{Y_i^k\right\}\right) \\
&\leq \sum_{i \in S^k} -\frac{\gamma}{2} \left\| Z^{k+1} - Z^k \right\|_F^2 + \left(\frac{L_i^2}{\rho_i} - \frac{\mu_i}{2} \right) \left\| U_i^{k+1} - U_i^k \right\|_F^2
\end{aligned} \tag{3.29}$$

This implies that the value of the augmented Lagrangian will always decrease if the following condition is satisfied:

$$\rho_i \mu_i \geq 2L_i^2 \tag{3.30}$$

Lower bound for the augmented Lagrangian. We continue showing that $\mathcal{L}\left(\left\{U_i^k\right\}, Z^k, \{Y_i^k\}\right)$ convergence, i.e., $\mathcal{L}$ is bounded below.

$$\begin{aligned}
\mathcal{L}\left(\left\{U_i^{k+1}\right\}, Z^{k+1}, \left\{Y_i^{k+1}\right\}\right) &= \sum_{i=1}^{N} \left(f_i\left(U_i^{k+1}\right) + \left\langle Y_i^{k+1}, U_i^{k+1} - Z^{k+1} \right\rangle + \frac{\rho_i}{2} \left\| U_i^{k+1} - Z^{k+1} \right\|_F^2 \right) \\
&\overset{(e)}{=} \sum_{i=1}^{N} \left(f_i\left(U_i^{k+1}\right) + \left\langle \nabla f\left(U_i^{k+1}\right), Z^{k+1} - U_i^{k+1} \right\rangle + \frac{\rho_i}{2} \left\| U_i^{k+1} - Z^{k+1} \right\|_F^2 \right) \\
&\overset{(f)}{\geq} \sum_{i=1}^{N} f_i\left(Z^{k+1}\right) = \underline{\mathcal{L}}\left(Z^{k+1}\right)
\end{aligned}$$

where (e) can be explained as following:

(e.1) If $i \in S^k$, based on (3.24):

$$\left\langle Y_i^{k+1}, U^{k+1} - Z^{k+1} \right\rangle = \left\langle \nabla_f\left(U_i^{k+1}\right), Z^{k+1} - U^{k+1} \right\rangle$$

(e.2) If $i \notin S^k$, $Y_i^{k+1} = Y_i^k = Y_i^{k(i)}$, $U_i^{k+1} = U_i^k = U^{k(i)}$

$$\begin{aligned}
\left\langle Y_i^{k+1}, U_i^{k+1} - Z^{k+1} \right\rangle &= \left\langle \nabla f\left(U_i^{k(i)}\right), Z^{k+1} - U_i^{k(i)} \right\rangle \\
&= \left\langle \nabla f\left(U_i^{k+1}\right), Z^{k+1} - U_i^{k+1} \right\rangle
\end{aligned}$$

From combining (e.1) and (e.2), we deduce that (e) holds true. Consequently, due to the Lipschitz continuity of the gradient of f_i, (f) is also true.

Remark 2. Lemma 2 shows that the augmented Lagrangian decreases by a sufficient amount after each iteration. In practice, however, the penalty parameter ρ is difficult to estimate. Hence, for our experiments, we fine-tune ρ from a predefined range of values to select the best one.

3.3.3 Convergence of the augmented Lagrangian

Here, we combine Lemma (1) and (2) to establish the following theorem.

Theorem 1. *Suppose the following are true:*

- *After T iterations, all clients have participated in training at least once.*
- *The global loss function $\mathcal{L}(Z^k, \{U_i^k\}, \{Y_i^k\})$ is bounded below by a finite quantity $\underline{\mathcal{L}}$.*
- *The hyperparameter ρ_i for the augmented Lagrangian is chosen such that $\rho_i \mu_i \geq 2L_i^2$.*

Then the augmented Lagrangian $\mathcal{L}$ will monotonically decrease and is convergent to a quantity of at least $\underline{\mathcal{L}}$. Further, for all $i = 1, \ldots, N$ we have $\lim_{k\to\infty} \|Z^{k+T} - U_i^{k+T}\| = 0$.

Proof. Theorem 1 implies that the augmented Lagrangian decreases by a non-negative amount after every T global iterations. Given the fact that every client will be updated at least once in the interval $[k, k+T]$, we conclude that the limit $\lim_{k\to\infty} \mathcal{L}(Z^{k+T}, \{U_i^{k+T}\}, \{Y_i^{k+T}\})$ exists and is at least $\underline{\mathcal{L}}$.

Now we prove the second statement. From Equation (3.29) and the fact that $\mathcal{L}$ converges, we conclude that as $k \to \infty$,

$$\|Z^{k+T} - Z^k\| \to 0, \quad \|U_i^{k+T} - U_i^k\| \to 0.$$

Combining this with Lemma 1, we have $\|Y_i^{k+1} - Y_i^k\| \to 0$. Based on the definition of Y_i^{t+1}, this implies that

$$\|U_i^{k+1} - Z^{k+1}\| \to 0$$

Remark 3. Theorem 1 establishes that the sequence of primal and dual variables updated after each T global iterations of Algorithm 1 and Algorithm 5 converges. We further have $\|U_i^{k+1} - Z^{k+1}\| \to 0$, i.e., the consensus constraint is satisfied.

3.3.4 Convergence to stationary point

In the below theorem we present another guarantee that the limit of the sequence $(Z^k, \{U_i^k\}, \{Y_i^k\})$ is the a stationary solution for Problem Equation (3.5).

Theorem 2. *Suppose the assumptions in Theorem 1 hold. Then the limit point $(Z^*, \{U_i^*\}, \{Y_i^*\})$ of the sequence $(Z^k, \{U_i^k\}, \{Y_i^k\})$ is a stationary solution to Problem Equation (3.5). That is, for all i,*

$$\nabla f_i(Z^*) + Y_i^* = 0 \text{ and } Z^* = U_i^*.$$

Proof. See the proof to [193, Theorem 2.4]. Here we make some remarks. First, because `FedPE` and `FedPG` allow for client sampling, the proof relies on an assumption that all clients will participate after some fixed number of rounds (in particular, T rounds). Second, since local objectives are smooth (possibly non-convex), the primal variables U_i are updated using first-order methods like SGD or projected GD. Finally, the proof aims to show that after every T rounds, we obtain a sufficient decrease in the augmented Lagrangian; coupled with the fact that the augmented Lagrangian is bounded below, the sequence of variables converges to a stationary point of $\mathcal{L}$.

3.3.5 Convergence rate of the augmented Lagrangian

We first define gradient of the Augmented Lagrangian as:

$$\hat{\nabla}\mathcal{L}\left(\{U_i\}, Z, Y\right) := \begin{bmatrix} \nabla_{U_1}\mathcal{L}\left(\{U_i\}, Z, Y\right) \\ \nabla_{U_2}\mathcal{L}\left(\{U_i\}, Z, Y\right) \\ \vdots \\ \nabla_{U_N}\mathcal{L}\left(\{U_i\}, Z, Y\right) \end{bmatrix}$$

We aim to use the below quantity to establish the progress of the proposed methods:

$$P(\{U_i^k\}, Z^k, Y_i^k) := \sum_{i=1}^{N} ||U_i^k - Z^k||^2 + ||\hat{\nabla}\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k)||^2$$

It can be verified that if $P(\{U_i^k\}, Z^k, Y_i^k) \to 0$, then a stationary solution is obtained, which U_i converges to Z and gradient for Augmented Lagrangian of each client i is equal to zero.

Theorem 3. *Suppose that Assumptions A.1, A.2, A.3 and Theorem 1 are satisfied. Let $T(\epsilon)$ denote an iteration index in which the following inequality is achieved:*

$$T(\epsilon) := \min(\{k | P(\{U_i^k\}, Z^k, Y_i^k)\} \leq \epsilon, k \geq 0)$$

for some $\epsilon > 0$. Then there exists some constant $\beta > 0$ such that

$$T(\epsilon) \leq \frac{\beta(\mathcal{L}(\{U_i^1\}, Z^1, Y_i^1) - \underline{\mathcal{L}})}{\epsilon}$$

where $\underline{\mathcal{L}}$ is defined by Assumption A.3.

Proof. Based on the optimal condition of U_i (3.7):

$$\nabla f_i(U_i^{k+1}) + Y_i^k + \rho_i(U_i^{k+1} - Z^{k+1}) = 0 \tag{3.31}$$

By definition of $\mathcal{L}$, we can have:

$$\begin{aligned} &||\nabla_{U_i}\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k)|| = ||\nabla f_i(U_i^k) + Y_i^k + \rho_i(U_i^k - Z^k)|| \\ &\overset{(g)}{=} ||\nabla f_i(U_i^k) + Y_i^k + \rho_i(U_i^k - Z^k) - (\nabla f_i(U_i^{k+1}) + Y_i^k + \rho_i(U_i^{k+1} - Z^{k+1}))|| \\ &\leq ||\nabla f_i(U_i^k) - \nabla f_i(U_i^{k+1})|| + \rho_i||U_i^{k+1} - U_i^k|| + \rho_i||Z^{k+1} - Z^k|| \\ &\overset{(h)}{\leq} L_i||U_i^{k+1} - U_i^k|| + \rho_i||U_i^{k+1} - U_i^k|| + \rho_i||Z^{k+1} - Z^k|| \\ &= (L_i + \rho_i)||U_i^{k+1} - U_i^k|| + \rho_i||Z^{k+1} - Z^k|| \end{aligned} \tag{3.32}$$

Where (g) is obtained by optimal condition of U_i in (3.31). To derive (h), the first term is bounded by assumption $A.1$ and the second term is bounded by Lemma 1. Based on (3.32), we have the following:

$$||\hat{\nabla}\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k)|| \leq \sum_{i=1}^{N}(L_i + \rho_i)||U_i^{k+1} - U_i^k|| + \sum_{i=1}^{N}\rho_i||Z^{k+1} - Z^k|| \tag{3.33}$$

By taking $\sigma_1 = \max\{\sum_{i=1}^{N}\rho_i, L_1 + \rho_1, ..., L_N + \rho_N\}$, it is apparent that $\sigma_1 > 0$. We can have following:

$$||\hat{\nabla}\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k)|| \leq \sigma_1(||Z^{k+1} - Z^k|| + \sum_{i=1}^{N}||U_i^{k+1} - U_i^k||) \tag{3.34}$$

From (4.11), we have

$$\begin{aligned}\sum_{i=1}^{N}||U_i^{k+1} - Z^{k+1}|| &= \sum_{i=1}^{N}\frac{1}{\rho_i}||Y_i^{k+1} - Y_i^k|| \\ &\leq \sum_{i=1}^{N}\frac{L_i}{\rho_i}||U_i^{k+1} - U_i^k||\end{aligned} \tag{3.35}$$

(3.34) and (3.35) imply for some $\sigma_3 \geq 0$:

$$\begin{aligned}&\sum_{i=1}^{N}||U_i^k - Z^k||^2 + ||\hat{\nabla}\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k)||^2 \\ &\leq \sigma_3(||Z^{k+1} - Z^k||^2 + \sum_{i=1}^{N}||U_i^{k+1} - U_i^k||^2)\end{aligned} \tag{3.36}$$

It is noteworthy that we have changed from $||U_i^{k+1} - Z^{k+1}||^2$ to $||U_i^k - Z^k||^2$. This is because we have proved that $||U_i^{k+1} - Z^{k+1}|| \to 0$, which means the gap between these two terms $||U_i^{k+1} - Z^{k+1}||^2$ and $||U_i^k - Z^k||^2$ are finite. Therefore, we always can find a finite number $\sigma_3 \geq 0$ to satisfy $||U_i^k - Z^k||^2 \leq \sum_{i=1}^{N}\sigma_3||U_i^{k+1} - U_i^k||^2$. From lemma 2, there exists a constant $\sigma_2 = min\{\frac{L_i^2}{\rho_i} - \frac{\mu_i}{2}, \mu\}_{i=1}^{N}$ such that:

$$\begin{aligned}&\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k) - \mathcal{L}(\{U_i^{k+1}\}, Z^{k+1}, Y_i^{k+1}) \\ &\geq \sigma_2(||Z^{k+1} - Z^k||^2 + \sum_{i=1}^{N}||U_i^{k+1} - U_i^k||^2)\end{aligned} \tag{3.37}$$

Combining (3.36) and (3.37) we have:

$$\begin{aligned}&\sum_{i=1}^{N}||U_i^k - Z^k||^2 + ||\hat{\nabla}\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k)||^2 \\ &\leq \frac{\sigma_3}{\sigma_2}(\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k) - \mathcal{L}(\{U_i^{k+1}\}, Z^{k+1}, Y_i^{k+1}))\end{aligned} \tag{3.38}$$

Summing both sides of the above inequality over $k = 1, ..., T$. We have the following:

$$\begin{aligned}
&\sum_{k=1}^{T}(\sum_{i=1}^{N}||U_i^k - Z^k||^2 + ||\hat{\nabla}\mathcal{L}(\{U_i^k\}, Z^k, Y_i^k)||^2) \\
&\leq \frac{\sigma_3}{\sigma_2}(\mathcal{L}(\{U_i^1\}, Z^1, Y_i^1) - \mathcal{L}(\{U_i^{T+1}\}, Z^{T+1}, Y_i^{T+1})) \\
&\leq \frac{\sigma_3}{\sigma_2}(\mathcal{L}(\{U_i^1\}, Z^1, Y_i^1) - \underline{\mathcal{L}})
\end{aligned} \tag{3.39}$$

The last inequality uses the fact that $L(\{U_i^{T+1}\}, Z^{T+1}, Y_i^{T+1})$ is decreased and lower bounded by $\underline{\mathcal{L}}$ by $A.3$. By putting $T(\epsilon) := \min\{k|P(\{U_i^k\}, Z^k, Y_i^k) \leq \epsilon, k \geq 0\}$, we can rewrite (3.39) as follows.

$$T(\epsilon)\epsilon \leq \frac{\sigma_3}{\sigma_2}(\mathcal{L}(\{U_i^1\}, Z^1, Y_i^1) - \underline{\mathcal{L}}) \tag{3.40}$$

or

$$T(\epsilon) \leq \frac{\sigma_3}{\sigma_2}\frac{(\mathcal{L}(\{U_i^1\}, Z^1, Y_i^1) - \underline{\mathcal{L}})}{\epsilon} \tag{3.41}$$

This means that for a given ϵ, the proposed algorithm converges with rate of $O(\frac{1}{T(\epsilon)})$. We note that the use of $\sum_i^N$ in the above proof does not mean that we require participation from all clients. This is used to find an upper bound. Let $\beta = \frac{\sigma_3}{\sigma_2}$, theorem 3 is achieved.

3.3.6 Convergence rates of the proposed methods

Theorem 4. *Let $T(\epsilon)$ be defined as in Theorem 3. Suppose the local update (3.7) in FedPE is performed using gradient descent with C iterations. Then, the convergence rates for FedPE and FedPG are $O\left(\frac{1}{CT(\epsilon)}\right)$ and $O\left(\frac{1}{C^2T(\epsilon)}\right)$, respectively.*

Proof. Based on Theorem 3, the general convergence rates of $O(\frac{1}{T(\epsilon)})$ are proved for both `FedPE` and `FedPG`. While the learning rate of solving problem (4.16) can be accelerated by using gradient descent on Grassmann manifold with convergence rate of $O(1/C^2)$ [194, Theorem 1], convergence rate for gradient descent on Euclidean space to solve (3.7) is $O(1/C)$ for strongly convex functions [195]. Therefore, the convergence rates of `FedPE` and `FedPG` can be estimated as $O(\frac{1}{CT(\epsilon)})$ and $O(\frac{1}{C^2T(\epsilon)})$ respectively.

3.4 Experimental Results

In this section, we first present the datasets, model settings and measures for performance assessment. Subsequently, to illustrate the efficacy of the proposed methods, we contrast `FedPE` and `FedPG` with various benchmark strategies within the context of an IDS deployment in IoT networks.

3.4.1 Experimental settings

In our research, all experiments are conducted in a coding environment powered by an AMD Ryzen 3970X Processor with 64 cores and 256GB of RAM. Additionally, four NVIDIA GeForce RTX 3090 GPUs are employed to accelerate the

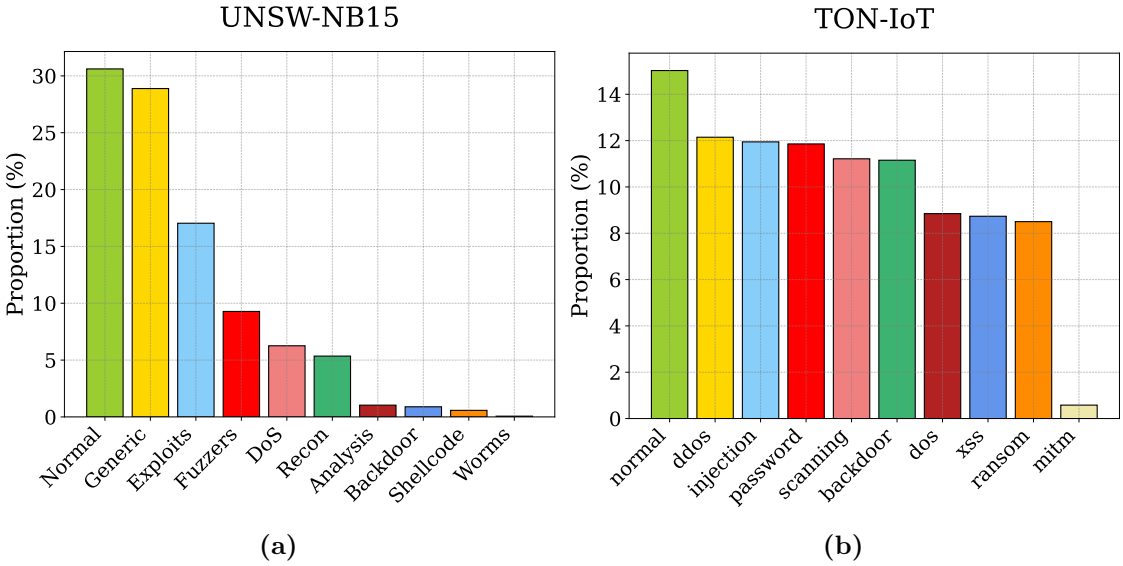

Figure 3.3: Class distribution of test sets for UNSW-NB15 and TON-IoT

training process. For software, we rely on Python 3 as our primary language, supplemented by advanced machine learning and optimization frameworks such as PyTorch [196] and Scikit-Learn [197]. Our code can be found at: https://github.com/dual-grp/FedPCA_AnomalyDetection.

3.4.1.1 Datasets and Metrics

In this study, we employ the UNSW-NB15 [186] and TON-IoT Network [187] datasets. The former, a benchmark in network intrusion detection system (NIDS) research, encompasses a myriad of network traffic scenarios, with various attack types interlaced with normal activities commonly found in real-world network (internet) environments. The latter, TON-IoT, includes heterogeneous data sourced from Telemetry datasets of IoT and IIoT sensors, amassed through parallel processing to capture a spectrum of normal and cyber-attack events. Following preprocessing, from the UNSW-NB15 dataset, we utilized a training set of 56,000 normal samples and a test set comprising 20,000 normal and 45,332 abnormal samples, all with 39 numerical features. From the TON-IoT dataset, our training set consists of 114,956 normal samples, while the test set includes 10,000 normal samples and 56,557 abnormal samples, all with 49 numerical features. The class distribution of the test sets for both datasets is illustrated in Fig.3.3a and Fig.3.3b respectively. These carefully selected and prepared datasets serve as a solid foundation for the evaluation of our proposed approach. Each data sample is considered an independent record comprising feature vectors and corresponding labels, without modeling temporal dependencies or sequential correlations inherent in time-series or network flow data.

In the setting of a host-based ML-IDS deployment, each IoT device functions as a localized ML-IDS, using its unique data to develop an anomaly detection model. This approach reflects the inherent diversity of traffic distribution across disparate clients, and to mimic this realistically, the comprehensive training set is equally subdivided into 100 individual client datasets, each representing non-identically distributed (non-i.i.d.) local data. This subdivision is achieved by grouping records based on the number of

connections that contain the same service (*ct_srv_src*) for UNSW-NB15 and the range of source ports (*srcports*) for TON-IoT.

3.4.1.2 Baselines

In order to highlight the efficacy of `FedPE` and `FedPG` in IoT network anomaly detection, we conduct a comprehensive comparative study against a variety of unsupervised techniques including:

a) *Self-learning PCA*: a stand-alone method where each client individually learns its PCA matrix using only its local dataset. By identifying the principal patterns within this dataset, anomalies are detected when there's a substantial discrepancy between the original data and its PCA-reconstructed version.

b) *AutoEncoder (AE)*: an unsupervised learning method constructed of an encoder and a decoder in the form of neural networks. AE facilitates anomaly detection by compressing input data during encoding and then attempting to reconstruct it. Anomalies are identified when there's a notable difference between the input data and its reconstructed version, suggesting the input doesn't conform to typical patterns within the dataset [175].

c) *Bidirectional Generative Adversarial Network (BiGAN)*: a generative model composed of a generator, a discriminator and an encoder. During BiGAN's adversarial training, the encoder maps real data into latent representations. The generator then tries to recreate data from these representations, aiming to closely match the original input. Concurrently, the discriminator distinguishes between pairs of real data and their latent encodings, and the output from the generator with its corresponding latent representations. When trained solely on normal data, BiGAN can detect anomalies by noting high reconstruction errors, which indicate deviations from the learned representations of normal data [176, 177].

We subsequently employ the de-facto FedAvg [14] to ensure consistency in the training of AE and BiGAN within federated settings. In line with this, we establish FedAE and FedBiGAN – which utilize two-layer neural networks for their components, and FedAE-2 and FedBiGAN-2 – which incorporate four-layer neural networks for more complex non-linear structures. For Self-learning PCA, we train a distinct PCA matrix tailored for each client's dataset and then assess the average performance across all clients on detecting anomalies on the test set. The selection of these baseline models, widely acclaimed in the field and each offering distinct methodologies for anomaly detection, ensures a robust evaluation of our methods against diverse unsupervised paradigms.

3.4.1.3 Performance Metrics

In this work, we are primarily concerned with identifying anomalies in network traffic. Hence, the main metrics employed are:

1. **Accuracy**: calculates the ratio of correctly identified both normal and anomalies to the total instances in the dataset, offering a general perspective on the model performance.

2. **Precision**: represents the ratio of true anomalies to all detections labeled as anomalies, assessing the exactness of the identified anomalies.

3. **Recall**: quantifies the proportion of actual anomalies that are correctly flagged by the detection system, assessing the ability of a model to capture all potential threats.

4. **F1-Score**: calculates the harmonic mean of precision and recall, underscoring the balance between the detection of true anomalies and the risk of false alarms.

5. **False Negative Rate (FNR)**: Represents the fraction of actual anomalies that are incorrectly flagged as normal.

6. **Area Under the Receiver Operating Characteristic curve (AUC-ROC)**: A graphical representation plotting True Positive Rate (TPR) against False Positive Rate (FPR) across varying threshold settings, furnishing a comprehensive performance overview.

7. **Precision-Recall (PR) curves**: Serving as a visual instrument to gauge the relationship between precision and recall, this becomes vital, particularly in imbalanced datasets where anomalies are few.

3.4.1.4 Training Details

To provide a practical test for our proposed methods, we train our federated PCA models on each client's dataset, without inter-client data sharing. The comprehensive testing set, inclusive of unknown attacks, is applied uniformly to all methods. Preceding training, all features undergo normalization through a z-score function, utilizing the standard deviation and mean values calculated from the corresponding training set. The following experiments have hyperparameters fine-tuned via a grid search to ensure optimal test performance. Furthermore, each communication round includes a randomly selected subset of clients, constituting 10% of the total count, replicating real-world scenarios where not all devices might be available for communication simultaneously.

3.4.2 Experimental Results

3.4.2.1 Efficiency of Federated PCA on Grassmann Manifold

In our analysis, we scrutinize the convergence efficacy of our Federated PCA algorithms — `FedPE` (Federated PCA over Euclidean Space) and `FedPG` (Federated PCA over Grassmann Manifold) — as depicted in Fig.3.4, Fig. 3.6 and Fig. 3.5. Notably, for the same number of local communication rounds ($LE = 10$), FedPG displays a significantly faster convergence compared to FedPE. It's worth noting that `FedPE` can achieve quicker convergence by augmenting the value of LE, although this also extends the training duration. These findings underscore `FedPG`'s ability to deliver robust detection performance with decreased training time, an attribute highly sought after in practical IDS development contexts. Fig.3.4a, 3.4b illustrate the precise performance of `FedPG` across various thresholds. In our study, we uniformly set $\rho = 1.0$ across all methods and investigate their detection performance by various metrics.

Table 3.1: UNSW-NB15 Dataset

Category	Training Set		Test Set	
	#Record	Rate(%)	#Record	Rate(%)
Normal	67,343	53.46	9,711	43.08
Generic	45,927	36.46	7,458	33.08
Exploits	11,656	9.25	2,421	10.74
Fuzzers	995	0.79	2,754	12.22
DoS	52	0.04	200	0.89
Reconnaissance	52	0.04	200	0.89
Analysis	52	0.04	200	0.89
Backdoor	52	0.04	200	0.89
Shellcode	52	0.04	200	0.89
Worms	52	0.04	200	0.89
Total	*125,973*	*100*	*22,544*	*100*

Table 3.2: TON-IoT Dataset

Category	Training Set		Test Set	
	#Record	Rate(%)	#Record	Rate(%)
Normal	67,343	53.46	9,711	43.08
DoS	45,927	36.46	7,458	33.08
DDoS	11,656	9.25	2,421	10.74
Scanning	995	0.79	2,754	12.22
Ransomware	52	0.04	200	0.89
Backdoor	52	0.04	200	0.89
Injection	52	0.04	200	0.89
Cross-site Scripting	52	0.04	200	0.89
Password	52	0.04	200	0.89
Man-In-The-Middle	52	0.04	200	0.89
Total	*125,973*	*100*	*22,544*	*100*

The impact of low-rank approximation is presented in Fig. 3.5a and 3.5b. Surprisingly, increasing the rank-k does not necessarily lead to an improvement in the performance of both `FedPG` and `FedPE`, as observed in both datasets. This suggests that our PCA-based anomaly detection algorithms can maintain a low memory footprint and operate efficiently with low computational complexity, making them ideal for resource-constrained environments. Despite `FedPG` and `FedPE` providing better approximations for a given dataset with larger rank-k, they also tend to precisely recover data points lying in high-dimensional space. However, high-dimensional data samples often act as outliers, resembling attacks at times. This creates a dilemma between achieving optimal performance and accurate approximation, as seen in the loss-accuracy contrast presented in Fig. 3.5c, 3.5d, and Fig. 3.5a, 3.5b. Abnormal detection using PCA relies on measuring the distance between data points and principal components, making it challenging to differentiate between high-dimensional data points and malicious data.

To examine the effectiveness of `FedPG` against `FedPE` in terms of computational complexity, we investigate the training time of both algorithms on the same machine.

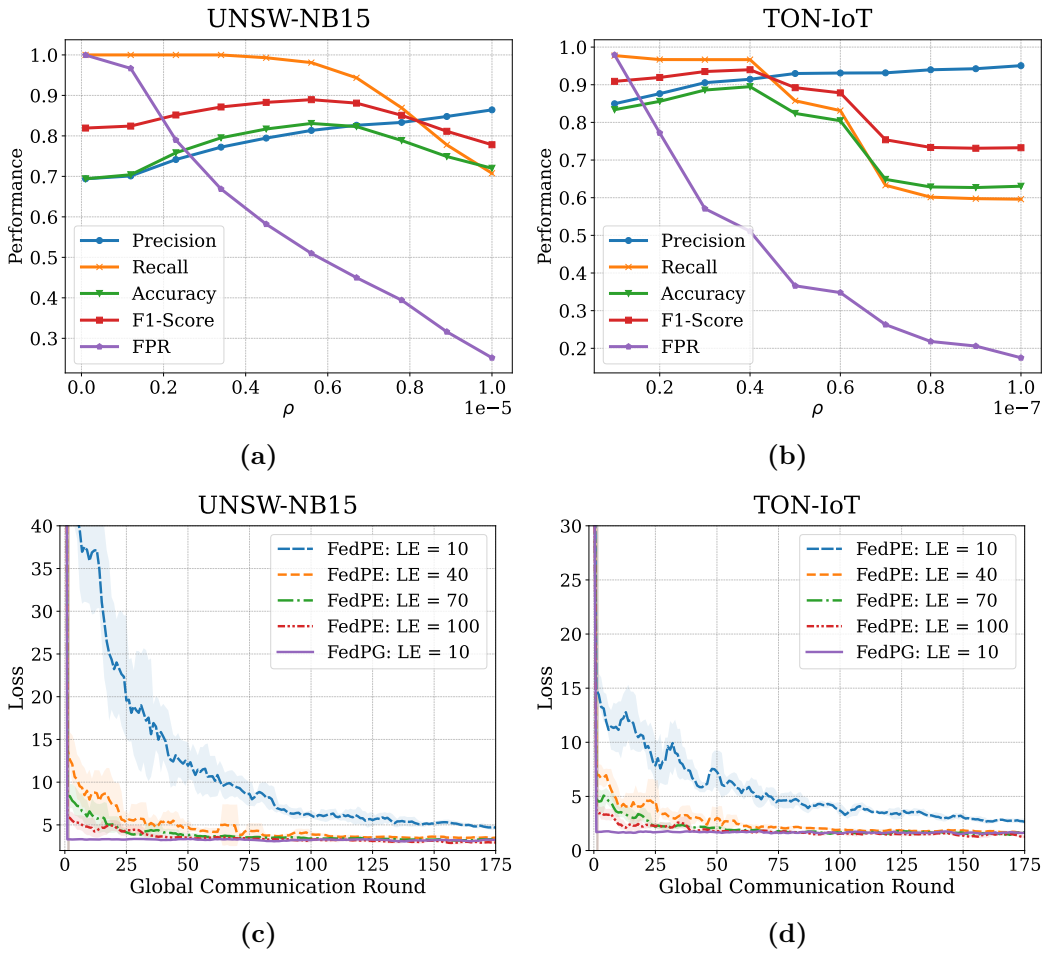

Figure 3.4: (a,b) Effects of ρ on the performance of `FedPG` and (c,d) Training loss over global iterations

While Fig. 3.6a and 3.6c demonstrate that `FedPE` surpasses `FedPG` in terms of the total number of communication rounds, our investigation reveals that `FedPG` is more effective in managing computational costs, as depicted in Fig. 3.6b and 3.6d. It is worth noting that `FedPG` sacrifices a small amount of computational resources in exchange for significantly higher performance.

3.4.2.2 Effects of number clients on performance

We evaluate the performance of `FedPE` and `FedPG` on different scales of IoT systems by varying the number of clients. The results illustrated in Fig. 3.7 reveal a noticeable trade-off between training time and accuracy for these algorithms.

For both datasets, `FedPG` consistently demonstrated a significantly shorter training time as compared to `FedPE`, regardless of the number of clients. For example, on the TON-IoT dataset, the training time of `FedPG` was less than a sixth of the training time of `FedPE` for the same number of clients. In contrast, the accuracy of `FedPG` remained relatively constant and independent of the number of clients in both datasets. `FedPE`, on the other hand, demonstrated an improvement in accuracy as the number of clients increased. However, even with this improvement, it was unable to reach the accuracy of `FedPG` on the TON-IoT dataset. For the UNSW-NB15 dataset, both algorithms

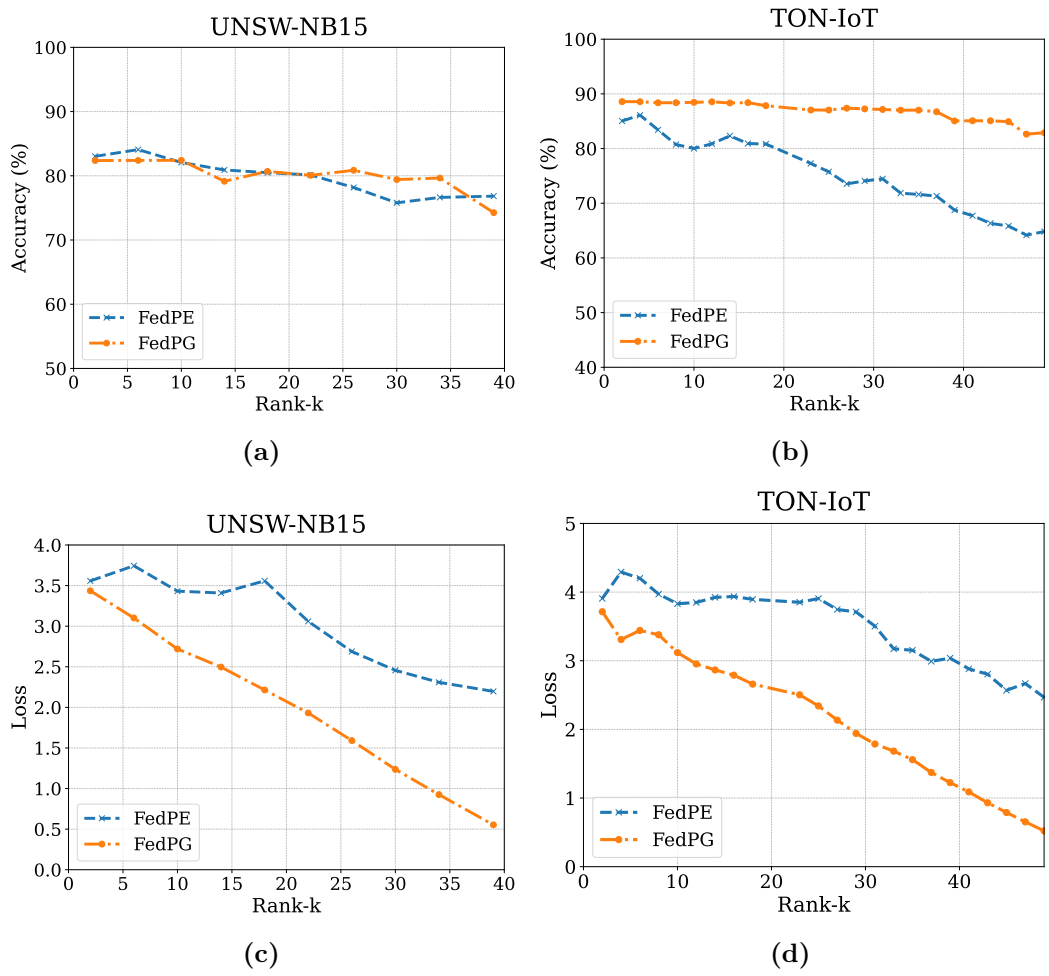

Figure 3.5: Effects of rank-k on the performance of `FedPE` and `FedPG`

performed comparably in terms of accuracy, with `FedPE` even slightly surpassing `FedPG` in fewer clients, but then slightly underperforming in more clients.

These results suggest that `FedPG` could offer substantial time savings over `FedPE` without compromising detection accuracy, particularly for large numbers of clients. However, the appropriate choice of algorithm may also depend on the specific requirements and constraints of the task and system at hand, such as the importance of speed versus detection accuracy and the number of clients in the system.

3.4.2.3 Performance of Network Anomaly Detection Tasks

We evaluate the performance of `FedPG` alongside other baselines in the context of network anomaly detection on both the UNSW-NB15 and TON-IoT datasets. For all methods, we establish optimal thresholds to delineate normal from anomalous behavior using the AUC-ROC curve corresponding to their reconstruction errors. This approach aims to strike a balance by maximizing TPR and minimizing FPR, ensuring robustness and precision in anomaly detection. Such an evidence-driven thresholding strategy underscores the reliability of the reconstruction error as a metric for detecting anomalies in network traffic.

As indicated in Table 3.3, Self-learning PCA, while offering a foundational unsuper-

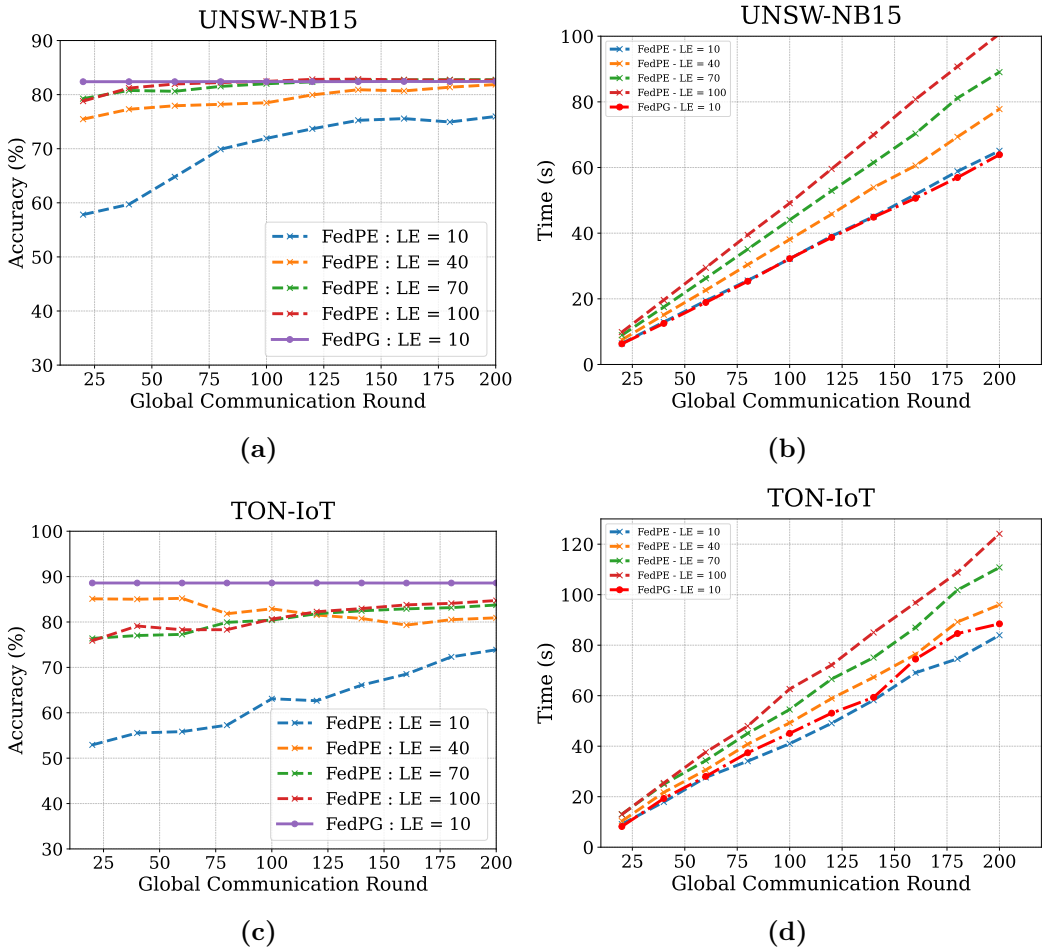

Figure 3.6: Test accuracy and training time over global iterations of `FedPE` and `FedPG` on UNSW-NB15 (a,b) and TON-IoT (c,d)

vised technique, lags behind in performance when compared to other methods. This outcome aligns with expectations, given the diverse and highly heterogeneous nature of clients' data. The performance of `FedPG` and `FedPE` is comparable and slightly better than FedAE and FedBiGAN. Additionally, they demonstrate a competitive performance against more sophisticated non-linear structures such as FedAE-2 and FedBiGAN-2. It should be noted that while `FedPG` may not outperform all other techniques across all metrics, it does exhibit a commendable balance of measures.

On the UNSW-NB15 dataset, `FedPG` not only shows a high accuracy (81.95%) but also balances this with a relatively low false negative rate (6.63%), indicating a robust ability to correctly classify normal behavior while minimizing the chance of missing true anomalies. Similarly, on TON-IoT, `FedPG` holds its own by showcasing the highest recall (94.32%), reflecting its superior ability to correctly identify anomalous events. Moreover, it has the lowest false negative rate (5.68%) among the evaluated methods, which underscores its performance stability. Furthermore, the linear nature of `FedPG` simplifies model interpretability, often leading to more computationally and communication-efficient solutions while still being able to capture significant patterns in the data for network anomaly detection. Meanwhile, although FedAE, FedAE-2, FedBiGAN, and FedBiGAN-2 can potentially capture more complex relationships in

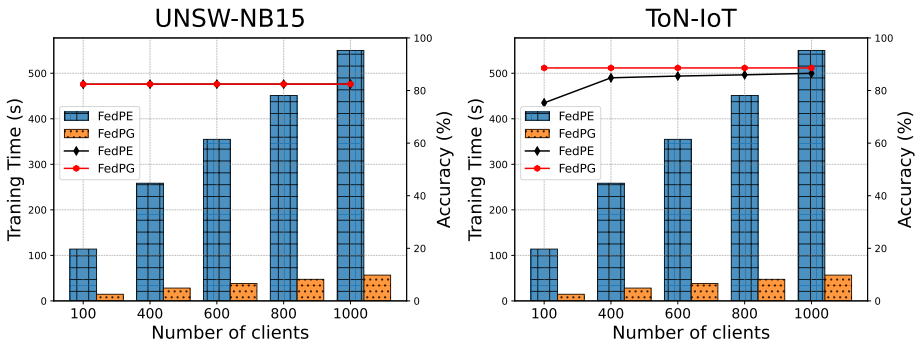

Figure 3.7: Effects of number of clients on training time and performance of `FedPE` and `FedPG`

Table 3.3: Performance of network anomaly detection tasks on UNSW-NB15 and TON-IoT datasets.

Dataset	UNSW-NB15					TON-IoT				
Method	**Acc**	**Pre**	**Recall**	**F1-Score**	**FNR**	**Acc**	**Pre**	**Recall**	**F1-Score**	**FNR**
Self-learning PCA	63.80	83.19	59.94	69.68	40.06	51.81	86.79	51.07	64.30	48.93
FedAE	80.88	80.66	95.27	87.36	4.73	87.28	93.52	91.36	92.43	8.64
FedAE-2	82.80	83.45	93.80	88.32	6.20	89.09	94.68	92.34	93.50	7.66
FedBiGAN	81.21	83.04	91.63	87.12	8.36	88.53	93.98	92.42	93.19	7.57
FedBiGAN-2	81.59	81.98	94.16	87.65	5.84	89.08	95.05	91.93	93.47	8.07
`FedPE` **(ours)**	**82.15**	**81.06**	**95.14**	**87.53**	**6.25**	**87.06**	**89.82**	**94.94**	**92.31**	**6.13**
`FedPG` **(ours)**	**81.95**	**82.82**	**93.36**	**87.77**	**6.63**	**88.94**	**92.79**	**94.32**	**93.55**	**5.68**

the data, they tend to require more computational and memory resources, harder to interpret and may not always result in better performance. This suggests that the balanced performance of our linear method `FedPG` is an attractive alternative for real-world IDS where simplicity, efficiency, and interpretability are crucial.

The superior performance of `FedPG` is further validated by the area under the Receiver Operating Characteristic (AUC-ROC) and Precision-Recall curves, depicted in Fig. 3.8. On both the UNSW-NB15 and TON-IoT datasets, `FedPG` demonstrates higher AUC-ROC values of 0.82, surpassing other baselines (Fig. 3.8a, 3.8b), indicating its enhanced ability to maintain high true positive rates while reducing false positive rates. This performance superiority is corroborated by the Precision-Recall curves, with `FedPG` achieving an Average Precision (AP) of 0.89 and 0.96 on UNSW-NB15 and TON-IoT datasets respectively (Fig. 3.8c, 3.8d), comparable with the competing methods. These results collectively affirm the robustness of `FedPG` in achieving high precision at various levels of recall, underscoring its potential as an effective solution for network anomaly detection.

To delve deeper into the efficacy of the suggested technique, we employed 3D scatter plots to illustrate the transformation of UNSW-NB15 traffic logs as per the techniques described, as depicted in Fig.3.9. For the purpose of plotting, we chose three features[1], following which we reconstructed the original logs from the latent

[1]Network features in Fig.3.9: duration ('dur'), source TCP base sequence number ('stcpb'), source interpacket arrival time ('sinpkt')

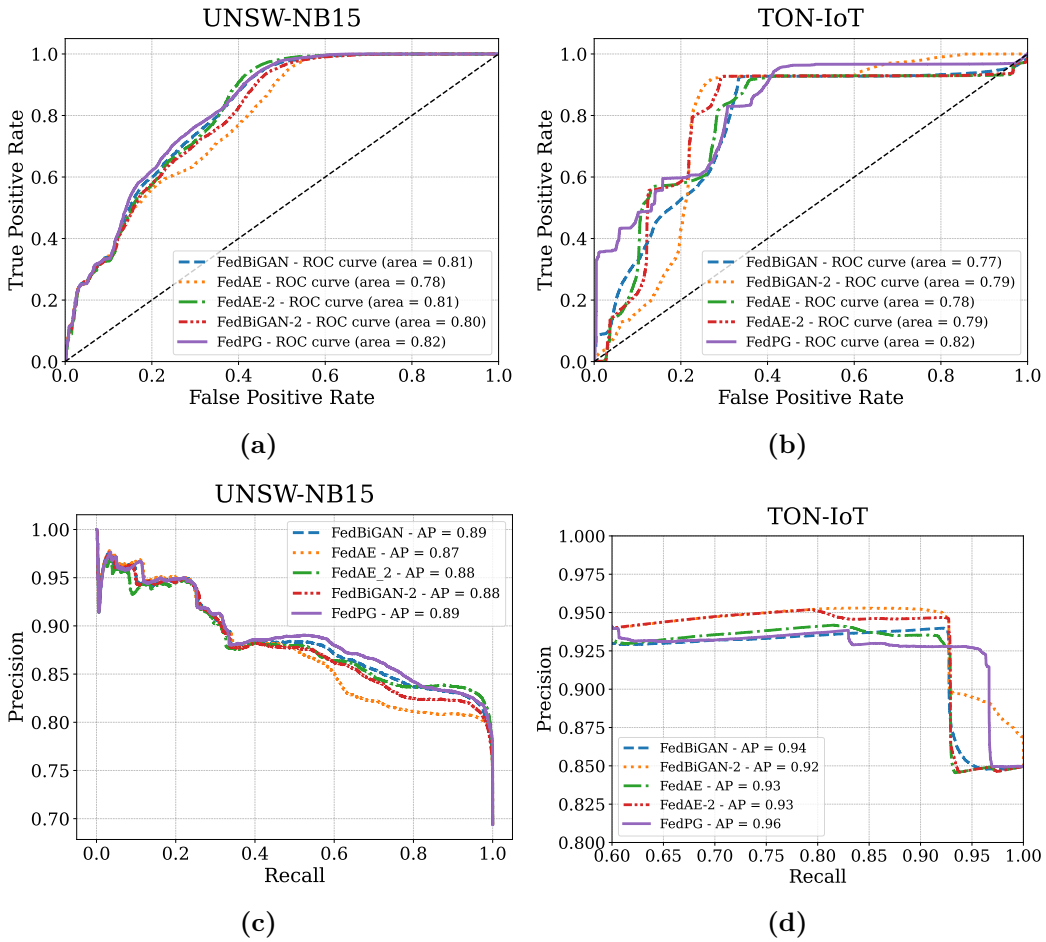

Figure 3.8: (a,b) Receiver Operating Characteristic (ROC) curve and (c,d) Precision-Recall curve

representation discerned by Self-Learning PCA, FedAE, and `FedPG`. It's evident that `FedPG` essentially repositions original logs from an established coordinate framework into an alternate one, proficient in distinguishing between regular logs and anomalies. This affirms that when `FedPG` redevelops the original logs from the primary components, it results in a minimal reconstruction discrepancy for regular data but a substantial error.

3.4.2.4 Communication Efficiency

In the realm of IoT anomaly detection, the transmission size of models during communication rounds plays a pivotal role, directly influencing bandwidth and overall efficiency. Our study reveals a significant communication advantage of `FedPG` when juxtaposed against FedAE, FedBiGAN, FedAE-2, and FedBiGAN-2. Specifically, when considering the UNSW-NB15 datasets, `FedPG` boasts a communication efficiency that's 6.7 and 10 times superior to FedAE and FedBiGAN respectively. While FedAE-2 and FedBiGAN-2 may have a slight edge performance-wise, their demand for bandwidth is significantly higher, necessitating model sizes 10 and 15 times that of `FedPG`. A similar efficiency is observed for the TON-IoT dataset, where `FedPG` outperforms by factors of up to 20 times compared to other methods. These results hold considerable

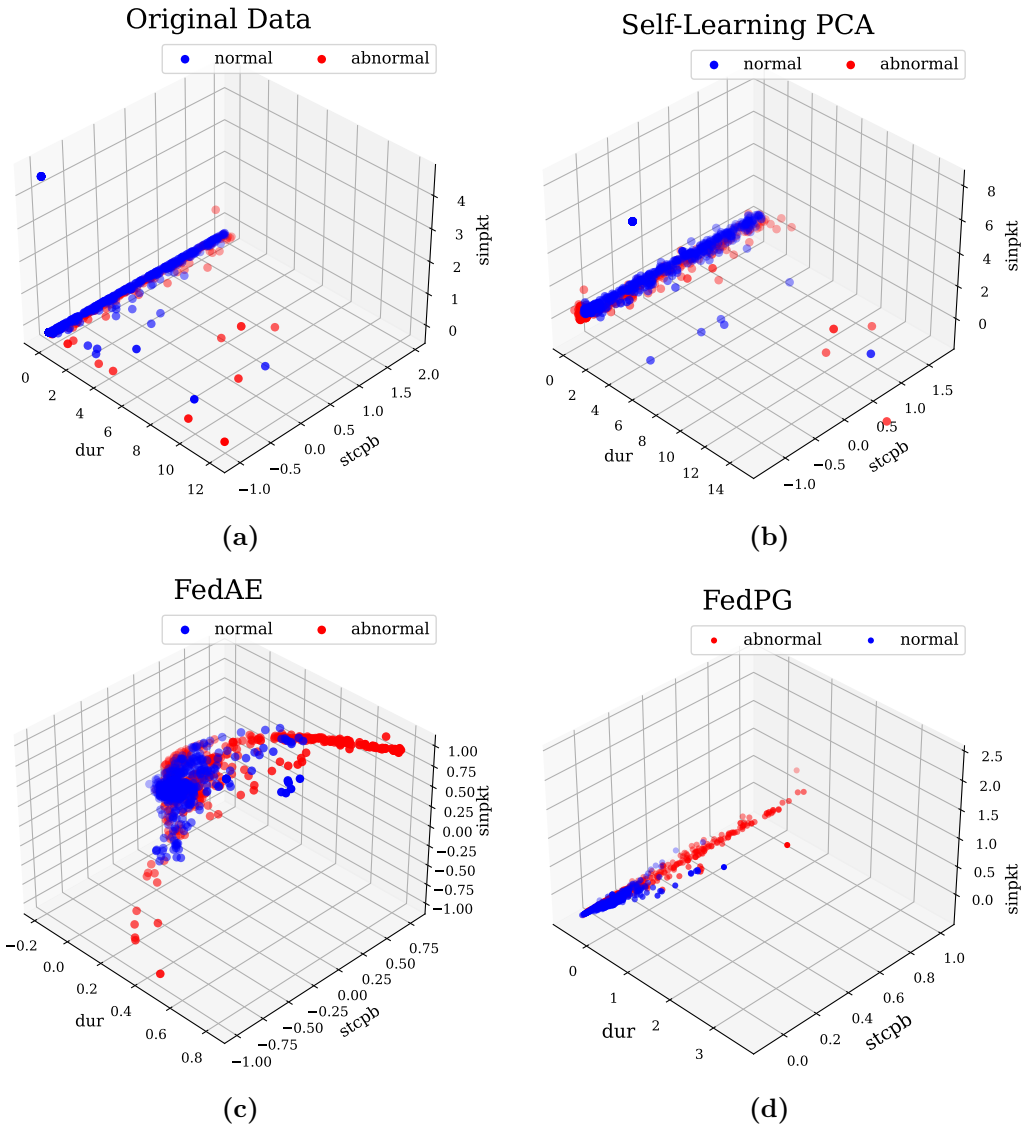

Figure 3.9: Visualization of UNSW-NB15 traffic after reconstruction using different methods

significance in the context of FL, where the reduction of bandwidth utilization is a paramount concern. This evidence underscores the ability of `FedPG` to strike an effective balance between reliable anomaly detection and optimal resource usage, thus bolstering its suitability for real-world NIDS applications.

3.4.2.5 Memory and Time Complexity

We analyze all models through the lens of GPU memory consumption (measured in a percentage) and wall-clock time for inference (measured in a microsecond - μs), a perceptible trade-off emerges between these two metrics. As depicted in Fig. 3.11, `FedPG` and `FedPE` are relatively memory-efficient, consuming around 9.03% and 8.01% of GPU memory during training, respectively. In contrast, the more sophisticated models, FedAE, FedAE-2, FedBiGAN, and FedBiGAN-2, are more resource-demanding, utilizing larger amounts of memory. Despite the smaller memory footprint, `FedPG` and

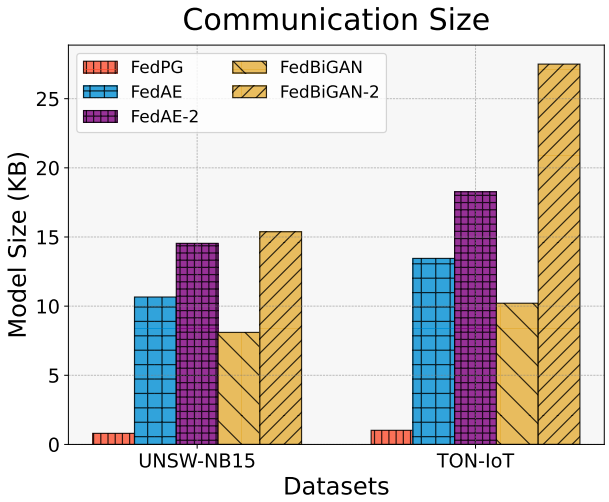

Figure 3.10: The size of model exchanged in each communication round.

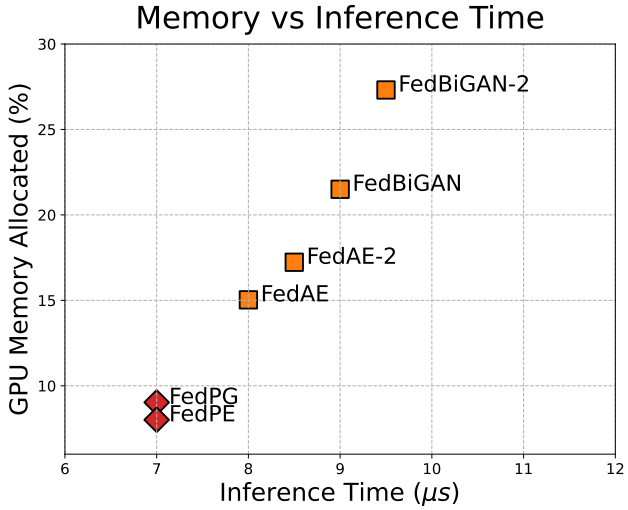

Figure 3.11: The memory consumption and inference time of all methods.

`FedPE` do not compromise on the inference speed, clocking an impressive inference wall-clock time of approximately $7\mu s$. This slightly outperforms the inference time taken by other non-linear methods. Thus, the superior balance struck by `FedPG` and `FedPE` between GPU memory usage and wall-clock time solidifies their efficiency, making them well-suited for resource-restricted environments.

Chapter 4

Communication-Efficient Federated Koopman Learning for Non-stationary IoT Data on Grassmann Manifolds

4.1 `FedKoop`: Communication-Efficient Federated Koopman Learning for Non-stationary IoT Time Series on Grassmann Manifolds

4.1.1 Introduction

Modern IoT systems like traffic control, climate forecasting, and energy management consist of multiple components that emit multivariate time series (MTS) data from local sensors, creating challenges in correlated and decentralized time series analysis [198]. For instance, predicting congestion using long-term data from multiple local traffic sensors necessitates considering the influences of neighboring sites on traffic flow.

The primary goal of MTS prediction task is to capture the complex spatio-temporal structures of high-dimensional data. The prediction model must extract both the temporal dynamics and spatial dependencies, i.e., the inter-relationships across the time series, as correlations often arise from sensor locations. For example, nearby traffic sensors may exhibit similar patterns due to local events and interconnected road systems. However, the inherent non-stationarity of MTS data, caused by distributional shifts where statistical properties change over time, presents significant challenges. Non-stationarity can occurs in each step of MTS, potentially concealing the primary temporal dynamics. Non-stationarity can result from environmental changes, seasonality, trend, or other underlying processes [102]. Additionally, inherent noise and missing values in the data introduce further distributional shifts, complicating real-world time series prediction by obscuring main temporal dependencies. Moreover, training models must consider data sharing constraints due to privacy concerns and the computational limitations of individual devices or subsystems in modern cyber-physical systems.

To overcome these challenges, non-stationary MTS can be represented as a matrix and analyzed using statistical methods. While traditional approaches such as

ARIMA [104] and VAR [199] excel at identifying linear relationships and simple trends, they struggle to capture complex patterns and handle missing values. Alternative approaches, such as ETS [200] and deep learning models (e.g., LSTM [201], DeepAR [202]), can offer more robust solutions. RNNs [105], particularly LSTMs, are notable for learning and retaining information over extended periods, capturing complex, non-linear interactions in time series data. Recently, state-of-the-art architectures based on attention mechanisms, such as Informer [106], Autoformer [107], and Non-Stationary Transformer (NST) [110], have emerged to process long sequences and address non-stationarity in MTS.

While existing methods for MTS achieve promising results, they typically require centralized data for training and overlook the decentralized nature of data, data privacy, and access restrictions. For instance, NST excels at capturing non-stationarity in centralized MTS by estimating mean and variance. However, in decentralized settings, each client has different mean and variance due to the non-independent and non-identical distribution of data, and privacy constraints prevent sharing this information between clients. In recent years, FL has emerged as a solution, allowing machine learning models to be trained with decentralized data while safeguarding against data leakage. In FL, models are trained locally on clients' devices using their own data. Recent FL variants for time series prediction introduce decentralized training on multiple clients, considering spatio-temporal dependencies through a graph structure [198]. However, these approaches overlook the non-stationarity of MTS, highlighting the need for a robust forecasting architecture that captures spatio-temporal relationships while accommodating decentralized non-stationary MTS.

A promising direction for addressing non-stationarity in multivariate time series (MTS) is offered by Koopman operator theory [70, 203, 204], which reformulates the analysis of nonlinear dynamical systems in terms of a linear evolution operator acting on a space of observables. Rather than modeling trajectories directly in the original state space—where nonlinear interactions and time-varying statistics complicate predictive modeling—the Koopman operator enables a lifting of the dynamics into a (possibly infinite-dimensional) function space, within which temporal evolution is governed by a linear operator. This transformation facilitates the application of powerful spectral and modal decomposition techniques to systems that are otherwise intractable in their native coordinates.

This linearization property is particularly advantageous in the context of non-stationary systems, where evolving statistical properties (e.g., mean, variance, and correlation structure) can obscure underlying invariant dynamics [205, 206]. By identifying a Koopman-invariant subspace, it becomes possible to decompose the observed signal into stationary and non-stationary components. Slowly evolving trends, regime shifts, and seasonal variations can be disentangled from stationary residual dynamics, allowing forecasting models to concentrate on stable sub-dynamics while treating non-stationary variations as explicit, separable modes as illustrated in Fig. 4.1.

Such decomposition confers two major benefits for time series forecasting. First, it improves robustness by isolating the stationary structure from distributional drift, enabling more reliable long-term predictions [207]. Second, it enhances generalization in heterogeneous or decentralized settings, particularly within FL frameworks [26]. In

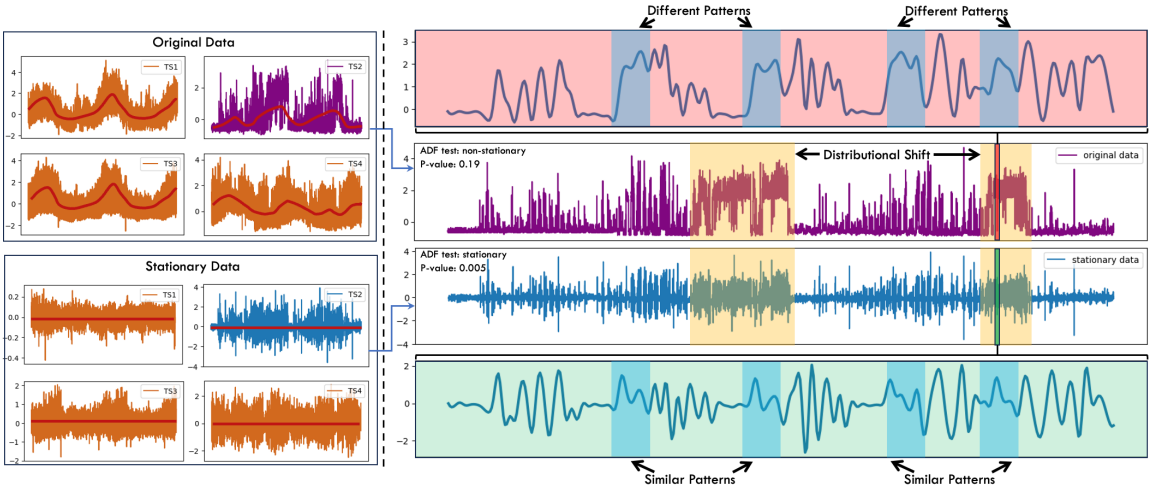

Figure 4.1: Koopman theory removes non-stationary components from non-stationary MTS for electricity dataset [4] in FL, producing stationary data with a zero-centered mean and revealing stationary patterns. The non-stationarity arises from distributional shifts, changing trends, and dominant seasonality, which affect the mean and variance of the time series.

FL, client data is typically non-i.i.d., with local non-stationary characteristics driven by contextual or environmental differences. Direct sharing of raw statistics is precluded by privacy constraints, but local Koopman-based decomposition allows each client to project its data into a common stationary representation. This alignment mitigates cross-client distributional mismatches while preserving privacy, thereby supporting accurate and privacy-preserving MTS forecasting in decentralized environments.

Building upon previous progress, our study introduces **Fed**erated **Koop**man (`FedKoop`) learning, tailored for high-dimensional non-stationary MTS prediction. ***`FedKoop`*** *integrates the decentralized, privacy-preserving attributes of FL with the analytical and spatio-temporal predictive capacities of Koopman theory.* This combination within the FL framework facilitates predictive and privacy-preserving modeling of non-stationary distributed MTS. In this section, the main contributions are summarized as follows:

- We propose `FedKoop`, a privacy-preserving FL framework for predicting non-stationary MTS. By leveraging Koopman theory, `FedKoop` separately learns the non-stationary and stationary components of MTS. This enables handling temporal distributional shifts and capturing hidden spatio-temporal dynamics across multiple clients while maintaining data decentralization.

- We propose a formulation to learn Koopman operators for FL as a consensus optimization problem. To accelerate the convergence rate and reduce computational complexity, we embed the gradient movement on a Grassmann manifold. This approach is suitable for FL clients with resource-limited computing and privacy-conscious setting.

- Experimental results show that our `FedKoop` surpasses other baselines in non-stationary MTS forecasting performance while demanding fewer computational resources and memory. It achieves significantly lower training and prediction times and can handle large numbers of clients.

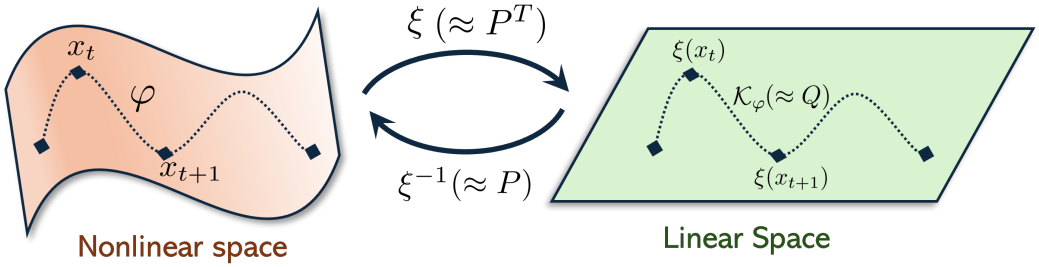

Figure 4.2: Koopman transformation of a nonlinear dynamics $x_{t+1} = \varphi(x_t)$ to a linear dynamics $\xi(x_{t+1}) = K_\varphi \xi(x_t)$, where ξ with its inverse ξ^{-1} and the Koopman operator K_φ are approximated using finite-rank matrices P^T, P, and Q, respectively.

4.1.2 Multivariate Time Series as Dynamics

4.1.2.1 Koopman Theory

Over the past decade, Koopman theory [208] is a prominent framework for the analysis and prediction of dynamical systems, which requires an understanding the effect of how changes in one observation affect others. A dynamical system can be formulated by a time-invariant model $x_{t+1} = \varphi(x_t)$, where x_t denotes the system state at time t and φ the function governing the dynamics from time t to time $t+1$. However, detecting system transitions directly within the state is challenging due to nonlinearity or data disturbances. To address this, Koopman theory proposes a data transformation $\xi(x)$ for the nonlinear system $\varphi(x)$, representing state evolution through the Koopman operator $K_\varphi \xi(x)$:

$$K_\varphi \xi(x_t) = \xi \circ \varphi(x_t) = \xi(\varphi(x_t)) = \xi(x_{t+1})$$

Although K_φ is theoretically infinite-dimensional, it is commonly assumed in practice that a transformation ξ that conjugated with K_φ to yield a finite-dimensional approximation of Koopman operator in the original space of observations x_t:

$$A = \xi^{-1} \circ K_\varphi \circ \xi \tag{4.1}$$

where ξ, ξ^{-1}, and K_φ are illustrated in Fig. 4.2.

4.1.2.2 Optimal Mode Decomposition

Optimal Mode Decomposition (OMD) [209] excels in Koopman approximation by efficiently learning the Koopman matrix through an optimization problem. This enhances system behavior representation with low computational cost. In contrast, other methods, such as KAE and eDMD, are computationally expensive and create a bandwidth burden for large-scale systems. In centralized settings, with a dataset $D = [x_1|x_2|...|x_n] \in \mathbb{R}^{m \times n}$, arranged into matrices (X, Y), where $X = [x_1|x_2|...|x_{n-1}]$ and $Y = [x_2|x_3|...|x_n]$, OMD aims to identify the optimal modes of Koopman operators representing system dynamics by minimizing $||Y - PQP^T X||_F^2$ subject to $P^T P = I$, where $||.||_F$ denotes the Frobenius norm. The constraint ensures that $P \in \mathbb{R}^{m \times r}$ represents the optimal low-dimensional subspace describing the data's dynamics using a linear model. Here, P^T provides a projection $\mathbb{R}^{m \times 1} \mapsto \mathbb{R}^{r \times 1}$, $Q \in \mathbb{R}^{r \times r}$ acts as an operator in r-dimensional subspace, and P offers a lifting $\mathbb{R}^{r \times 1} \mapsto \mathbb{R}^{m \times 1}$, $m \gg r$.

4.1.2.3 Multivariate Time Series as Dynamics

MTS analysis, unlike uni-variate analysis, captures interactions between multiple interdependent variables, reflecting real-world phenomena more precisely. Modeling MTS as a dynamical system facilitates comprehension of the interdependencies, showing how changes in one series affect others. For example, network congestion in one node influences traffic in others, wind speed variations impact temperatures elsewhere, and gasoline price shifts affect other goods' prices. Neglecting these relationships potentially degrades analysis precision, emphasizing the importance of treating MTS as dynamics to integrate temporal and spatial correlations for enhancing model robustness. Hence, we leverage Koopman theory to effectively model non-stationary MTS as dynamics, offering a comprehensive data-driven view of system behavior.

4.1.3 Federated Koopman Methodology

Our objective is to learn the spatio-temporal dependencies of decentralized MTS and predict future values in the presence of non-stationarity. To achieve this, we propose a novel `FedKoop` framework, consisting of two main components. The first component, **Fed**erated **K**oopman learning on **G**rassmann manifolds (`FedKG`), involves a novel optimization approach to learn Koopman operators over distributed datasets. This separates stationary components from non-stationary MTS and predicts non-stationary components by capturing the spatio-temporal relationships and underlying dynamics of decentralized non-stationary MTS. The second component, **Fed**erated **S**tationary **P**rediction (`FedSP`), employs a federated neural network to utilize stationary MTS for learning patterns among distributed data, thus enhancing the long-term prediction capabilities of Koopman predictors. In this section, we begin by introducing the problem formulation and matrix-based methods for Koopman modeling, followed by the motivation and details of the proposed FedKoop. A graphical illustration of the key steps is shown in Fig.4.3.

4.1.3.1 Problem Statement

Assuming a FL system with N clients, where each client $i \in \{1, ..., N\}$ has a local non-stationary MTS with n records (X_i, Y_i) as in 4.1.2.2. Inspired by OMD, our objective is to determine Koopman by a finite-rank operator (PQP^T) across all client datasets. Specifically, we aim to identify a low-rank matrix P and governing matrix Q that minimize the reconstruction error averaged across all clients. Drawing inspiration from the finite-sum formulation, we introduce the problem as:

$$\min_{P,Q} \quad \sum_{i=1}^{N} ||Y_i - PQP^T X_i||_F^2 \quad \text{s.t} \quad P^T P = I \tag{4.2}$$

4.1.3.2 ADMM for Federated Koopman Optimization

To address the problem (4.2) in a privacy-preserving manner, it is essential to reformulate (4.2) to fit a distributed algorithmic structure. By introducing a set of local

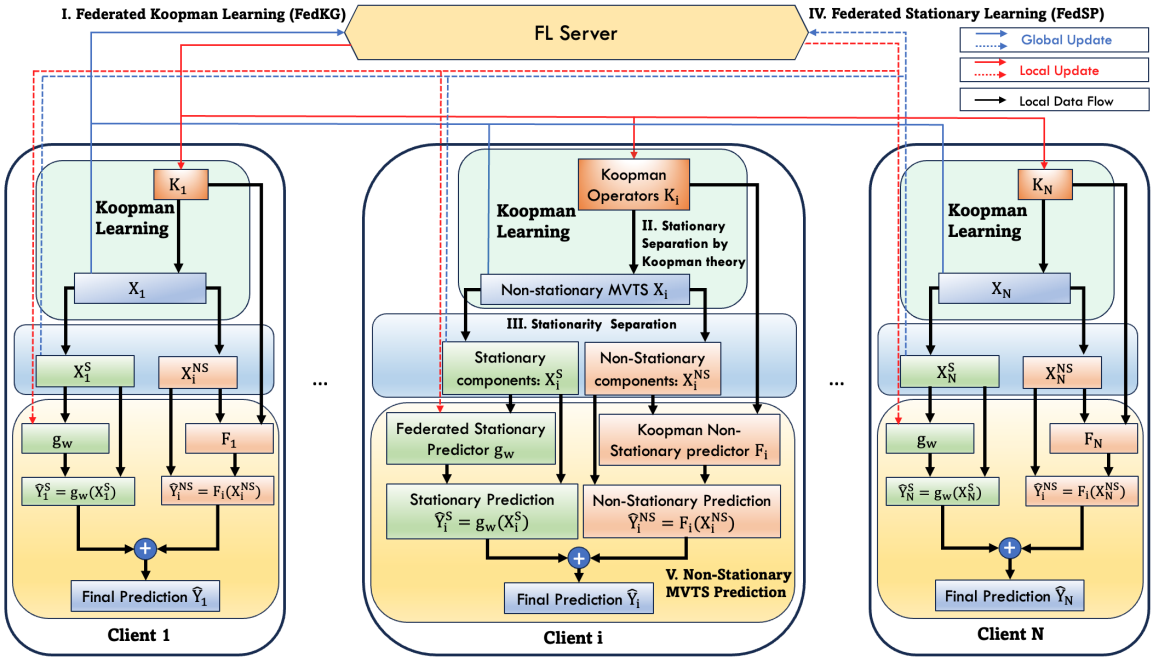

Figure 4.3: Federated Koopman learning for non-stationary MTS prediction

variables $\{P_1, ..., P_N\}$ and $\{Q_1, ..., Q_N\}$ we establish an equivalent problem to:

$$\begin{aligned} \min_{P_1,..,P_N,Q_1,...,Q_N} \quad & \sum_{i=1}^{N} \{f_i := ||Y_i - P_i Q_i P_i^T X_i||_F^2\} \\ \text{s.t} \quad & P_i = Z, \quad \forall i \in \{1, ..., N\} \\ & P_i^T P_i = I, \quad \forall i \in \{1, ..., N\} \end{aligned} \tag{4.3}$$

Given that the optimal dynamics represented by Q hinge on the chosen subspace, the solution to this constrained optimization problem is denoted as $Q_i^*(P_i)$. Notably, an analytical expression for Q_i can be derived by setting the partial derivative $\frac{\partial ||Y_i - P_i Q_i P_i^T X_i||_F^2}{\partial Q_i} = 2[-P_i^T X_i Y_i^T P + P^T X_i X_i^T P Q_i^T]^T$ to zero. The solution is:

$$Q_i^*(P_i) := P_i^T Y_i X_i^T P_i (P_i^T X_i X_i^T P_i)^{-1}.$$

However, determining the optimal solution is challenging due to the complexity introduced by the inverse operation, especially with a large matrix X_i (i.e., when the number of MTS is large and data is captured over a long period). Instead, we can seek an approximation $\hat{Q}_i(P_i)$ of Q_i^* by solving the following using an iterative optimization approach such as gradient descent.

$$\hat{Q}_i(P_i) = \underset{Q_i}{\text{argmin}} ||Y_i - P_i Q_i P_i^T X_i||_F^2. \tag{4.4}$$

Since Q_i depends on the chosen subspace, the task becomes finding the best dynamical data transformation for that low-dimensional subspace. Assuming $\hat{Q}_i(P_i)$ is updated

according to (4.4), we can reformulate (4.3) as follows:

$$\begin{aligned}\min_{P_1,..,P_N} \quad & \sum_{i=1}^{N}\{f_i := ||Y_i - P_i\hat{Q}_i P_i^T X_i||_F^2\} \\ \text{s.t} \quad & P_i = Z, \forall i \in \{1,...,N\} \\ & P_i^T P_i = I, \forall i \in \{1,...,N\}\end{aligned}$$

The first linear constraint ensures the "consensus" towards a shared low-rank matrix Z [190] for all local P_i, while the second constraint maintains consistency across clients by enforcing orthonormality for each P_i. Handling the non-linearity from the orthonormality constraint is challenging due to the problem's non-convex nature. To address this, we adopt a surrogate technique outlined in [151] to manage the non-linear constraint.

$$h_i(P_i) = \max\{0, P_i^T P_i - I\}^2. \tag{4.5}$$

Then the problem is cast to the following:

$$\begin{aligned}\min_{P_1,P_2,...,P_N} \quad & \sum_{i=1}^{N} ||Y_i - P_i\hat{Q}_i P_i^T X_i||_F^2 \\ \text{s.t} \quad & P_i = Z, \forall i \in \{1,...,N\} \\ & h_i(P_i) \leq 0, \forall i \in \{1,...,N\}\end{aligned} \tag{4.6}$$

Given the characteristics of problem (4.6) as a consensus optimization with linear constraints, we opt to use ADMM [150] for its resolution. The method alternates updates of primal and dual variables, aiming to align P_i across clients with a global representation Z and gradually minimize the objective function. To apply this approach to our specific problem, we begin by formulating the augmented Lagrangian function associated with the objective (4.6) as follows:

$$\begin{aligned}\mathcal{L}(\{P_i\}, Z, \{Y_i\}, \{T_i\}) = & \sum_{i=1}^{N} f_i(\hat{Q}_i, P_i) + \sum_{i=1}^{N}\langle Y_i, P_i - Z\rangle_F \\ + \sum_{i=1}^{N}\langle T_i, h_i(P_i)\rangle_F + \frac{\rho}{2}\sum_{i=1}^{N}\|P_i - Z\|_F^2 & + \frac{\rho}{2}\sum_{i=1}^{N}\|h_i(P_i)\|_F^2,\end{aligned} \tag{4.7}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product between two matrices, and ρ is penalty parameter for constraints. The updating procedure of ADMM for Federated Koopman in Euclidean space can be explained in the following steps:

Primal Update (Local Updates)

$$\hat{Q}^{k+1} = \underset{Q_i}{\text{argmin}}\{f_i(Q_i, P_i^k)\}. \tag{4.8}$$

$$\begin{aligned}P_i^{k+1} = \underset{P_i}{\text{argmin}}\Big\{ & f_i(\hat{Q}^{k+1}, P_i) + \langle Y_i^k, P_i - Z^k\rangle_F \\ & + \langle T_i^k, h_i(P_i)\rangle_F + \frac{\rho}{2}\|P_i - Z^k\|_F^2 + \frac{\rho}{2}\|h_i(P_i)\|_F^2\Big\}.\end{aligned} \tag{4.9}$$

Consensus Update (Global Update)

$$Z^{k+1} = \frac{1}{N}\sum_{i=1}^{N}\Big(P_i^{k+1} + \frac{1}{\rho}Y_i^k\Big). \tag{4.10}$$

Dual Update (Local Updates)

$$Y_i^{k+1} = Y_i^k + \rho\Big(P_i^{k+1} - Z^{k+1}\Big). \tag{4.11}$$

$$T_i^{k+1} = T_i^k + \rho\, h_i(P_i^{k+1}). \tag{4.12}$$

It is noteworthy that by substituting Z^{k+1} to Y_i^{k+1} leads to:

$$\frac{1}{N}\sum_{i=1}^{N} Y_i^{k+1} = 0. \tag{4.13}$$

This means Z update can be rewritten as:

$$Z^{k+1} = \frac{1}{N}\sum_{i=1}^{N} P_i^{k+1}. \tag{4.14}$$

By transforming the aggregation equation from (4.10) to (4.14), the ADMM is tailored for Federated Learning settings. This facilitates the distribution of computational workload across devices in the network, requiring only an aggregation server. Furthermore, communication efficiency is maintained by exchanging only two low-rank matrices P and Z.

4.1.3.3 Federated Koopman on Grassmann Manifold

Notably, the inclusion of the orthonormality constraint in equation (4.3) implies that the parameters from (4.9) must be projected onto Grassmann manifolds to accelerate convergence [185]. We introduce `FedKG` by formulating problem (4.3) within the context of the Grassmann manifold. The augmented Lagrangian (4.7) can be rephrased as:

$$\mathcal{L}(\{P_i\}, Z, \{Y_i\}) = \sum_{i=1}^{N} f_i(Q_i, P_i) + \sum_{i=1}^{N} Y_i^\top (P_i - Z) + \frac{\rho}{2}\sum_{i=1}^{N}\|P_i - Z\|_F^2. \tag{4.15}$$

In this context, the components related to $h_i(P_i)$ are excluded as the inherent orthonormality constraints are naturally met. The local update for each client i at the k-th iteration can be expressed as:

$$F_i^k(P_i) := f_i(Q_i^{k+1}, P_i) + Y_i^{k\top}(P_i - Z^k) + \frac{\rho}{2}\|P_i - Z^k\|_F^2.$$

The update regulations for P_i and Z within the Grassmann manifold are reformulated accordingly:

$$P_i^{k+1} = \underset{P_i}{\operatorname{argmin}}\Big\{F_i^k(P_i)\Big\}. \tag{4.16}$$

Algorithm 3 Federated Koopman on Grassmann Manifold (`FedKG`)

1: Randomly initialize Z^0 and $P_i^0, \quad \forall i = 1, \ldots, N$
2: **for** $k = 0, \ldots, T-1$ **do** ▷ *Global rounds*
3: **for** client $i = 1, \ldots, N$ in parallel **do**
4: $Q^{k+1} = \underset{Q_i}{\text{argmin}}\{f_i(\hat{Q}_i, P_i^k)\}$
5: $P_i^{k,0} = P_i^k$
6: **for** $c = 0, \ldots, C-1$ **do** ▷ *Local rounds*
7: $G_i^k(P_i^{k,c}) = (I - P_i^{k,c} P_i^{k,c,T}) \nabla_P F_i^k(P_i^{k,c})$
8: $P_i^{k,c+1} = R(P_i^{k,c} - \eta G_i^k(P_i^{k,c}))$
9: **end for**
10: $P_i^{k+1} = P_i^{k,C}$
11: **end for**
12: Updates $Z^{k+1} = \frac{1}{N} \sum_{i=1}^{N} P_i^{k+1}$ ▷ *Global update*
13: Broadcasts Z^{k+1} to all clients
14: **for** client $i = 1, \ldots, N$ in parallel **do** ▷ *Local update*
15: $Y_i^{k+1} = Y_i^k + \rho\Big(P_i^{k+1} - Z^{k+1}\Big)$
16: **end for**
17: **end for**

As outlined in Algorithm 3 (lines 7,8), projected gradient descent on the Grassmannian manifold is used at each local iteration on clients to determine the update of P_i^{k+1} in (4.16). We enforce the second constraint ${P_i}^T P_i = I$ by employing projected gradient descent on the Grassmannian manifold (Algorithm 5, line 8) as follows:

$$P_i^{k+1} = R(P_i^k - \eta G_i^k(P_i^k)), \tag{4.17}$$

Here, η represents the step size, and $\nabla_P F_i^k(P_i^k)$ denotes the gradient of $F_i^k(P_i^k)$ at point P_i^k. The retraction function $R(\cdot)$ corresponds to the projection operation conducted by QR decomposition [185]. We compute the updates by projecting the Euclidean gradient onto the tangent space of the manifold using orthogonal projection $\nabla_P F_i^k(P_i^k) \to G_i^k(P_i^k)$ (Algorithm 3, line 7) as follows.

$$G_i^k(P_i^k) = (I - P_i^k {P_i^k}^\top) \nabla_P F_i^k(P_i^k). \tag{4.18}$$

Note that the two equations (4.17) and (4.18) are repeatedly updated in C local rounds at lines 7-8 in Algorithm 3 to solve (4.16).

4.1.3.4 Stationary Component Separation

With an obtained global representation Z from `FedKG`, we now can find eigenvalues and dynamic modes for each client. Given $j \in \{1, \ldots, r\}$, the eigenvalues are defined by $\sigma_j := \frac{\log \lambda_j(\hat{Q}_i)}{\delta t}$, where $\lambda_j(\hat{Q}_i)$ are the eigenvalues of low order dynamics matrix $\hat{Q}_i$ and δt is the lag between the vectors. Associated with each eigenvalue σ_j is a dynamic mode $\phi_j := Z \upsilon_j \in \mathbb{R}^{m \times 1}$, where ϕ_j is defined in terms of the eigenvector υ_j and the low order subspace basis Z satisfying $\hat{Q}_i \upsilon_j = \lambda_j(\hat{Q}_i) \upsilon_j$. If $\Theta \in \mathbb{R}^{r \times r}$ is the matrix whose columns are the eigenvectors $\upsilon_j \in \mathbb{R}^{r \times 1}$ and $\Lambda \in \mathbb{R}^{r \times r}$ is a diagonal matrix containing the eigenvalues $\lambda_j(\hat{Q}_i)$, a finite-rank Koopman operator K_i of client i can

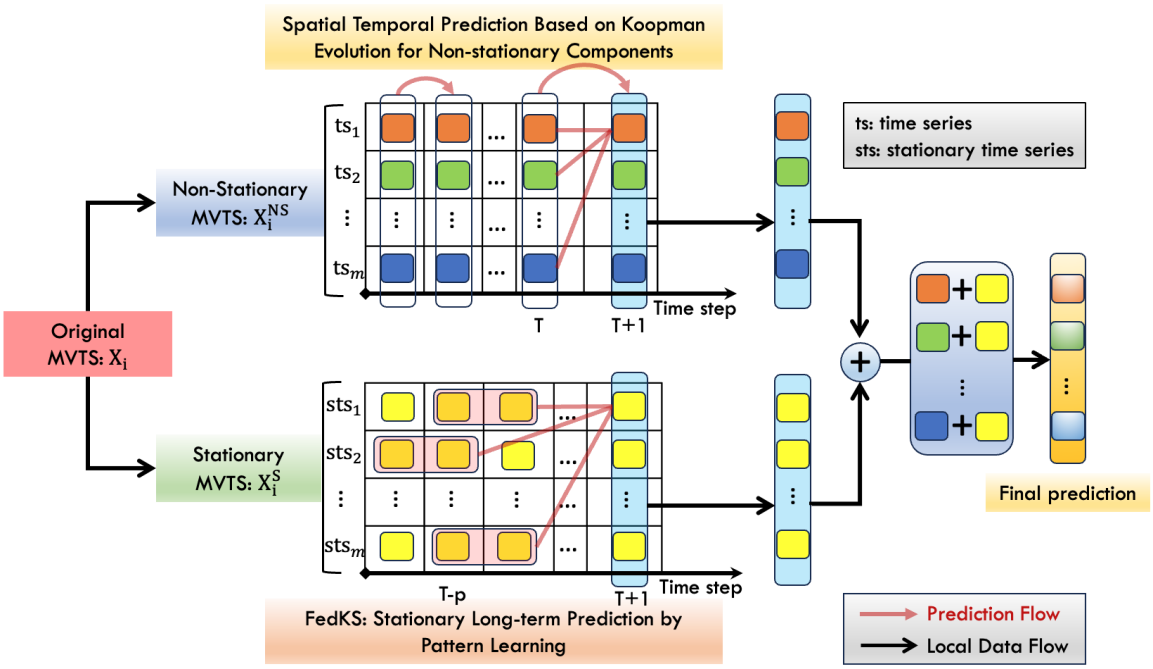

Figure 4.4: FedKoop for non-stationary MTS prediction

be express as:

$$K_i := Z\hat{Q}_i Z^T = (Z\Theta)\Lambda(\Theta^{-1}Z^T). \tag{4.19}$$

The columns of $Z\Theta$ represent the modes ϕ_j, with their frequencies determined by the Koopman eigenvalue σ_j. Each σ_j corresponds to a structure in the time series data, characterized by a temporal growth rate $Re(\sigma_j)$ and a frequency $Im(\sigma_j)$. Frequencies near the origin, where $||Im(\sigma_j)|| \approx 0$ indicate trend and seasonality, while those farther from the origin include harmonics of the time series [210]. Define $\Psi := [\psi_1, \psi_2, \ldots, \psi_r] := (\Theta^{-1})^T \in \mathbb{R}^{r\times r}$, and let

$$S = \{j \in \{1, \ldots, r\} | Im(\sigma_j) \approx 0\} \tag{4.20}$$

The non-stationary components is gathered by collecting modes in S:

$$X_i^{NS} := \sum_{j\in S} (Z\upsilon_j)\lambda_j(\psi_j Z^T)X_i \tag{4.21}$$

The stationary components are obtained by removing the non-stationary components:

$$X_i^S = X_i - \sum_{j\in S} (Z\upsilon_j)\lambda_j(\psi_j Z^T)X_i, \tag{4.22}$$

4.1.3.5 Long-Term Prediction with FedKoop

The Koopman operator, based on temporal dependence, is only accurate in the temporal prediction [211]. However, it effectively captures long term non-stationarity [211] by targeting dominant modes of the underlying dynamics. Koopman theory leverages temporal dependence, enhancing robustness to noise and data disturbances, as noise impacts only individual samples. Additionally, Koopman learning eliminates

Algorithm 4 Federated Stationary Prediction (FedSP)

initial $w_i^{(0)} \quad \forall i = 1, \ldots, N$
for $k = 0, \ldots, T-1$ **do** ▷ *Global rounds*
  Sample subset clients $\mathcal{S}^k$
  for each client $i \in \mathcal{S}^k$ in parallel **do** ▷ *Local rounds*
    compute mini-batch gradient $\nabla g_{w_i}^{(t)}$
    $w_{i,r+1}^{(t)} := w_{i,r}^{(t)} - \mu \nabla g_{w_i}^{(t)}$ ▷ *Local update*
  end for
  $w^{(k+1)} := \frac{1}{|S^k|} \sum_{i \in S^k} w_i^k$ ▷ *Global update*
end for

the need to select an appropriate window length for time series analysis, unlike Fourier Transform-based methods.

As illustrated in Fig. 4.4, a non-stationary MTS X_i of client i can be decomposed as a stationary component X_i^S and a non-stationary component X_i^{NS}. The prediction for non-stationary components, denoted as Y_i^{NS}, can be modeled by Koopman evolution (4.19) as:

$$\begin{aligned} Y_i^{NS} &= (Z\Theta)\Lambda(\Theta^{-1}Z^T)X_i^{NS} \\ &= (Z\Theta)\Lambda(\Theta^{-1}Z^T)[x_1 \ldots \quad x_n] \\ &= (Z\Theta)\Lambda(\Theta^{-1}Z^T)[x_1 \ldots \quad (Z\Theta)\Lambda^{n-1}(\Theta^{-1}Z^T)x_1] \end{aligned}$$

Thus, given an observed state at a previous time point $t \in \{1, .., n\}$, a future state at time $h > t$ can be estimated as:

$$y_h = (Z\Theta)\Lambda^{h-t}(\Theta^{-1}Z^T)x_t = \sum\nolimits_{j \in S} (Z\upsilon_j)\lambda_j^{h-t}(\psi_j Z^T)x_t.$$

The predictor $F_i(X_{NS})$ over p-steps for the non-stationary component X_i^{NS} is determined by:

$$F_i(X_i^{NS}) = \sum_{j \in S} (Z\upsilon_j)\lambda_j^p(\psi_j Z^T)x_t. \tag{4.23}$$

As shown in Fig. 4.1, the stationary MTS derived from the original non-stationary MTS retains similar patterns, which can be found across all clients. To mitigate weaknesses in Koopman evolution for long-term prediction, we utilize these patterns to enhance the model's predictive capability. Hence, we employ a FL system to develop a global predictive model g_w, parameterized by w, using distributed stationary data X_i^S. To facilitate pattern learning and reduce FL network communication overhead, g_w is implemented as a simple neural network with multi-input and multi-output. For a sequence x_k, where x_k is a column of X_i^S, with lookback window length B and and forecasting window length H satisfying $n - B > H > B \geq 1$, g_w is defined as follows

$$g_w : (x_B, ..., x_{B+L}) \mapsto (x_{B+L+1}, ..., x_{B+H}).$$

The detailed learning of g_w is implemented by FedSP, as presented in Algorithm 4. The prediction for non-stationary MTS data (X_i, Y_i) is illustrated in Fig. 4.4 and formulated as:

$$\hat{Y}_i = g_w(X_i^S) + F_i(X_i^{NS}).$$

4.1.4 Experimental Results

Table 4.1: Dataset Statistics

Dataset	No.time series (M)	No.training samples	Missing data (percentage)	Forecast horizon (H)
Traffic	860	17208	1.35%	$\{24, 48, 96, 168, 336\}$
Electricity	370	25968	7.65%	$\{24, 48, 96, 168, 336\}$
Kdd Cup 2018	210	10562	17.19%	$\{24, 48, 96, 168, 336\}$

4.1.4.1 Experimental Setup

Dataset: We conduct experiments on three widely used public datasets for spatio-temporal MTS prediction, which are representative of key applications in cyber-physical systems, including traffic control, climate forecasting, and energy management. The statistics of the datasets are summarized in Table 4.1, with further details provided below:

- **Electricity Dataset.** [4] The hourly electricity usage of 370 customers from 2012 to 2014. We convert the data to hourly consumption, consistent with prior research [212]. We utilize the initial 25968 data points for training and perform 336-hour ahead predictions for testing.
- **Traffic Dataset**[1]**.** The collection of 860 hourly time series depicts the road occupancy rates on the freeways in the San Francisco Bay area between 2015 and 2016. We use the first 17208 data points for training and conduct 336-hour ahead forecasts for testing.
- **KDD Cup 2018**[2]**.** The hourly air quality levels through 270 time series data. However, the last 60 time series are recorded in different periods with varying sample sizes. As a result, our decision is to focus on the initial 210 time series, which are recorded in the same period and consist of identical number of samples. Of these, 10562 samples are allocated for training and the last 336 hourly samples are dedicated for testing.

Federated Learning Settings: In a FL setting, local clients collect data for model training. Previous studies [198] have shown that Electricity, Traffic and KDD datasets are well-suited for spatio-temporal modeling in federated learning. To account for the diverse data distributions among clients, we divide the entire training set into $N = 10$ subsets, each containing M/N local time series. The scalability of the proposed method with respect to N is is analyzed in Section 4.1.4.2. In all experiments, the rank of P is chosen to be 20% of the original MTS, to ensure it captures the dominant modes. The FL aggregation server is implemented by FedAvg. At each global communication round k, we randomly select $|\mathcal{S}^k| = 10\%$ of total number of clients to participate in training. The number of global and local rounds are set to $T = 30$, $C = 5$ for `FedKG` and $T = 100$, $C = 50$ for the others. In the `FedKG`, trend and prevailing

[1] http://pems.dot.ca.gov
[2] https://zenodo.org/record/4656719

seasonalities are captured by setting (4.20) as $S = \{j \in \{1, \ldots, r\} | Im(\sigma_j) \leq \gamma\}$, where $\gamma = 0.1$ across all experiments. The lookback window size B is set to 168 for recursive multi-step forecast, and to $2H$ for multiple output forecast where H is the length of the forecasting horizon. The performance evaluation of MTS forecasting includes root mean squared error (RMSE) and mean absolute error (MAE) as key metrics. A better forecasting accuracy is inferred from lower values of RMSE and MAE. These metrics are chosen to ensure a fair comparison with the baseline results reported in the original papers, which use the same metrics. All reported results in the paper are obtained by averaging the outcomes of three runs for each algorithm.

Implementation Details: All experiments are conducted on a system with an AMD Ryzen 3970X processor (64 cores), 256GB RAM, and an NVIDIA GeForce RTX 3090 GPU. Models are trained using L2 loss and the Adam optimizer with a learning rate of 10^{-3}. Batch size is set to 32.

Table 4.2: Baseline Details

Algorithm	Configuration	Reference Source Code
FedSP	num-layers=2, hidden-size=64	
DeepAR	num-layers=4, hidden-size=64	https://ts.gluon.ai/
Prophet	fourier-order=10, prior-scale=0.1	https://github.com/facebook/prophet
FedInformer	num-encoder-layers=6, num-decoders-layers=4, hidden-size=64	https://huggingface.co
FedN-BEATS	n-blocks = [1, 1, 1], mlp-units = [[128, 128], [128, 128], [128, 128]]	https://pypi.org/project/NBEATS/
FedAutoformer	num-encoder-layers=6, num-decoders-layer=4, hidden-size=64, num-decompositional-blocks=25	https://huggingface.co
Fed-NST	num-encoder-layers=6, num-decoders-layer=4, hidden-size=64, num-decompositional-blocks=25, de-stationary-blocks=2, hidden-size=256	https://github.com/thuml/Nonstationary_Transformers

4.1.4.2 Experimental Results

Baselines: We thoroughly compare `FedKoop` (recursive multi-step forecast) and FedKoop-m (multiple output forecast, input-B output-H) with advanced approaches tailored for non-stationary time series. We firstly adapt Informer, Autoformer, and NST for FL as FedInformer, FedAutoformer, and Fed-NST. Informer denotes for an attention based mechanism approaches, showcasing superior long-term prediction performance compared to traditional methods. Autoformer represents for a class of models employing decomposition blocks and Fourier Transform to address non-stationary components, with forecasting performance enhanced by attention mechanism. NST stands for a class of algorithms that renders time series stationary by adding a block to eliminate non-stationary MTS variance and mean. Additionally, we incorporate DeepAR and Prophet [213], two industrial packages for time series forecasting known for their exceptional performance, to facilitate a better comparison with centralized settings. The configuration details are given in Table 4.2. Baselines' models are tested in a centralized setting to match their reported original performance before adaptation to FL version, while their model sizes are kept as small as possible to ensure a fair convergence study.

Forecasting Performance Comparison: The time series results of `FedKoop` and baseline methods across three datasets are presented in Table 4.3 highlighting the best results in bold and the second-best underlined. The results show that `FedKoop` and `FedKoop`-m generally outperform other baselines. However, `FedKoop`-m performs worse than `FedKoop` because the `FedKG` non-stationary prediction inherited from Koopman approach is optimized for one-step prediction. Multi-step predictions with `FedKoop` encounter challenges from accumulated errors, as indicated by equation (4.23). The superior performance of `FedKoop` is attributed to `FedKG`, which learns Koopman

Table 4.3: MTS forecasting results with $H \in \{24, 48, 96, 168, 336\}$.

Method		FedKoop (ours)		FedKoop-m (ours)		FedN-BEATS		Fed-Informer		Fed-Autoformer		Fed-NST		Prophet		DeepAR	
Metric		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Traffic	24	**0.18**	**0.06**	0.48	0.15	0.58	0.39	0.69	0.45	0.45	0.29	0.54	0.35	0.73	0.55	0.46	0.28
	48	**0.23**	**0.06**	0.39	0.14	0.52	0.34	0.79	0.61	0.61	0.34	0.61	0.39	0.82	0.56	0.54	0.31
	96	**0.26**	**0.07**	0.42	0.16	0.51	0.32	1.17	0.62	0.66	0.37	0.72	0.42	0.86	0.57	0.69	0.46
	168	**0.27**	**0.07**	0.48	0.19	0.61	0.43	1.21	0.89	0.65	0.35	0.73	0.47	0.87	0.58	0.85	0.49
	336	**0.35**	**0.08**	0.59	0.21	0.81	0.57	0.95	0.56	0.68	0.38	0.79	0.51	0.93	0.61	0.85	0.51
Electricity	24	**0.29**	**0.13**	0.41	0.24	0.70	0.53	1.29	0.93	0.54	0.51	1.29	0.98	1.28	0.88	0.48	0.32
	48	**0.29**	**0.12**	0.49	0.26	0.62	0.45	1.19	0.88	0.65	0.52	0.94	0.65	1.27	0.83	0.69	0.46
	96	**0.26**	**0.12**	0.51	0.25	0.75	0.55	1.88	1.14	0.79	0.56	0.86	0.59	1.21	0.81	0.86	0.65
	168	**0.27**	**0.11**	0.62	0.29	0.86	0.64	1.54	1.03	0.81	0.55	1.12	0.74	1.24	0.85	0.91	0.76
	336	**0.26**	**0.11**	0.63	0.31	0.93	0.72	2.03	0.71	0.79	0.55	0.96	0.71	1.19	0.81	0.89	0.72
Kdd Cup 2018	24	**0.21**	**0.11**	0.34	0.26	0.62	0.36	0.68	0.41	0.62	0.31	0.96	0.67	0.96	0.41	0.61	0.42
	48	**0.25**	**0.12**	0.45	0.29	0.61	0.43	0.86	0.49	0.71	0.39	1.01	0.74	1.02	0.41	0.73	0.49
	96	**0.29**	**0.11**	0.65	0.29	0.69	0.38	1.22	0.58	0.72	0.39	1.12	0.76	1.07	0.41	0.91	0.64
	168	**0.31**	**0.11**	0.64	0.28	0.75	0.47	0.72	0.29	0.73	0.41	1.02	0.75	1.01	0.37	0.96	0.71
	336	**0.32**	**0.11**	0.65	0.32	0.71	0.50	1.01	0.39	0.74	0.43	1.02	0.76	0.93	0.33	0.85	0.57

operators to identify stationary patterns and predict non-stationary components, as illustrated in Fig. 4.1. Notably, `FedKG` not only centers the multivariate time series around zero but also ensure identical variance. Fig. 4.1 confirms the separation of stationary and non-stationary time series in Electricity dataset, supported by the Augmented Dickey-Fuller test (p-value = 0.005) [6]. In contrast, FedAutoformer and FedN-BEATS use information from the same univariate time series to predict non-stationary components. However, changes in trend or seasonality in one time series can influence others in a multivariate context. Surprisingly, NST outperforms other baselines in centralized settings by using stationarization blocks that learn the mean and variance of non-stationary MTS. However, Fed-NST underperforms compared to FedAutoformer and FedN-BEATS because each client has distinct mean and variance, preventing the stationarization blocks from uniformly capturing non-stationary components across all clients. Although adapted to a Federated Learning setting, FedAutoformer and FedN-BEATS outperform conventional baselines like DeepAR and Prophet, consistent with their original reported results. While DeepAR performs well for 24-step predictions, its long-term predictions are less accurate compared to baselines like `FedKoop`, FedN-BEATS and FedAutoformer that address non-stationarity. Without this capability, FedInformer fails to predict non-stationary time series under FL settings.

Based on Table 4.1 and 4.3, the presence of missing data generates another type of distributional shifts and non-stationarity, degrading the performance of all algorithms as the number of missing samples increases. However, `FedKoop` is the least affected by this disturbance. The largest performance gap between `FedKoop` and the best baseline, FedAutoformer, is witnessed in the KDD dataset, where the missing ratio is 17.19%. FedAutoformer, which relies on Fourier Transform and Auto Correlation Function, is sensitive to missing data because it requires careful window length selection to capture the frequencies of univariate time series. This issue, known as spectral leakage [214], impairs its robustness. In contrast, `FedKoop` leverages spatio-temporal learning (as shown in Fig. 4.4). Thus, if missing data occur in a univariate time series, other time series still contribute to predictions, making `FedKoop` more robust to noise.

Scalability Study: We analyze the sensitivity of `FedKoop`-m to the number of clients N, due to the greater instability of `FedKoop`-m compared to `FedKoop`. Figure 4.5 illustrates 48-hour predictions for selected datasets, demonstrating that FedKoop-m

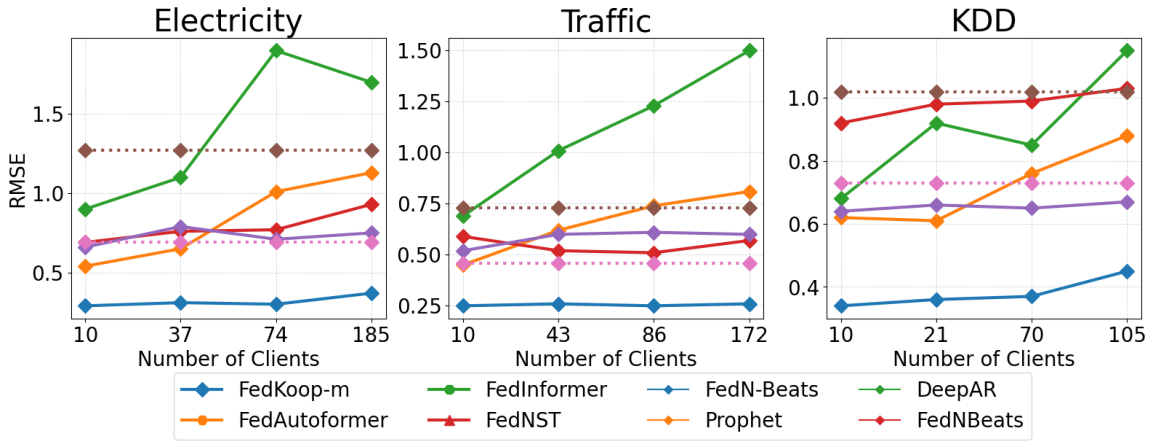

Figure 4.5: Large Scale Prediction Performance

exhibits superior robustness to N compared to other methods. As the number of clients increases, other baseline models struggle to learn the diverse patterns inherent in non-stationary data. Since the sub-sampling scheme is used in FL, the data coverage of baselines is reduced as N increases, leading to degraded performance of FL algorithms. Overall, the results confirm that `FedKoop` maintains stability and efficiency across varying numbers of clients.

Table 4.4: Training Time Record for 48-hour Forecast (s/global-iter)

Dataset	FedKoop-m (ours)	FedAutoformer	FedN-BEATS	FedInformer	Fed-NST
Traffic	**35.1**	63.7	66.5	71.2	69.5
Electricity	**22.5**	55.9	77.8	80.9	45.3
Kdd Cup 2018	**11.2**	39.4	48.3	35.4	42.6

Convergence Study: By recording the RMSE at each global iteration, we investigate the convergence rate, as shown in Fig. 4.6. For a balanced comparison, we evaluate our `FedKoop`-m against selected baselines. Due to the numerous baselines listed in Table 4.3, visualization becomes challenging. Thus, we typically select FedAutoformer, the best-performing baseline, to represent algorithms capable of handling non-stationary time series, and FedInformer to represent those without this capability. The results indicate that `FedKoop` achieves a faster convergence rate than other models, reaching convergence within five global training rounds. These findings highlight its efficiency in learning stationary data over a large number of clients in FL. All algorithms in Fig. 4.6 converge more quickly with fewer clients and struggle as the number of clients increases, consistent with FL scenarios where more clients reduce data coverage. FedInformer, lacking the ability to handle non-stationary data, exhibits greater fluctuations compared to the others. Additionally, Table 4.4 demonstrates that `FedKoop` achieves the shortest wall-clock training time. In contrast, other attention-based approaches may lack GPU support due to resource scarity. This further demonstrates that `FedKoop` is well-suited for cyber-physical systems, where clients are equipped with resource-limited computing.

Ablation Study:

Experimental results in Section 4.2.4.2 highlight the superior performance of `FedKoop`, attributed to `FedKG`, where r and γ are discussed as critical factors in

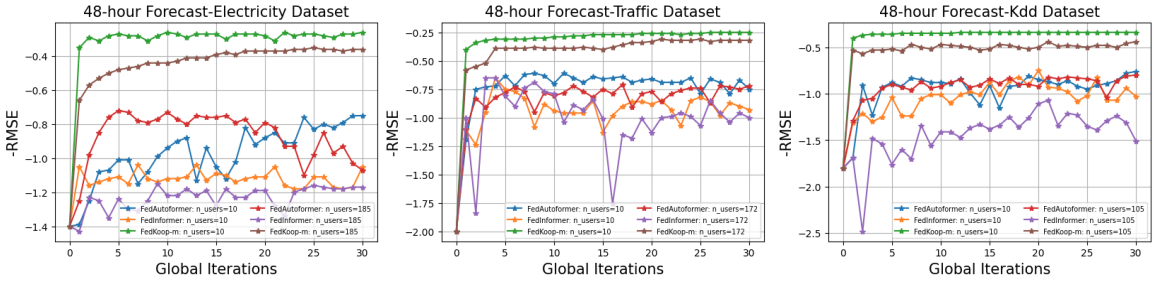

Figure 4.6: Convergence Analysis

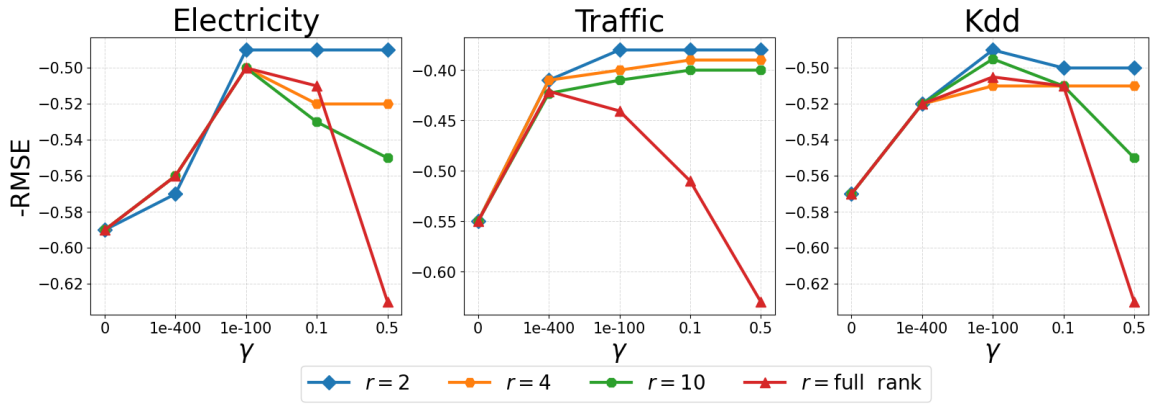

Figure 4.7: Effect of γ and r on `FedKoop` Performance

Section 4.2.3.6. A lower value of r isolates dominant Koopman modes, typically capturing core trends and strong seasonal patterns. In contrast, a high-dimensional r makes γ essential for distinguishing between non-stationary and stationary components, as discussed in (4.22). Ablation results in Fig. 4.7 show that smaller r values simplify the selection of an appropriate γ threshold. For example, with $r = 2$, a threshold of $\gamma = 0.1$ effectively identifies two dominant modes. However, increasing r leads `FedKoop` to include modes with minimal influence, highlighting the vital role of γ in non-stationary identification.

4.2 FedKooL: Communication-Efficient Federated Koopman Learning for Wireless Traffic Prediction on Grassmann Manifolds

4.2.1 Introduction

The dawn of fifth-generation (5G) networks in 2019 revolutionized global communication, laying the groundwork for sixth-generation (6G) technologies. Innovations such as terahertz spectrum, space-air-ground communications, large reflecting surfaces, and cognitive radios are poised to reshape our digital landscape [215]. These advancements coincide with the proliferation of heterogeneous service requirements, necessitating seamless integration of high-rate, real-time transmission capabilities [216]. This highlights the critical need for wireless traffic prediction, essential for optimizing network management and ensuring Quality of Service (QoS) and Quality of Experience (QoE) from both the network and user perspectives [217]. For instance, early forecasts of wireless traffic enhance network performance by proactively offloading traffic from congested macro cells, thereby reducing latency and significantly improving the overall QoE for users.

The wireless traffic prediction task primarily involves modelling complex spatio-temporal structures in MTS data [218]. Given the nature of traffic data, which often exhibits spatial dependencies due to the interconnection of network systems, it is crucial for the prediction model to capture not only the temporal dynamics of traffic but also the spatial inter-relationships among different data points [219]. However, this task is complicated by the inherent non-stationarity of MTS traffic data, where statistical properties evolve over time [220]. These changes can mask primary spatio-temporal dependencies, as illustrated in Fig.4.8. In future network systems, non-stationarity can arise from numerous sources, including environmental changes, seasonality, trends, or other underlying processes [102]. For example, prediction models trained on 4G or 5G data may not generalize to 6G due to significant shifts in data usage trends. Moreover, missing data resulting from unstable connections in Heterogeneous Networks (HetNets) [221–223] can distort statistical properties, such as mean and variance, leading to non-stationarity. This further complicates real-world traffic prediction by obscuring temporal dependencies. Despite these detrimental effects, understanding non-stationarity provides valuable insights into user behavior, which is crucial for effective network traffic management.

To cope with these problems, frequent modeling with updated datasets using statistical methods including ARIMA [104] and VAR [199] is popular. However, these methods handle only univariate time series and ignore spatial dependencies. Alternatives like ETS [200] and deep learning models such as RNNs [201] and DeepAR [202] offer more robust solutions with capability of learning spatial relations and retaining information over extended periods, but they still overlook non-stationarity obstacles. Very recently, advanced machine learning architectures with attention mechanisms, such as Autoformer [107] and Non-Stationary Transformer (NST) [110], have emerged for handling long sequences and addressing non-stationarity effectively. While these approaches show promising results for MTS prediction, they rely heavily on centralized data and do not fit for modeling wireless traffic for several reasons.

First, these methods neglect the distributed nature of data and overlook the inherent spatial inter-relationship of wireless traffic, as well as the presence of missing data produced by HetNets. **Second**, gathering all traffic data is often constrained by privacy concerns, access license agreements, collection costs, and computational limitations.

In recent years, FL has emerged as a solution, enabling machine learning models to be trained on decentralized data while protecting against data leakage. In FL, models are trained locally on clients' devices using their own data. Recent FL variants for wireless traffic prediction incorporate advanced models to capture spatio-temporal dependencies [198, 218, 224]. However, these approaches, like their centralized counterparts, often increase deep learning model sizes, which impairs interpretability and incurs significant communication bandwidth costs. Additionally, capturing spatial inter-relationships with presence of non-stationarity for traffic data remains an unanswered question. As a result, a robust wireless traffic prediction architecture is needed that captures spatio-temporal dependencies under non-stationarity while considering the effect of other clients, maintains data decentralization, and conforms to resource constraints in FL.

In this work, we introduce `FedKooL`, an innovative framework inspired by `FedKoop`, which adapts Koopman theory for federated learning scenarios. Both `FedKoop` and `FedKooL` can address decentralized, non-stationary time series prediction, but `FedKooL` showcases an alternative approach for solving the objective function. Specifically, `FedKooL` tailors the framework for wireless traffic prediction while combining the decentralized, privacy-preserving features of federated learning with the analytical and spatio-temporal predictive capabilities of Koopman operators. By examining dependencies between wireless traffic data through the lens of dynamical systems, `FedKooL` effectively models interactions and impacts across different clients, addressing challenges posed by non-stationary components. This integration enables a robust understanding of complex traffic behaviors while maintaining data privacy and eliminating the need for centralized data aggregation. The main contributions are summarized as follows:

- We propose `FedKooL`, a privacy-preserving FL framework for modeling wireless traffic data. By leveraging Koopman theory, `FedKooL` learns the non-stationary and stationary components of traffic data separately. This enables the capture of hidden spatio-temporal dynamics across multiple clients and effectively manages the evolution of non-stationarity.

- We propose an optimization framework in `FedKooL`. Inspired by the manifold hypothesis, which posits that many high-dimensional data sets lie along low-dimensional subspace, `FedKooL` utilizes low-rank matrices to capture the dynamics of time series data on these low-dimensional linear subspace. We employ Rockafellar's envelope to leverage the advantages of low-rank parameters, addressing the computational complexity challenges inherent in the `FedKooL`'s optimization. Moreover, the learning process of `FedKooL` is accelerated by implementing gradient movement on Grassmann manifolds.

- Experimental results show that our `FedKooL` surpasses other baselines in wireless traffic prediction performance and better generalizes to never-before-seen data

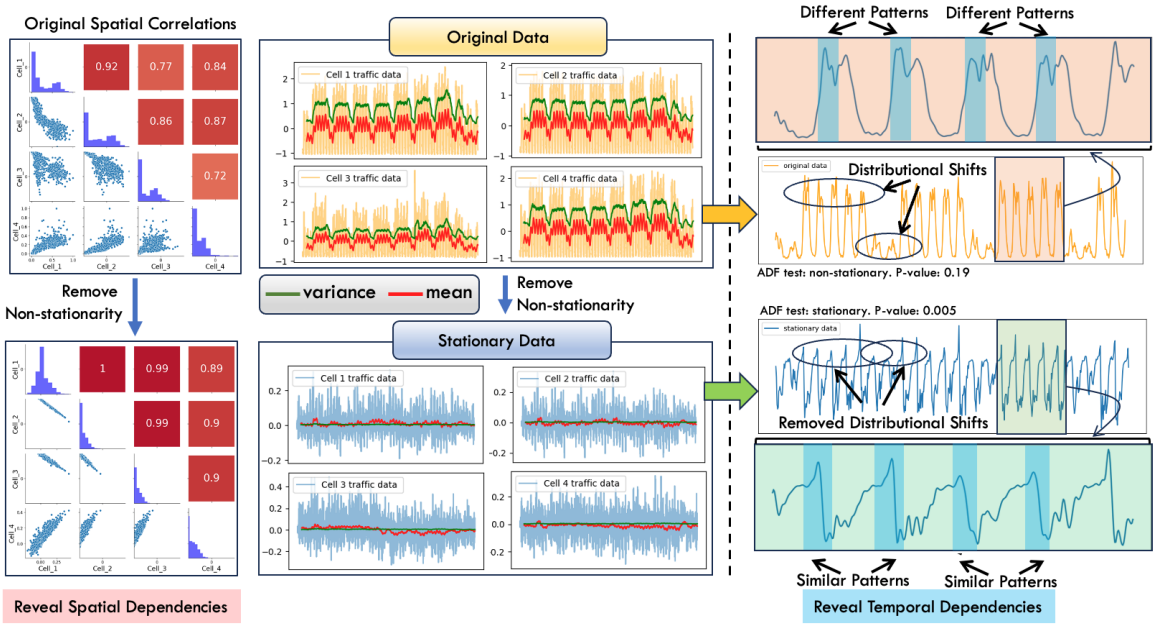

Figure 4.8: Koopman theory is applied to the Telecom Italia dataset [5] to separate non-stationary components from stationary dynamics by projecting the time series into Koopman-invariant subspaces. This yields zero-mean stationary data, aligns patterns to a common range, and uncovers spatial correlations between traffic series. Stationarity is confirmed using the Augmented Dickey-Fuller test with a p-value of 0.005 [6].

while demanding fewer computational resources with less memory. It achieves significantly lower training and prediction times and can handle a large number of clients.

4.2.2 Wireless Traffic Time Series as Dynamics

In wireless traffic modelling, capturing the relationships between interdependent traffic data is vital for network management and optimization [219, 225, 226]. These relationships can include dependencies within groups of base stations (intra-spatial) [226] or between different groups (inter-spatial) [219] as illustrated in Fig. 4.9. Without understanding these dynamic interactions, network operators may struggle to predict and address performance issues, resulting in poor QoE for users and inefficient resource allocation [216, 218]. For example, during a major sports event, a surge in traffic at one base station can cause congestion, leading users to offload to nearby stations. If these adjacent stations are not prepared to handle the sudden influx of users, they may become congested, causing a cascading network degradation. By modelling these interdependencies as a dynamic system that accounts for how changes at one point affect others, operators can better manipulate traffic shifts and implement more effective load balancing for congestion management, ultimately enhancing overall network performance.

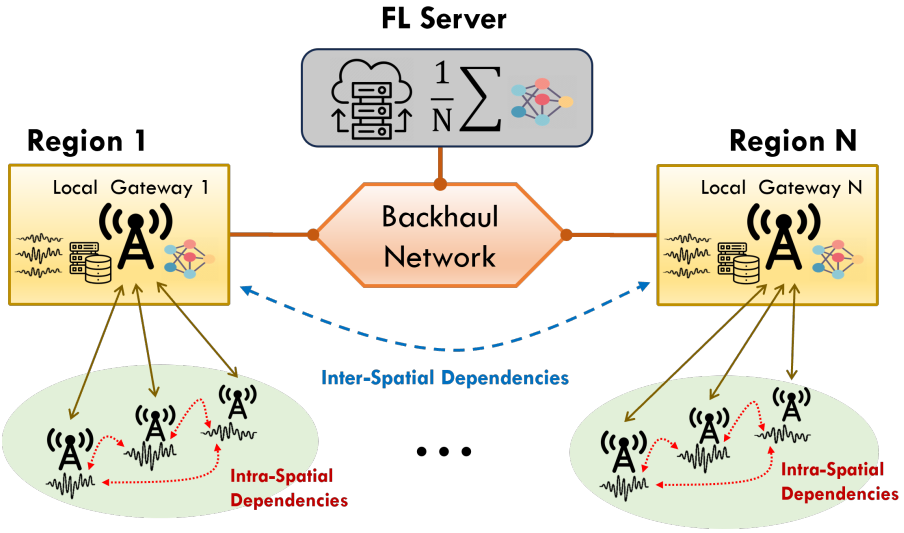

Figure 4.9: System model.

4.2.3 Federated Koopman Methodology

4.2.3.1 System Model

As illustrated in Figure 4.9, we propose a federated wireless traffic prediction framework for HetNets [221, 223] in future network systems, which involve multiple local cell gateways (GWs) managing traffic across base stations. Notably, the task of managing traffic data in cell GW is vital for maintaining seamless connectivity in HetNets and plays an important role in the allocation of resources among small cells to maximize coverage, capacity, and QoS [222]. By integrating FL into these GWs, they act as secure FL clients, performing model training on local traffic data. These GW then communicate with a global FL server via a reliable backhaul network to facilitate traffic prediction across the HetNets. This integration provides several benefits: it efficiently manages numerous small cells while preserving data privacy by keeping sensitive data locally and only sharing model updates. It also distributes the computational load, reducing the burden on centralized systems, thereby enhancing scalability and resilience. Moreover, this approach optimizes bandwidth utilization by minimizing data transmission, crucial for handling high volumes of diverse data. Predictive capabilities for traffic in small cell GWs enable better anticipation and management of traffic surges, leading to more efficient resource allocation, thereby improving QoS, and QoE for users.

The FL process between local cell GWs and the FL server involves initial server-client computations to derive a low-rank Koopman representation of network traffic. During each global communication iteration, the FL server broadcasts the current global model to GWs, which update their local models using their respective data. The FL server aggregates these updated local models from a subset of clients to iteratively refine a new global model until convergence is achieved. After obtaining the global Koopman representation, each client uses it to personalize Koopman operators for non-stationary removal, thereby enhancing spatial-temporal correlations among GWs' traffic data. The resulting stationary data is then used in another FL training phase

between GW and the FL server, leveraging a shallow neural network for patterns learning. Utilizing global Koopman representations facilitates efficient identification of non-stationary components and improves the generalizability of the trained neural network to distributional shifts. Additionally, this Koopman based approach is noted for its speed and lightweight nature, making it suitable for resource-constrained environments (details in Section 4.2.4.2).

4.2.3.2 Centralized Koopman Learning

In our system model, each GW client $i \in \{1, \ldots, N\}$ holds a local dataset $D_i = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$, where $x_1 \in \mathbb{R}^m$ represents a record of m base stations (e.g., a GW client i logs traffic data of $m = 100$ base stations with hourly interval of $n = 24$ record values during 24 hours). Koopman learning aims to identify the Koopman operator that minimizes the reconstruction error, given by $\sum_{t=1}^{n-1} \|\xi(x_{t+1}) - K_\varphi \xi(x_t)\|$ [207, 227]. Current advanced methods for approximating the Koopman operator, such as Koopman Autoencoder [227] and Extended Dynamic Mode Decomposition [207], often rely on multilayer neural networks to solve the loss function. These approaches are computationally intensive and exhibit performance uncertainties due to their limited adherence to Koopman theory constraints [228]. In contrast, OMD [209] focuses on learning the dominant Koopman operator through solving an optimization problem with lower computational costs. By rearranging the dataset D_i into matrices (X_i, Y_i) where $X_i = [x_1, x_2, \ldots, x_{n-1}]$ and $Y_i = [x_2, x_3, \ldots, x_n]$, OMD identifies optimal Koopman operator by minimizing $\|Y_i - PQP^T X_i\|_F^2$ subject to $P^T P = I$, where $\|.\|_F$ is the Frobenius norm. This constraint ensures that orthogonal columns of $P^T \in \mathbb{R}^{m \times r}$ span a r-dimensional subspace. The Koopman matrix in (4.1) is approximated by $A \approx PQP^T$, with P^T as ξ transformation and Q serving as K_φ to govern state transitions. Typically, P^T projects the original data to a low-dimension subspace with rank r, $Q \in \mathbb{R}^{r \times r}$ operates linearly in this r-dimensional subspace, and P lifts it back to the original space with rank m, where $r \leq m$.

4.2.3.3 Problem Formulation

Assume a FL system consisting of N gateway clients, where each client $i \in \{1, \ldots, N\}$ possesses m local MTS traffic data with n records from cells (X_i, Y_i) as described in Section 4.2.3.2. Inspired from OMD and the manifold hypothesis [229], which posits that many high-dimensional datasets found in real-world scenarios are embedded along low-dimensional latent manifolds within the high-dimensional space [230], our objective is to approximate each client's Koopman operator through a finite-rank matrix representation PQ_iP^T using all clients' datasets (X_i, Y_i), $\forall i$. Specifically, we aim to identify a common low-rank matrix P and a governing matrix Q_i of each client that minimizes the average reconstruction error as follows.

$$\texttt{FedKooL}: \quad \min_{P,\{Q_i\}} \quad \sum_{i=1}^{N} ||Y_i - PQ_iP^T X_i||_F^2 \qquad \text{s.t.} \quad P^T P = I. \tag{4.24}$$

`FedKooL` poses notable challenges such as non-convexity due to the term PQ_iP^T and the orthogonality constraint.

4.2.3.4 Federated Koopman Learning

To solve `FedKooL` in a privacy-preserving manner, we reformulate it to fit a distributed algorithmic structure. By introducing local variables $\{P_1, \ldots, P_N\}$, we have an equivalent problem:

$$\begin{aligned} \min_{\{P_i\},\{Q_i\}} \quad & \sum\nolimits_{i=1}^{N} \{f_i(Q_i, P_i) \coloneqq ||Y_i - P_i Q_i P_i^T X_i||_F^2\} \\ \text{s.t.} \quad & P_i^T P_i = I, \quad \forall i \in \{1, ..., N\} \\ & P = P_1 = P_2 = \ldots = P_N. \end{aligned} \tag{4.25}$$

Given that the objective of the above optimization problem can be decomposed for each client i, we further simplify the problem by finding $Q_i^*(P_i)$ as follows:

$$Q_i^*(P_i) = \text{argmin}_{Q_i} \; f_i(Q_i, P_i). \tag{4.26}$$

Notably, an analytical expression for $Q_i^*(P_i)$ can be derived by setting $\frac{\partial ||Y_i - P_i Q_i P_i^T X_i||_F^2}{\partial Q_i} = 2[-P_i^T X_i Y_i^T P + P^T X_i X_i^T P Q_i^T]^T = 0$, which gives the following result

$$Q_i^*(P_i) = P_i^T Y_i X_i^T P_i (P_i^T X_i X_i^T P_i)^{-1}. \tag{4.27}$$

At first, readers might be intimidated by the matrix inversion term. But we note that $(P_i^T X_i X_i^T P_i)$ is a square matrix of rank r; thus its inverse's complexity is $O(r^3)$. Considering the low-dimension parameter r is in the order of tens or hundreds, this matrix inverse's complexity is negligible in modern computing.

Then `FedKooL` is equivalent to the following:

$$\begin{aligned} \min_{\{P_i\}} \quad & \sum\nolimits_{i=1}^{N} \{h_i(P_i) \coloneqq ||Y_i - P_i Q_i^*(P_i) P_i^T X_i||_F^2\} \\ \text{s.t.} \quad & Q_i^*(P_i) = P_i^T Y_i X_i^T P_i (P_i^T X_i X_i^T P_i)^{-1} \\ & P_i^T P_i = I, \forall i \in \{1, ..., N\} \\ & P = P_1 = P_2 = \ldots = P_N. \end{aligned} \tag{4.28}$$

To solve the above problem based on iterative gradient descent methods, we next show how to obtain $\nabla h_i(P_i)$. Since (i) f_i is continuously differentiable in P_i for all Q_i, (ii) $\nabla_1 f_i$ (i.e., the gradient of f_i w.r.t. its first variable) is continuous, and (iii) the minimum $Q_i^*(P_i)$ is unique, h_i is differentiable and thus by Rockafellar's envelope theorem [231] we obtain:

$$\begin{aligned} \nabla h_i(P_i) &= \nabla_2 f_i(Q_i^*(P_i), P_i) \\ &= 2X_i X_i^T P_i Q_i^{*T} Q_i^* + 2P_i Q_i^* P_i^T X_i X_i^T P_i Q_i^{*T} \\ &\quad - 2X_i Y_i^T P_i Q_i^* - 2Y_i X_i^T P_i Q_i^{*T}. \end{aligned} \tag{4.29}$$

Then, we propose Algorithm 5 to solve the optimization problem (4.28). If we temporarily remove the last "consensus" constraint of problem (4.28), we see that this problem is client-decomposable. Thus, each client i can independently minimize its objective $h_i(P_i)$ while satisfying other constraints by using gradient descent in finite C iterations at lines 6–11 of Alg. 5.

Algorithm 5 Federated Koopman Learning Algorithm

1: Randomly initialize $P_i^0, \quad \forall i = 1, \ldots, N$
2: Each client i computes $X_i X_i^T$, $X_i Y_i^T$, and $Y_i X_i^T$.
3: **for** $k = 0, \ldots, T-1$ **do** ▷ *Global rounds*
4: **for** client $i = 1, \ldots, N$ in parallel **do**
5: $P_i^{k,0} = P^k$
6: **for** $c = 0, \ldots, C-1$ **do** ▷ *GW client updates*
7: Update $\nabla h_i(P_i)$ by Eq. (4.29)
8: $G_i^k(P_i^{k,c}) = (I - P_i^{k,c} P_i^{k,c\top}) \nabla h_i(P_i^{k,c})$
9: $P_i^{k,c+1} = R(P_i^{k,c} - \eta G_i^k(P_i^{k,c}))$
10: **end for**
11: $P_i^{k+1} = P_i^{k,C}$
12: **end for**
13: Updates $P^{k+1} = \frac{1}{N} \sum_{i=1}^N P_i^{k+1}$ ▷ *Server update*
14: $P^{k+1} = R(P^{k+1}) \leftarrow$ projected to the Grassmann manifold
15: Broadcasts P^{k+1} to all clients
16: **end for**

Specifically, since $P_i^T P_i = I$ implies that P_i lies on a Grassmann manifold, we first project the Euclidean gradient $\nabla h_i(P_i^k)$ onto the tangent plane of the Grassmann manifold using orthogonal projection $\nabla h_i(P_i^k) \to G_i^k(P_i^k)$ as follows:

$$G_i^k(P_i^k) = (I - P_i^k P_i^{k\top}) \nabla h_i(P_i^k). \tag{4.30}$$

Then, P_i will be updated on the tangent plane $P_i^k - \eta G_i^k(P_i^k)$, and projected back to the Grassmann manifold

$$P_i^{k+1} = R(P_i^k - \eta G_i^k(P_i^k)), \tag{4.31}$$

where η is the step size and $R(\cdot)$ is a function performing Grassmannian projection using QR decomposition [185].

Finally, the last "consensus" constraint of the problem (4.28) will be handled using the standard FL technique. At each global communication round denoted by k, each client i sends its updated local version P_i^{k+1} to the FL server for global update (line 13 of Alg. 5). The global representation is guaranteed by a projection to the Grassmann manifold (line 14 of Alg. 5). And this global P^{k+1} will be broadcast to each client for the next local update (line 4).

4.2.3.5 `FedKooL`: Computational Complexity Analysis

We delineate the computational complexity of FedKooL, examining both local and global computational demands. We first recap the primary variables and their dimensions integral to the computation process of each client i, including: (1) the data matrices $X_i, Y_i \in \mathbb{R}^{m \times n}$; (2) the low-rank matrix $P_i \in \mathbb{R}^{m \times r}$; and (3) the operator $Q_i \in \mathbb{R}^{r \times r}$, where $r \leq m$. In addition, we denote C and T as the number of local and global rounds, respectively.

The local computational overhead in FedKooL (lines 7–9 of Alg. 5) is primarily driven by two key steps executed within each local round. Particularly, the gradient $\nabla h(P_i)$ is updated according to Eq. (4.29) (line 7 of Alg. 5), which require the computation of

the optimal operator Q_i^* alongside multiple matrix products that integrate all relevant matrix variables. It is important to note that operations involving data matrices, such as $X_i X_i^T$, $X_i Y_i^T$, and $Y_i X_i^T$, are computed only once with a complexity of $O(m^2 n)$ at the outset of the training phase, thus amortizing their costs across subsequent gradient calculations. The calculation of Q_i^* using Eq. (4.27) requires matrix multiplications and an inversion, resulting in the complexity of $O(m^2 r + r^2 m + r^3)$. This, combined with the other matrix multiplication steps for updating $\nabla h(P_i)$, leads to an overall computational complexity per local iteration estimated at $O(m^2 r + m r^2 + r^3)$.

Afterward, the projection of the gradient $\nabla h(P_i)$ onto the manifold's tangent space (line 8 of Alg. 5) entails basic matrix multiplications with a computational cost of $O(mr^2)$. The retraction step (line 9 of Alg. 5), which updates P_i via QR decomposition, similarly demands a complexity of $O(mr^2)$. Taking all of these complexities into account, the total computational burden per client per local iteration is $O(m^2 k + mk^2 + k^3)$. Accumulating the costs over C rounds results in a total complexity of $O(C(m^2 r + m r^2 + r^3))$. This suggests that `FedKooL` is capable of handling time-series data of varying lengths, as its computational complexity depends solely on the dimensions m and r, with $r \leq m$. This characteristic not only facilitates the effective management of high-dimensional data through substantial dimensionality reduction, but also ensures that increases in the temporal depth of the data do not directly impact computational demands.

At each global communication round, the global coordinator necessitates the averaging of matrix P across a selected subset S of clients (line 13 of Alg. 5) and the retraction step to project P to manifold (line 14 of Alg. 5), resulting in an overall complexity of $O(S(mr^2))$. With T global rounds, the overarching computational load for the global coordinator culminates to $O(TS(mr^2))$.

4.2.3.6 Stationary Component Separation

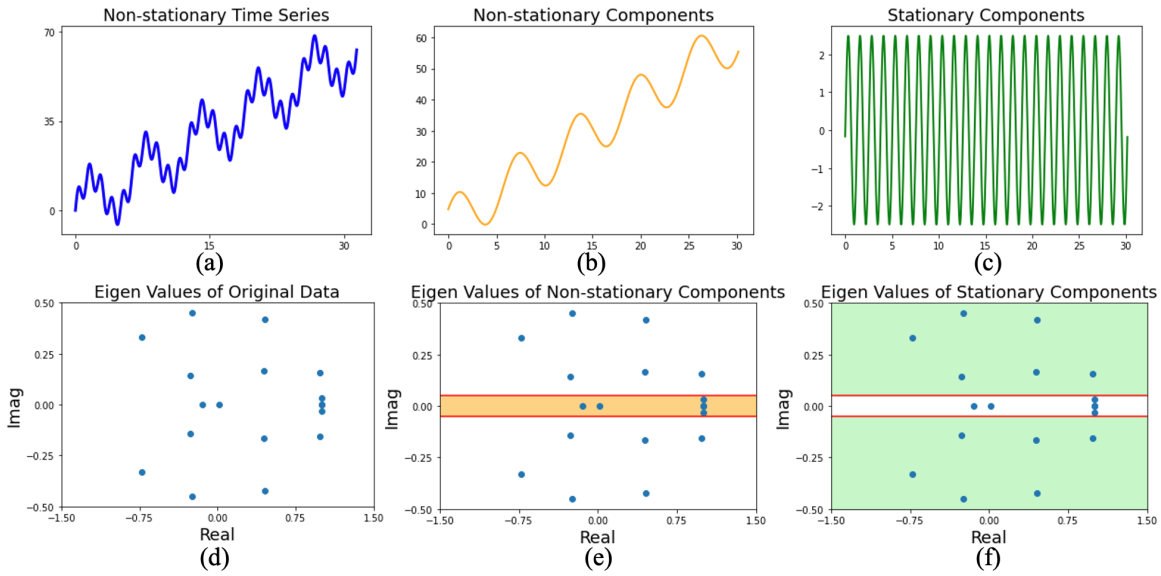

Figure 4.10: Non-stationary components removal with Koopman

Employing the global representation P from `FedKooL`, we can separate non-stationarity using the eigenvalues and dynamic modes of Koopman operators for each client. For

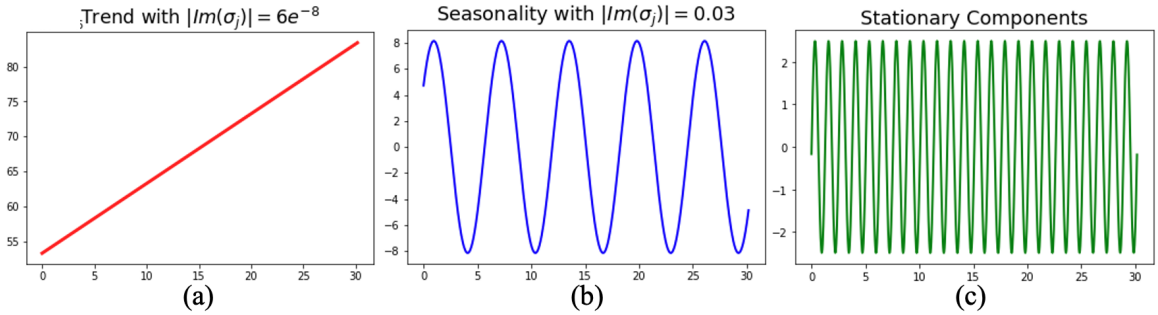

Figure 4.11: Trend and seasonalities deconstruction with Koopman

$j \in \{1, \ldots, r\}$, the eigenvalues are defined as $\sigma_j := \frac{\log \lambda_j(Q_i^*)}{\delta t}$, where $\lambda_j(Q_i^*)$ are the eigenvalues of the dynamics matrix Q_i^*, and δt is the time lag between snapshots x_t and $x_{t+\delta t}$. Each σ_j is associated with a dynamic mode $\phi_j := P\upsilon_j \in \mathbb{R}^{m\times 1}$, defined by the eigenvector υ_j and the subspace basis P, satisfying $Q_i^*\upsilon_j = \lambda_j(Q_i^*)\upsilon_j$. Let $\Theta \in \mathbb{R}^{r\times r}$ be the matrix with columns as eigenvectors $\upsilon_j \in \mathbb{R}^{r\times 1}$, and $\Lambda_i \in \mathbb{R}^{r\times r}$ be the diagonal matrix of eigenvalues $\lambda_j(Q_i^*)$. The finite-dimensional Koopman operator approximation for client i is represented as:

$$PQ_i^*P^T = (P\Theta)\Lambda_i(\Theta^{-1}P^T). \tag{4.32}$$

The columns of $P\Theta$ correspond to the modes ϕ_j, with frequencies specified by the Koopman eigenvalue σ_j. Each σ_j encapsulates a time series pattern, characterized by a growth rate $Re(\sigma_j)$ and an oscillatory component $Im(\sigma_j)$. Frequencies close to zero, where $||Im(\sigma_j)|| \approx 0$, indicate trends and seasonal components, while those farther from zero signify higher-order harmonics [210]. Define $\Psi := [\psi_1, \psi_2, \ldots, \psi_r] := (\Theta^{-1})^T \in \mathbb{R}^{r\times r}$, and let

$$\mathcal{S} := \{j \in \{1, \ldots, r\} | ||Im(\sigma_j)|| \approx 0\}. \tag{4.33}$$

Then, the non-stationary components are identified by:

$$X_i^{NS} = \sum\nolimits_{j\in\mathcal{S}} (P\upsilon_j)\lambda_j(\psi_j P^T)X_i. \tag{4.34}$$

The stationary components are derived by subtracting the non-stationary components:

$$X_i^S = X_i - \sum\nolimits_{j\in\mathcal{S}} (P\upsilon_j)\lambda_j(\psi_j P^T)X_i. \tag{4.35}$$

To visualize of Koopman methods in detecting non-stationary components, we investigate the method with a synthetic data defined by Table 4.5. We generate a sequence $J = [f(1), f(2), \ldots, f(1000)]$ by sampling 1000 continuous points from the function $f(t)$. Using the system model described in (4.2.3.3), we simulate an FL system with 10 clients, each possessing 40 randomly drawn time series segments from J. We apply `FedKooL` to obtain the matrix P, which is used to identify the Koopman operators as defined in (4.32). By filtering the Koopman operators according to the criteria in (4.33), which correspond to the orange region in Fig. 4.10e, we extract non-stationary components using (4.34), as shown in Fig. 4.10b. Stationary components correspond to eigenvalues in the green region, illustrated in Fig. 4.10c and 4.10d. By analyzing different values of $|\mathrm{Im}(\sigma_j)|$, we further decompose the data into trends and seasonal patterns, as depicted in Fig. 4.11.

Table 4.5: Synthetic data

Trend	$f_1(t) = 2t$
Seasonality	$f_2(t) = 10 \sin t$
Stationary Pattern	$f_3(t) = 5 \sin 5t$
Noise	$\delta(t) \sim \mathcal{N}(0, 1)$
$f(t) = f_1(t) + f_2(t) + f_3(t) + \delta(t)$	

4.2.3.7 Long-term Prediction with `FedKooL`

The prediction for non-stationary components, denoted as Y_i^{NS}, can be modeled using the Koopman evolution (4.32) as follows:

$$\begin{aligned} Y_i^{NS} &= (Z\Theta)\Lambda_i(\Theta^{-1}Z^T)X_i^{NS} \\ &= (Z\Theta)\Lambda_i(\Theta^{-1}Z^T)[x_1 \dots \quad x_n] \\ &= (Z\Theta)\Lambda_i(\Theta^{-1}Z^T)[x_1 \dots \quad (Z\Theta)\Lambda_i^{n-1}(\Theta^{-1}Z^T)x_1] \end{aligned}$$

Thus, given an observed state at a previous time point $t \in \{1, .., n\}$, a future state at time $h > t$ can be estimated as:

$$y_i^h = (Z\Theta)\Lambda_i^{h-t}(\Theta^{-1}Z^T)x_t = \sum\nolimits_{j\in\mathcal{S}}(Z\upsilon_j)\lambda_j^{h-t}(\psi_j Z^T)x_t.$$

The predictor $F_i(X_{NS})$ over p-steps for the non-stationary component X_i^{NS} is given by:

$$F_i(X_i^{NS}) = \sum\nolimits_{j\in\mathcal{S}}(Z\upsilon_j)\lambda_j^{p}(\psi_j Z^T)x_t. \tag{4.36}$$

As depicted in Fig. 4.8, the stationary MTS derived from the original non-stationary MTS retains similar patterns across all clients. To enhance the predictive capability of the model and mitigate Koopman evolution's weaknesses in long-term prediction [211] due to reliance on recursive forecasting, we deploy a FL system to develop a global predictive model g_w, parameterized by w, using distributed stationary data X_i^S to facilitate pattern learning. g_w is implemented as a neural network with multi-input and multi-output capabilities. For a sequence x_k from X_i^S, with a lookback window length B and forecasting window length H satisfying $n - B > H > B \geq 1$, g_w is defined as:

$$g_w : (x_B, ..., x_{B+L}) \mapsto (x_{B+L+1}, ..., x_{B+H}).$$

The detailed learning of g_w is implemented by FedSP, as presented in Algorithm 4. Finally, the prediction for non-stationary MTS data (X_i, Y_i) is generally formulated as:

$$\hat{Y}_i = g_w(X_i^S) + F_i(X_i^{NS}). \tag{4.37}$$

4.2.4 Experimental Results

In this section, we present numerical results and evaluations to validate the effectiveness of `FedKooL` in comparison with several state-of-the-art approaches in wireless traffic prediction.

Table 4.6: Dataset Statistics

Dataset	No.time series (M)	No.training samples	Missing data (%) (train / test)	No.testing samples	ADF test statistics
Milano-Call	10000	8640	22.32% / 21.76%	288	-13.91
Milano-SMS	10000	8640	16.91% / 14.27%	288	-14.74
Milano-Internet	10000	8640	0.13% / 0.46%	288	-12.56
Trentino-Call	6259	8640	19.21% / 10.99%	288	-11.18
Trentino-SMS	6259	8640	12.17%/ 3.95%	288	-7.87
Trentino-Internet	6259	8640	0.02% / 0.01%	288	-6.95

4.2.4.1 Experimental Settings

Dataset: We utilize datasets from Telecom Italia's Big Data Challenge [5], comprising Call Detail Records (CDR) from two distinct regions in Italy: Milan and Trento. The Milan dataset is segmented into a spatial grid of 10000 cells, while the Trento dataset contains 6259 cells. Each cell in the grid represents a base station, which captures a variety of telecommunication activities, including SMS, voice calls, and Internet services. The data is collected at 10-minute intervals from November 1, 2013, to January 1, 2014. For our analysis, we employ the first 8640 data points to train the prediction models, followed by a prediction phase for the next 288 time points, equivalent to two days. Prior to model training, we standardize the traffic data from each base station using its respective mean and standard deviation.

Evaluation Metrics: Model performance is assessed using RMSE and MAE, which are standard metrics for evaluating the accuracy of time series predictions. Lower values of these metrics indicate more accurate predictions. *Promotion* is the reduction achieved by `FedKooL` compared to the best baseline. The reported results are the average of three experimental runs for each algorithm, ensuring the reliability and reproducibility of the findings. We use the ADF test statistic to quantitatively assess stationarity. A lower ADF test statistic indicates a higher *degree of stationarity* [110]. Detailed statistics of the dataset are presented in Table 4.6, while an illustration of non-stationary is given in Fig. 4.12.

Baseline Selections: We compare our proposed wireless traffic prediction framework with four baseline methods including both conventional statistical approach and state-of-the-art machine learning methods for wireless traffic prediction in context of recursive multi-step forecast.

- SARIMA [232]: An extended version of ARIMA [104] that employs a statistical approach with seasonality incorporation for enhancing univariate time series forecasting.
- Fed-ConvLSTM [233]: The FedAvg adaptation for convolutional LSTM (ConvLSTM) model, which simultaneously captures spatial dependencies via convolution operations and temporal dependencies using LSTM gates.
- FedNST [110]: The FedAvg adaptation of NST, which leverages deep neural networks to separate non-stationary components using mean and variance, and employ attention mechanisms to predict stationary patterns.

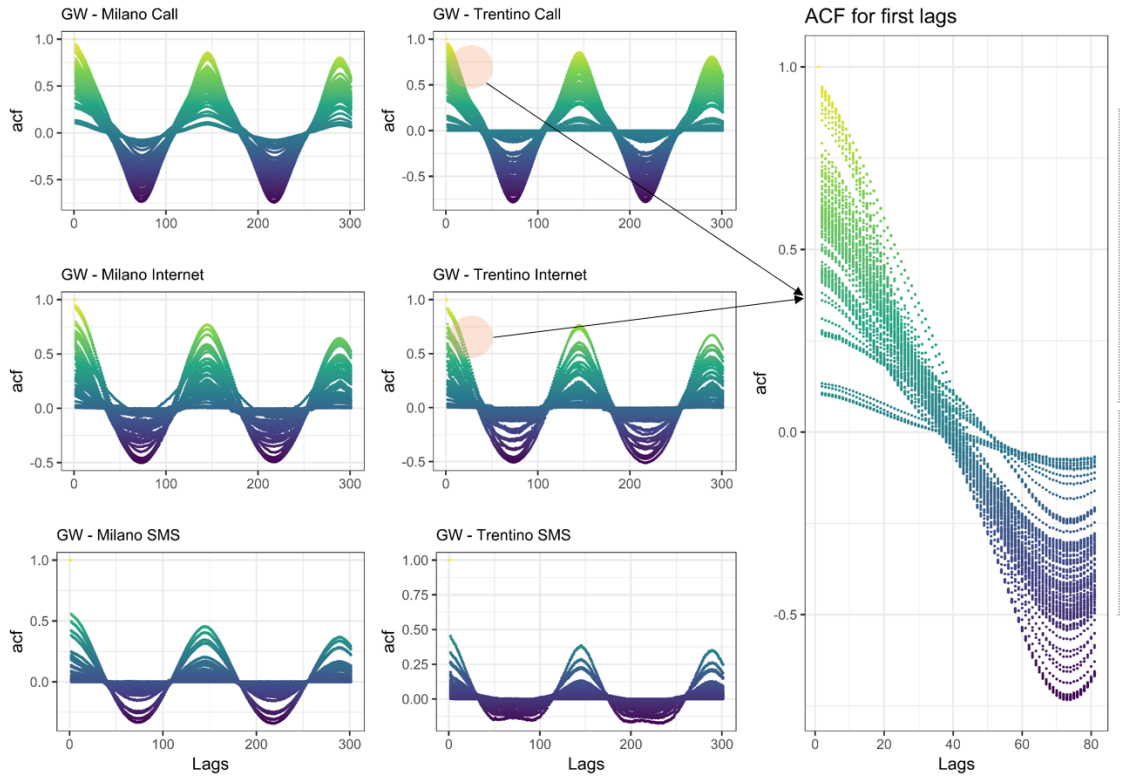

Figure 4.12: Seasonality in Milano and Trentino datasets. Trend can be verified due to large values of ACF for small lags and followed by a slow decrease, while the "scalloped" shape is due to the seasonality. Traffic data from a GW covering 100 base stations shows non-stationarity, with trends indicated by significant ACF values and seasonality reflected by "scalloped" shapes.

- FedDA [224]: A FL framework for wireless traffic prediction training model by dual attention-based aggregation.
- `FedKooL` without (w.o.) `FedSP` (ours): Using Koopman evolution (4.36).
- `FedKooL` with (w.) `FedSP` (ours): Combination of Koopman evolution (4.36) and pattern learning in (4.37).

Federated Learning Settings: In a HetNet deployment, base stations connect to a GW client, which collects data for wireless traffic prediction. To manage the diverse data distributions among GW clients, the training set is partitioned into $N = 10$ subsets, each containing $m = M/N$ local traffic time series. In all experiments, the rank-r of P is set to $r = 10$, with r optimized via grid search on the training set. The FL aggregation server employs FedAvg, and during each global communication round, 20% of the clients ($|\mathcal{S}^k| = 20\%$) are randomly selected for training. The number of global and local rounds are set to $T = 100$ and $C = 5$ for other methods. Specifically, FedSP utilizes a simple MLP with a lookback window size of $B = 48$, connected to two fully connected layers, each containing 64 hidden neurons mapping features to predictions.

Table 4.7: Italian Wireless Traffic Prediction Performance

Method	Milan						Trento					
	RMSE			MAE			RMSE			MAE		
	Call	Internet	SMS	Call	Internet	SMS	Call	Internet	SMS	Call	Internet	SMS
SARIMA	0.50±0.05	0.53±0.04	0.86±0.05	0.33±0.04	0.45±0.05	0.63±0.04	1.31±0.03	1.90±0.04	2.18±0.05	0.71±0.04	1.44±0.04	1.32±0.03
Fed-ConvLSTM	0.72±0.03	0.70±0.03	1.35±0.03	0.53±0.02	0.55±0.04	0.85±0.05	1.81±0.01	1.97±0.05	2.55±0.04	1.29±0.04	1.49±0.03	1.74±0.04
Fed-NST	0.62±0.03	0.63±0.05	1.31±0.02	0.36±0.03	0.44±0.03	0.74±0.01	1.42±0.04	1.55±0.03	2.87±0.04	0.86±0.01	1.07±0.05	1.69±0.04
FedDA	0.48±0.02	0.50±0.03	1.00±0.05	0.27±0.04	0.37±0.02	0.53±0.04	1.32±0.04	1.60±0.02	2.05±0.01	0.80±0.02	1.12±0.01	1.19±0.03
FedKooL w.o. FedSP	0.46±0.02	0.52±0.01	0.81±0.02	0.22±0.02	0.32±0.02	0.38±0.1	1.11±0.02	1.41±0.01	1.59±0.01	0.56±0.02	0.87±0.01	0.79±0.02
FedKooL w. FedSP	**0.41**±0.01	**0.27**±0.02	**0.68**±0.03	**0.16**±0.02	**0.13**±0.01	**0.29**±0.01	**0.67**±0.01	**0.93**±0.02	**1.11**±0.02	**0.30**±0.01	**0.38**±0.02	**0.48**±0.01
Promotion	+14.58%	+46%	+32%	+40.74%	+64.86%	+45.28%	+49.24%	+41.87%	+45.85%	+62.5%	+66.07%	+59.66%

Implementation Details: All experiments are conducted on a system equipped with an AMD Ryzen 3970X processor (64 cores), 256GB of RAM, and 8 NVIDIA GeForce RTX 3090 and 4090 GPUs. The models are trained using MSE loss and the Adam optimizer with a learning rate of 10^{-3}. Additionally, the batch size is uniformly set to 64. Each method shares the same hyper-parameter settings and FL aggregation server to maintain fairness in comparison.

4.2.4.2 Experimental Results

Table 4.8: FedKooL testing on Never-before-seen-data

Training Source		Testing Target											
		Milan						Trento					
		RMSE			MAE			RMSE			MAE		
		Call	Internet	SMS	Call	Internet	SMS	Call	Internet	SMS	Call	Internet	SMS
Milan	Call	0.41	0.28	0.69	0.16	0.14	0.28	0.69	0.96	1.13	0.31	0.41	0.49
	Internet	0.42	0.27	0.68	0.17	0.13	0.29	0.68	0.91	1.11	0.3	0.37	0.46
	SMS	0.41	0.28	0.68	0.16	0.14	0.29	0.68	0.95	1.12	0.3	0.4	0.48
Trento	Call	0.42	0.28	0.69	0.16	0.14	0.29	0.67	0.91	1.12	0.3	0.39	0.49
	Internet	0.42	0.27	0.68	0.17	0.13	0.29	0.68	0.93	1.12	0.3	0.38	0.48
	SMS	0.42	0.28	0.68	0.17	0.14	0.29	0.67	0.93	1.11	0.3	0.37	0.48

Forecasting Performance Comparison: The wireless traffic prediction performance of our proposed methods and baselines across two datasets is reported in Table 4.7, with the best results in bold and the second-best underlined. Our proposed approaches, including FedKooL and FedKooL w.o. FedSP, generally outperform other baselines for all types of wireless traffic in both datasets. Specifically, compared to the best baseline method, FedDA, FedKooL achieves significant RMSE improvements for call, SMS, and internet service traffic in the Milan dataset: 49.24%, 45.85%, and 41.87% respectively. In terms of MAE, FedKooL shows average gains of 50.28% for Milan and 62.95% for Trento. The performance improvements coincide with the degree of stationarity, with Trento showing significantly greater improvements than Milan. FedDA's inability to handle non-stationarity negatively impacts its performance in the presence of missing data. Surprisingly, FedNST, a state-of-the-art machine learning approach designed to handle non-stationary time series, underperforms in FL settings. This is because FL trains models on decentralized datasets with heterogeneous distributions, where each client has its own evolving mean and variance. Therefore, a global model predicting non-stationary characteristics based on mean and variance for all clients is unreasonable. FedKooL addresses this by modeling non-stationarity separately for each client. It also outperforms conventional machine learning methods

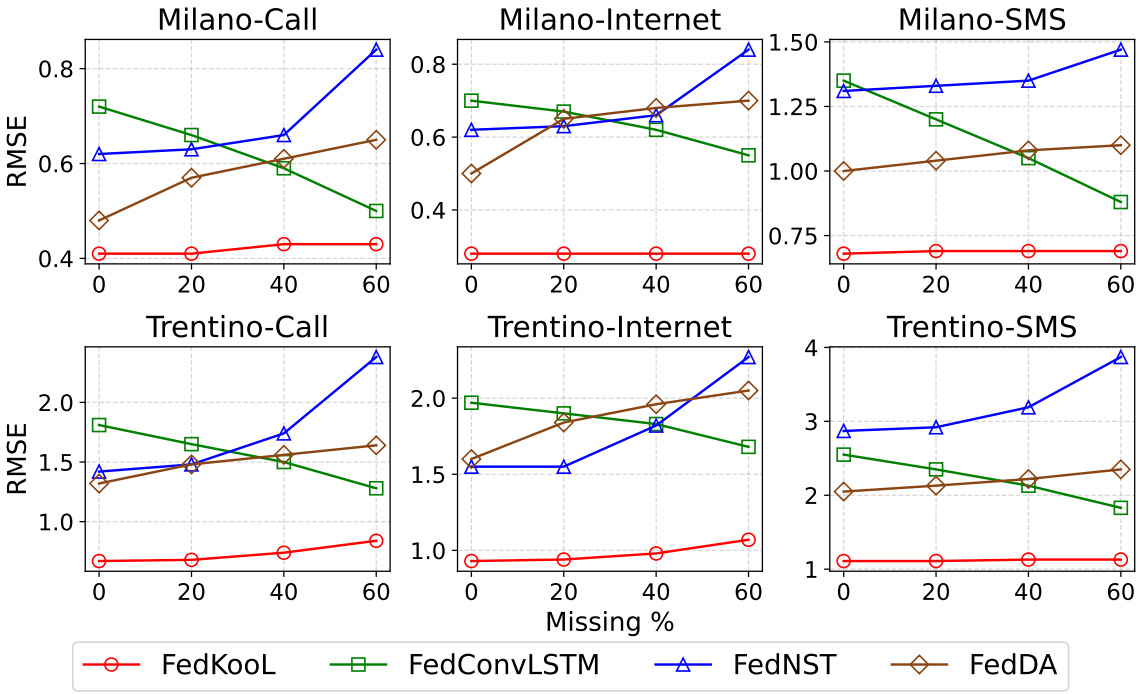

Figure 4.13: Impact of missing data on forecasting performance

like FedConvLSTM and traditional statistical methods like ARIMA. The success of `FedKooL` is attributed to Koopman learning, which captures both intra-spatial dependencies within a GW and inter-spatial relationships between GWs. However, performance of `FedKooL` is still inferior to `FedKooL` w.o `FedSP`, indicating the need for pattern learning models like `FedSP` to complement the limitations of using solely Koopman for prediction.

Missing value study: To reflect the nature of heterogeneous data in future network systems, we conducted experiments to examine the effect of missing values on the performance of wireless traffic prediction models. Initially, we injected additional missing data into the dataset, with the locations of the missing data determined using Pseudo-Random Number Generation [234]. We then evaluated the performance of the baseline model while varying the fraction of missing data. Our results in Fig. 4.13 indicate that, as the proportion of missing values increases, `FedKooL` consistently outperforms other benchmarks across all experiments. Notably, algorithms that account for spatial dependencies, such as `FedKooL` and FedConvLSTM, exhibit smaller forecasting errors compared to others. Conversely, models like FedDA and FedNST show high sensitivity to missing data, with their performance significantly deteriorating as the fraction of missing data increases. A particularly notable decline is observed in FedNST, which utilizes Fourier Transform to capture trends and seasonality. The presence of missing data exacerbates the issue of spectral leakage in Fourier Transform [214], leading to impaired performance in FedNST.

Convergence study: we analyzed the convergence rate of `FedKooL` compared to FedDA, a leading baseline known for efficiently handling heterogeneous data for FL wireless traffic prediction. By recording RMSE at each global iteration illustrated in Fig. 4.14, the results demonstrate that `FedKooL` converges faster than FedDA across

Table 4.9: Wall clock training time with 100 global iterations

Method	Average training time for each client (seconds)			
	Milan	*Inference time*	*Training time*	*Inference time*
Fed-ConvLSTM	0.72	0.70	1.35	0.53
Fed-NST	0.62	0.63	1.31	0.36
FedDA	0.49	0.50	1.03	0.23
FedKooL (ours)	0.41	0.27	0.68	0.16

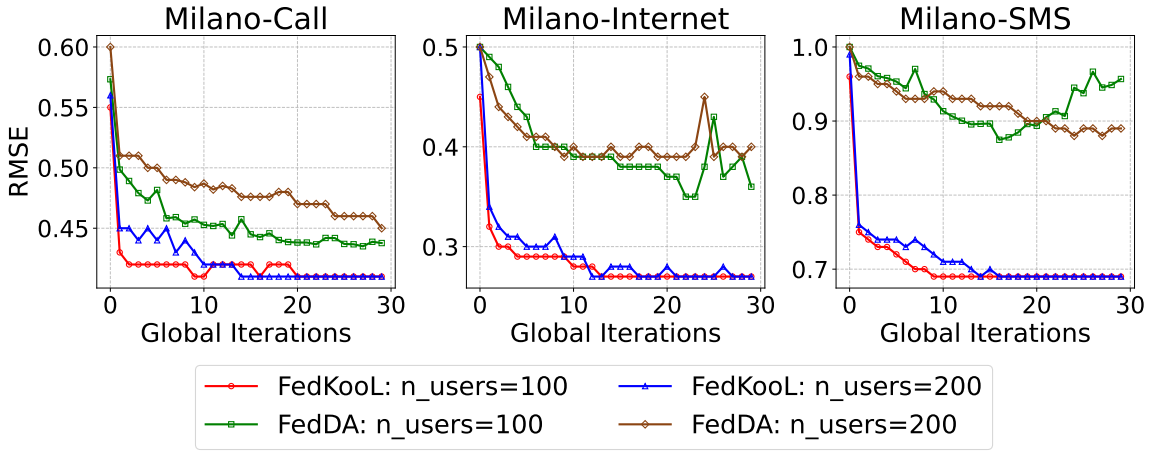

Figure 4.14: Convergence rate comparison: FedKooL vs FedDA

all experiments. When the number of clients increased, leading to less data coverage, FedDA shows a notably slower rate of convergence, whereas FedKooL maintains robust performance. Generally, FedKooL exhibits convergence after only 10 global iterations, while FedDA requires more than 30. Only in the Milano-internet dataset, which contains the smallest proportion of missing data, FedDA catch FedKooL's convergence rate. This finding reaffirms that FedDA remains highly sensitive to missing values, a common issue in future network systems. Furthermore, Table 4.10 shows that FedKooL requires the least training time compared to other baselines, while the inference time demonstrate capability of FedKooL serving in real-time.

Data coverage study: This section examines the efficiency of two models, FedKooL and FedKooL self-learning, which includes self-learning OMD and FedSP. We vary numbers of base stations covered by a GW in practical settings, where a GW can manage from a dozen to thousands of base stations [221–223]. Typically, the number of clients N is set to $\{10, 50, 100, 200\}$ for both the Milan and Trento datasets. Consequently, the number of base stations $m \in \{1000, 200, 100, 50\}$ for Milan and $m \in \{625, 125, 62, 31\}$ for Trento. As shown in Fig. 4.15, FedKooL demonstrates robustness while varying data coverage and scalability regarding client numbers. In contrast, the FedKooL self-learning struggles to model the dynamics with reduced coverage. It is only when a GW covers 1000 base stations that the FedKooL self-learning approach matches the performance of FedKooL. This similarity occurs because, as the amount of traffic data in a GW increases, the intra-relationship approximates the inter-relationship in the interconnected systems. However, a GW covering up to thousands of base stations is not a popular setting for HetNets.

Effectiveness on unseen data: As shown in Fig. 4.8, removing non-stationarity

Table 4.10: Average wall clock training/inference time (seconds) across clients per global iterations

Method	Milano		Trentino	
	Training Time	*Inference time*	*Training time*	*Inference time*
Fed-ConvLSTM	305.20	65.60	248.70	63.50
Fed-NST	154.99	40.92	158.22	35.00
FedDA	125.37	20.31	104.15	16.68
`FedKooL` w.o `FedSP`	1.54	0.39	1.21	0.11
`FedKooL` w. `FedSP`	112.31	1.19	98.47	0.91

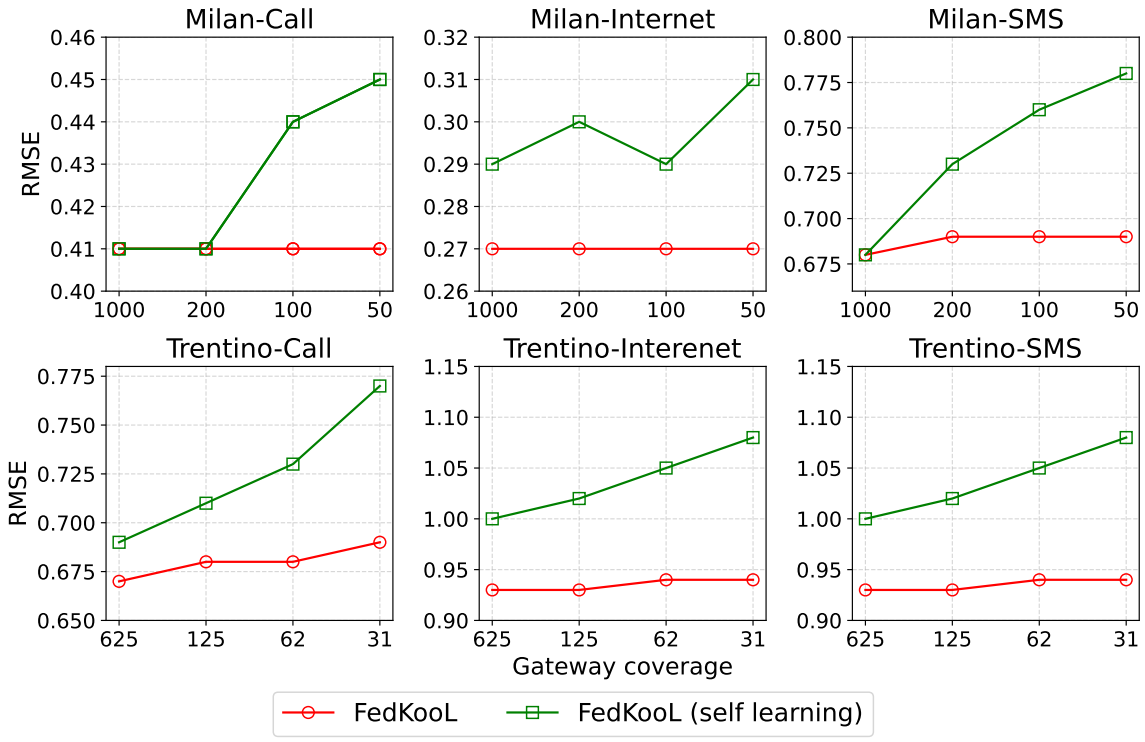

Figure 4.15: Impact of gateway coverage on forecasting performance

can reveal underlying user patterns. To investigate this, we conducted Koopman learning using `FedKooL` w.o `FedSP` for each dataset, then trained a `FedSP` model with the training source to predict all testing sources. The results in Table 4.8 indicate that similar patterns of stationary data appear across all datasets. Notably, there is a slight improvement when the `FedSP` model trained on the Milan dataset is tested on the Trento dataset, likely due to Milan's larger and more diverse training data. These findings demonstrate that a `FedSP`-trained model can be reused with Koopman operators for separating non-stationary components.

Chapter 5

Conclusion and Future Work

5.1 Summary

FL is an advanced paradigm well-suited to addressing the practical requirements of next-generation IoT networks. Specifically, FL offers significant advantages over centralized approaches, including enhanced data privacy and improved communication efficiency. Despite these benefits, the deployment of FL in real-world IoT applications faces several critical challenges. These include the presence of non-IID data, limited bandwidth availability, and the dynamic nature of IoT networks, which are further exacerbated by potential security threats.

This thesis introduces methodologies to tackle two essential challenges in FL: enabling intrusion detection and leveraging non-IID data in dynamic IoT environments. Grounded in the Manifold Hypothesis and Grassmann manifold theory, the proposed methods emphasize communication efficiency while accommodating the constrained computational resources of IoT devices. The first part of the thesis introduces an unsupervised FL algorithm specifically designed for anomaly detection in IoT networks. This method leverages the geometric properties of Grassmann manifolds to robustly model heterogeneous and non-IID data distributions while maintaining computational and communication efficiency. The second part delves into strategies for leveraging IoT data in dynamic and evolving environments. These approaches tackle core challenges such as non-stationary data distributions, bandwidth limitations, and unpredictable shifts in data patterns. By incorporating manifold-based techniques, the proposed solutions reveal intrinsic data structures, ensuring robustness, scalability, and adaptability in complex real-world IoT scenarios.

Specifically, in Chapter 3, we introduced `FedPCA`, a federated unsupervised anomaly detection framework leveraging Principal Component Analysis and the Alternating Directions Method of Multipliers. By proposing the algorithms `FedPE` in Euclidean space and `FedPG` on Grassmann manifolds, this chapter demonstrated the potential of FL for real-time intrusion detection while maintaining privacy and communication efficiency. Theoretical guarantees on convergence, even under sub-sampling, and empirical results on benchmark datasets validated the effectiveness of these methods in enhancing network resilience against security threats.

Building upon the communication efficiency principles established in Chapter 3, Chapter 4 extended the discussion to dynamic and non-stationary IoT data. This

chapter introduced `FedKoop`, a FL framework leveraging Koopman theory to model spatio-temporal dependencies in non-stationary multivariate time series. By incorporating gradient movement on Grassmann manifolds, `FedKoop` achieved superior forecasting accuracy with reduced computational and memory requirements, making it suitable for large-scale cyber-physical systems. Furthermore, the development of `FedKooL` for wireless traffic prediction showcased an application of the Koopman-based framework. By optimizing low-rank modeling on Grassmann manifolds and leveraging techniques such as Rockafellar's envelope, `FedKooL` achieved substantial improvements in predictive accuracy and efficiency, as evidenced by empirical results on real-world datasets.

5.2 Future Work

This thesis establishes a foundation for applying mathematical frameworks such as the Manifold Hypothesis, Grassmann manifolds, Federated PCA, consensus ADMM and Koopman theory to FL. Building on the methodologies developed in Chapters 3 and 4, future research can expand these approaches to tackle the unique challenges of training Large Language Models (LLMs) in FL. As LLMs have become indispensable in modern natural language processing (NLP), addressing issues such as communication efficiency, privacy preservation, and dynamic data distributions is essential. Integrating manifold-based methods, dimensionality reduction techniques, and dynamic modeling frameworks into LLM training holds great promise for improving scalability, adaptability, and resilience.

Manifold Hypothesis, which postulates that high-dimensional data often resides on low-dimensional manifolds, serves as a cornerstone for optimizing LLMs in FL. Grassmann manifolds, in particular, offer a powerful mathematical framework for representing high-dimensional updates or embeddings generated during LLM training. Encoding these updates as points on Grassmann manifolds enables efficient subspace modeling, significantly reducing communication overhead. This approach is particularly advantageous in federated settings where devices may have limited computational and communication resources.

Federated PCA adds another layer of enhancing the efficiency and privacy of LLM training in federated environments. By extracting the most informative principal components from model updates or embeddings, Federated PCA reduces dimensionality while preserving the essential features necessary for effective learning. This not only minimizes the volume of data exchanged across devices but also ensures the integrity of the training process, even under constrained bandwidth conditions [29]. One of the key advantages of Federated PCA lies in its ability to enhance communication efficiency during LLM training. By compressing updates into their principal components, it significantly reduces the size of transmitted data while maintaining high fidelity. This makes it particularly advantageous for FL setups where devices often have limited computational and communication resources. Such optimization enables the training of large-scale LLMs across diverse nodes without imposing excessive demands on network infrastructure. Moreover, Federated PCA ensures that these compressed representations focus on the most meaningful dimensions of the data, which facilitates faster convergence and better alignment of distributed model parameters [235]. Beyond communication efficiency, Federated PCA offers robust privacy-preserving capabilities.

By operating on principal components rather than raw data, it inherently obfuscates sensitive information, preventing adversaries from reconstructing private inputs. This makes Federated PCA a valuable tool for FL applications involving LLMs trained on sensitive data, such as personalized language models or healthcare-related text analysis. The ability to transmit only the most essential features further reduces the risk of information leakage, aligning with privacy regulations and user trust expectations [29]. Federated PCA also plays a significant role in addressing challenges related to non-IID data across federated nodes. In such heterogeneous environments, identifying shared global features becomes critical for effective model aggregation. Federated PCA excels in this context by isolating common structures within the data, enabling more robust and consistent model performance across diverse devices. By focusing on the primary directions of variance, Federated PCA mitigates the impact of non-IID data, ensuring that LLMs can learn generalized representations [236]. Additionally, Federated PCA facilitates the interpretability and adaptability of LLMs. By analyzing the extracted principal components, researchers can uncover latent structures and patterns in embeddings or updates, offering insights into how LLMs process and adapt to diverse datasets. This capability is particularly beneficial for domain adaptation, where task-specific components can be identified and fine-tuned efficiently. For example, in specialized applications such as medical or legal text analysis, Federated PCA can streamline the adaptation process, reducing computational overhead while maintaining task-specific accuracy [237]. Lastly, Federated PCA enhances the robustness of LLMs against adversarial attacks and noisy data. By projecting updates or embeddings onto principal components, it filters out irrelevant or noisy features, making LLMs less susceptible to adversarial perturbations. This noise-resilient property is crucial in ensuring reliable performance in real-world federated settings, where data quality can vary significantly [238].

Another critical challenge in FL is the dynamic nature of distributed language data, particularly in non-stationary environments where data distributions evolve over time. Here, Koopman theory offers a transformative framework for modeling temporal dynamics. By learning Koopman operators that capture spatio-temporal dependencies in distributed LLM parameters or embeddings, FL systems can predict and adapt to shifts in data distributions. This approach ensures that LLMs remain robust and accurate, even in the face of evolving data. Additionally, incorporating gradient movement on Grassmann manifolds enhances the efficiency of learning Koopman operators, further optimizing scalability. This synergy between dynamic modeling and manifold-based optimization provides a pathway for improving LLM performance in environments characterized by continual change [239].

In conclusion, this thesis provides a robust starting point for the application of mathematical frameworks to FL. Future research could extend these methodologies to explore novel directions, such as low-rank subspace modeling for LLMs, improved adaptive mechanisms for dynamic federated environments, and deeper integration of manifold learning with neural architectures. By advancing these ideas, the research community can unlock the full potential of LLMs in FL, enabling transformative innovations in NLP and beyond.

Bibliography

[1] T.-A. Nguyen *et al.*, "Federated pca on grassmann manifold for iot anomaly detection," *IEEE/ACM Transactions on Networking*, vol. 32, no. 5, pp. 4456–4471, 2024.

[2] T.-A. Nguyen, J. He, L. T. Le, W. Bao, and N. H. Tran, "Federated pca on grassmann manifold for anomaly detection in iot networks," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[3] T.-A. Nguyen, T. D. Nguyen, L. T. Le, C. T. Dinh, and N. H. Tran, "On the generalization of wasserstein robust federated learning," *arXiv preprint arXiv:2206.01432*, 2022.

[4] A. Trindade, "ElectricityLoadDiagrams20112014," UCI Machine Learning Repository, 2015.

[5] G. Barlacchi, M. D. Nadai, R. Larcher, A. Casella, and C. C. et al., "A multi-source dataset of urban life in the city of milan and the province of trentino," *Scientific Data*, vol. 2, 2015.

[6] R. Mushtaq, "Augmented dickey fuller test," 2011.

[7] D. Reinsel, J. Gantz, and J. Rydning, "The Digitization of the World from Edge to Core," p. 28, 2018.

[8] P. Li *et al.*, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, no. C, pp. 76–85, Sep. 2017.

[9] "Strava has published details about secret military bases, and an australian was the first to know," ABC News, 2018. [Online]. Available: https://www.abc.net.au/news/science/2018-01-29/strava-heat-map-shows-military-bases-and-supply-routes/9369490

[10] "California consumer privacy act (ccpa)," California Privacy Protection Agency, 2018, amended by the California Privacy Rights Act (CPRA) in 2020. [Online]. Available: http://cppa.ca.gov/regulations/

[11] "General personal data protection (lgpd)," Brasil, 2018. [Online]. Available: https://lgpd-brazil.info/

[12] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[13] X. Wang *et al.*, "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[15] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv:1602.05629*, Feb. 2017.

[16] Google LLC. (2025) Google home – manage your smart home. Accessed: 2025-08-03. [Online]. Available: https://home.google.com/

[17] L. Da Xu, W. He, and S. Li, "Internet of things for healthcare: A comprehensive survey," *IEEE Reviews in Biomedical Engineering*, vol. 12, pp. 4–20, 2019.

[18] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning in iot: Recent advances and future challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 803–12 823, 2021.

[19] T. Taleb *et al.*, "Multi-access edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.

[20] B. Varghese and R. Buyya, "Challenges and opportunities in edge computing," *IEEE Internet Computing*, vol. 21, no. 2, pp. 20–27, 2016.

[21] M. Chiang and T. Zhang, "Fog computing: Mitigating insider data theft attacks in cloud computing," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–53, 2016.

[22] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[23] B. Koonce and B. Koonce, "Mobilenetv3," *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pp. 125–144, 2021.

[24] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] OpenAI, "Gpt-4o mini: Advancing cost-efficient intelligence," 2024, accessed: 2024-10-20. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[26] P. Kairouz, H. B. McMahan *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[27] Y. Tan *et al.*, "Fedproto: Federated prototype learning over heterogeneous devices," *arXiv preprint arXiv:2105.00243*, vol. 3, 2021.

[28] Y. Zhao *et al.*, "Federated learning with non-iid data," in *arXiv preprint arXiv:1806.00582*, 2018.

[29] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

[30] S. P. Karimireddy *et al.*, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning (ICML)*, 2020, pp. 5132–5143.

[31] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[32] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 7611–7623, 2020.

[33] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in neural information processing systems*, vol. 33, pp. 21 394–21 405, 2020.

[34] L. Collins, H. Hassani, and K. Ramchandran, "Exploiting low-rank structure for federated learning via proximal gradient descent," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021, pp. 2089–2099.

[35] D. Yin, Y. Chen, S. Kannan, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning (ICML)*, 2018, pp. 5650–5659.

[36] C. Xie, O. Koyejo, and P. Gupta, "Zeno: Byzantine-robust stochastic gradient descent," in *International Conference on Machine Learning (ICML)*, 2019, pp. 6893–6901.

[37] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[38] J. Konečný *et al.*, "Federated Learning: Strategies for Improving Communication Efficiency," *arXiv:1610.05492*, Oct. 2017.

[39] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *arXiv:1908.07873*, Aug. 2019.

[40] T. S. Brisimi *et al.*, "Federated learning of predictive models from federated Electronic Health Records," *International journal of medical informatics*, vol. 112, pp. 59–67, 2018.

[41] V. Mothukuri *et al.*, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X20329848

[42] Z. Zhou *et al.*, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[43] L. Cui *et al.*, "A survey on application of machine learning for Internet of Things," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, pp. 1399–1417, Aug. 2018.

[44] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[45] V. Mothukuri *et al.*, "A survey on security and privacy issues in federated learning," *Journal of Network and Computer Applications*, 2021.

[46] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2021.

[47] W. Y. B. Lim *et al.*, "Federated learning for time series data: System design and model optimization," *ACM Computing Surveys*, vol. 55, no. 1, pp. 1–39, 2022.

[48] Q. Luo, Y. Ding, J. Zhang, Q. Wu, and W. Dou, "Handling concept drift in federated learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[49] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[50] J. Yoon, E. Kim, and E. Yang, "Federated continual learning with weighted inter-client transfer," in *International Conference on Machine Learning (ICML)*, 2021, pp. 12 073–12 086.

[51] Q. Zhang, Y. Bai, H. Yu, Y. Liu, and Y. Zhang, "Spatio-temporal graph neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[52] Y. Li, H. Yu, M. Tan, and Q. Yang, "Spatiofed: Spatio-temporal personalized federated learning for traffic forecasting," *IEEE Transactions on Mobile Computing*, 2024.

[53] Z. Chen, X. Wang, Y. Huang, X. Chen, and X. Lin, "Fedevent: Towards event-aware federated learning in real-time iot systems," in *Proceedings of the 41st IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2023, pp. 1621–1631.

[54] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with meta-learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 12 230–12 241.

[55] L. Collins *et al.*, "Exploiting shared representations for personalized federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 30 228–30 239.

[56] H. Wu, X. Xu, Y. Chen, J. Zhang, and Z. Lin, "Handling long-tailed data in federated learning with class rebalancing and augmentation," in *Proceedings of the 30th ACM International Conference on Multimedia*. ACM, 2022, pp. 3196–3205.

[57] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, vol. 29, no. 4, pp. 983–1049, 2016.

[58] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[59] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," in *Science*, vol. 313, no. 5786, 2006, pp. 504–507.

[60] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[61] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, and E. C. Strinati, "Federated learning with cooperating devices: A consensus approach for massive iot networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4641–4654, 2020.

[62] B. Zong *et al.*, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations (ICLR)*, 2018.

[63] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[64] K. Lee, H. Lee, K. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[65] Y. Zhu *et al.*, "Deep anomaly detection with self-supervised learning and kernel density estimation," *Pattern Recognition*, vol. 119, p. 108096, 2021.

[66] M. Taheri and A. H. Gandomi, "Data-driven anomaly detection in iot systems using one-class transfer learning," *Expert Systems with Applications*, vol. 159, p. 113594, 2020.

[67] P.-A. Absil, R. Mahony, and R. Sepulchre, "Riemannian geometry of grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematica*, vol. 80, no. 2, pp. 199–220, 2004.

[68] A. Edelman, T. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[69] Z. Huang and L. Van Gool, "Grassmannian learning: Embedding geometry awareness in shallow and deep learning," *arXiv preprint arXiv:1906.04399*, 2019.

[70] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dynamics*, vol. 41, no. 1-3, pp. 309–325, 2005.

[71] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS One*, vol. 11, no. 2, p. e0150171, 2016.

[72] J. Konečný, B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2015.

[73] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

[74] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1-2, pp. 1–210, 2021.

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

[76] W. Lim, J. Kim, and H. Lee, "Federated learning for internet of things: Recent advances, challenges, and future directions," *Sensors*, vol. 22, no. 12, p. 4582, 2022.

[77] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[78] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4424–4434.

[79] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[80] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," in *IEEE Transactions on Neural Networks and Learning Systems*. IEEE, 2019.

[81] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *arXiv preprint arXiv:1712.01887*, 2020.

[82] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[83] X. Chen, J. Liu, H. Su, and R. Xia, "Robust federated learning with byzantine clients," *arXiv preprint arXiv:2003.11453*, 2020.

[84] K. Shaloudegi *et al.*, "Federated continual learning in non-stationary iot environments," *arXiv preprint arXiv:2212.12121*, 2022.

[85] J. Zhang, Y. Zheng, and D. Qi, "Spatio-temporal graph neural networks: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[86] H. Li, J. Chen, and W. Zhao, "Federated learning for spatio-temporal data: Challenges and approaches," *arXiv preprint arXiv:2401.01234*, 2024.

[87] T. Li, M. Li, S. Li, L. Song, and V. S. Wang, "Feddyn: Federated learning with dynamic regularization," *arXiv preprint arXiv:2407.07421*, 2024.

[88] N. Neshenko, E. Bou-Harb, J. Crichigno, F. Salahdine, and A. Ayesh, "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[89] S. Srinivas *et al.*, "Knowledge transfer via distillation of activation boundaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5058–5067.

[90] J. Cheng, W. Li, F. Xu, and C. Zhao, "A survey of iot security: Attacks, challenges, and solutions," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7850–7872, 2019.

[91] Y. Meidan *et al.*, "Detection of unauthorized iot devices using machine learning techniques," in *Proceedings of the IEEE ICC*, 2018, pp. 1–7.

[92] K. Anthi, S. Sajadmanesh, M. Mohammadi, A. Khosravi, and S. Nahavandi, "Machine learning for anomaly detection in iot networks: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9767–9783, 2019.

[93] H. H. Pajouh, F. Guitton, P. Dandachi, and M. Debbabi, "A lightweight deep learning algorithm for iot intrusion detection," *Future Generation Computer Systems*, vol. 101, pp. 914–924, 2019.

[94] A. Zolanvari, M. Hitchens, J. Batalla, Y. Jararweh, and I. Alsmadi, "Network intrusion detection for iot security based on learning techniques," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2019.

[95] W. Wang, Q. Zhu, Y. Chen, Y. Zhang, and Y. Liu, "Iot intrusion detection system based on deep learning," *IEEE Access*, vol. 9, pp. 158 410–158 422, 2021.

[96] T. Qiu, J. Yang, R. Yu, S. Wang, and G. Luo, "Federated learning for iot: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8012–8028, 2021.

[97] H. Nguyen, A. Pathak, I. Papapanagiotou, and P. Papadopoulos, "Federated learning for iot intrusion detection," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[98] L. Wang, L. Chen, J. Li, B. Liu, and T. Huang, "Anomaly detection in iot networks based on federated learning with neural networks," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1681–1693, 2022.

[99] V. Mothukuri *et al.*, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2022.

[100] D. J. Cook and S. A. Das, "How smart are our environments? an updated look at the state of the art," *Pervasive and Mobile Computing*, vol. 70, p. 101260, 2020.

[101] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[102] K. Tanaka, *Time series analysis: nonstationary and noninvertible distribution theory*. John Wiley & Sons, 2017, vol. 4.

[103] B. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting. arxiv 2019," *arXiv preprint arXiv:1905.10437*, 2019.

[104] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc., 1990.

[105] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[106] H. Zhou *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[107] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[108] T. Kim *et al.*, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *International Conference on Learning Representations*, 2021.

[109] W. Fan *et al.*, "Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7522–7529.

[110] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, 2022.

[111] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[112] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 10, 2017.

[113] R. Wang, Y. Dong, S. O. Arik, and R. Yu, "Koopman neural operator forecaster for time-series with temporal distributional shifts," in *The Eleventh International Conference on Learning Representations*, 2022.

[114] Y. Liu, C. Li, J. Wang, and M. Long, "Koopa: Learning non-stationary time series dynamics with koopman predictors," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[115] S. Saha *et al.*, "Fog-based federated time series forecasting for iot," *Springer Journal of Network and Systems Management*, 2024.

[116] H. Zhang *et al.*, "Mo-ensfts: Embedding-based non-stationary fuzzy time series for iot applications," *Neural Computing and Applications*, 2023.

[117] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[118] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[119] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[120] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[121] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[122] P. P. Liang, T. Liu, X. Zhan, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 20 898–20 908.

[123] H. Sato, H. Kasai, and B. Mishra, "Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport," *SIAM Journal on Optimization*, vol. 29, no. 2, pp. 1444–1472, 2019.

[124] J. Zhang, G. Zhu, R. W. H. Jr., and K. Huang, "Grassmannian learning: Embedding geometry awareness in shallow and deep learning," *CoRR*, vol. abs/1808.02229, 2018.

[125] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Statistical analysis on stiefel and grassmann manifolds with applications in computer vision," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, June 2008.

[126] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical computations on grassmann and stiefel manifolds for image and video-based recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2273–2286, 2011.

[127] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *Proc. of International Conference on Machine Learning (ICML)*, Helsinki, July 2008.

[128] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Providence, June 2011.

[129] L. S. de Souza, H. Hino, and K. Fukui, "3d object recognition with enhanced grassmann discriminant analysis," in *Proc. of Asian Conference on Computer Vision (ACCV)*, Taipei, November 2016.

[130] H. E. Cetingul and R. Vidal, "Intrinsic mean shift for clustering on stiefel and grassmann manifolds," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Miami, June 2009.

[131] Q. Wang, J. Gao, and H. Li, "Grassmannian manifold optimization assisted sparse spectral clustering," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, July 2017.

[132] W. Dai, E. Kerman, and O. Milenkovic, "A geometric approach to low-rank matrix completion," *IEEE Trans. on Information Theory*, vol. 58, no. 1, pp. 237–247, 2012.

[133] N. Boumal and P.-A. Absil, "Low-rank matrix completion via preconditioned optimization on the grassmann manifold," *Linear Algebra and its Applications*, vol. 475, pp. 200–239, 2015.

[134] B. M. Hochwald and T. L. Marzetta, "Unitary space-time modulation for multiple-antenna communications in rayleigh flat fading," *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 543–564, 2000.

[135] L. Zheng and D. N. C. Tse, "Communication on the grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel," *IEEE Trans. on Information Theory*, vol. 48, no. 2, pp. 359–383, 2002.

[136] Y. Du, G. Zhu, J. Zhang, and K. Huang, "Automatic recognition of space-time constellations by learning on the grassmann manifold," Submitted to IEEE Trans. on Signal Processing, 2018, available: https://arxiv.org/abs/1804.03593.

[137] K. Hall and T. Hofmann, "Learning curved multinomial subfamilies for natural language processing and information retrieval," in *Proc. of International Conference on Machine Learning (ICML)*, San Francisco, June 2000.

[138] F. Monti *et al.*, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, July 2017.

[139] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. of IEEE International Conference on Computer Vision Workshop (ICCVW)*, Santiago, December 2015.

[140] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, "Exponential expressivity in deep neural networks through transient chaos," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, Barcelona, December 2016.

[141] Z. Huang, J. Wu, and L. V. Gool, "Building deep networks on grassmann manifolds," in *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, New Orleans, February 2018.

[142] S. Herath, M. Harandi, and F. Porikli, "Learning an invariant hilbert space for domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3845–3854.

[143] D. Acar *et al.*, "Federated learning based on dynamic regularization," *Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1–39, 2021.

[144] T. Yang, B. Marfoq, and A. Bellet, "Federated learning with local submodels," in *International Conference on Machine Learning (ICML)*, 2021.

[145] N. Tripuraneni, P. Jain, L. Bottou, and M. I. Jordan, "Provable federated learning: From local sgd to byzantine-robustness," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[146] P. Jawanpuria, P. Bhargav, R. Balan, B. Mishra, and B. Ravindran, "Learning low-rank shared representations in federated settings," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–38, 2022.

[147] S. Mukherjee, P. Jain, K. Kawaguchi, and A. Nori, "Provably efficient communication in distributed non-convex optimization via low-rank subspace projections," in *NeurIPS*, 2022.

[148] S. Singhal, A. Shilton, T. Le, and D. Nguyen, "Federated low-rank learning with personalization layers," in *International Conference on Machine Learning (ICML) Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2021.

[149] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *NeurIPS*, 2020.

[150] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[151] J. Giesen and R. Zimmer, "A distributed admm algorithm for solving large-scale linear inverse problems," *Journal of Computational and Applied Mathematics*, vol. 296, pp. 413–427, 2016.

[152] M. Hong and Z.-Q. Luo, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, 2015.

[153] T.-H. Chang, M. Hong, and X. Wang, "Asynchronous distributed admm for consensus optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3139–3153, 2016.

[154] Y. Zhang and M. Burger, "The alternating direction method of multipliers for machine learning," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 137–148, 2016.

[155] J. Feng and M. Tan, "Admm in signal processing and machine learning: a survey," *IEEE Access*, vol. 8, pp. 96 016–96 044, 2020.

[156] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 74, no. 2, pp. 868–895, 2018.

[157] Y. Wang and W. Yin, "Admm for nonconvex optimization: Recent advances and challenges," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 277–304, 2019.

[158] ——, "Global convergence of admm for nonconvex problems with nonlinear equality constraints," *SIAM Journal on Optimization*, vol. 29, no. 1, pp. 306–332, 2019.

[159] C. Chen, B. He, Y. Ye, and X. Yuan, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, vol. 155, pp. 105–144, 2016.

[160] J. Sun, H. Li, and Z. Xu, "Admm-nets: A deep learning approach for compressive sensing mri," *Advances in neural information processing systems*, vol. 29, 2016.

[161] X. Yuan, S. Liu, and X. Zhou, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[162] Q. Ling and A. Ribeiro, "Communication-efficient distributed optimization in networks via admm," in *IEEE Transactions on Signal Processing*, vol. 61, no. 10, 2013, pp. 2555–2560.

[163] S. Wang, R. Mathews, and H. V. Poor, "Decentralized federated learning: A segmented admm approach," *IEEE Transactions on Signal Processing*, vol. 70, pp. 293–306, 2022.

[164] N. Shlezinger and Y. C. Eldar, "Communication-efficient admm with quantized gradients," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 126–139, 2020.

[165] B. He and X. Yuan, "An adaptive penalty approach for consensus optimization with admm," *Mathematical Programming*, vol. 172, pp. 349–375, 2018.

[166] P. Kairouz, Z. Chen, and B. McMahan, "Hierarchical federated learning via admm," in *NeurIPS FL Workshop*, 2021.

[167] J. Zhang, B. He, and X. Yuan, "Accelerated admm for nonconvex optimization," *Computational Optimization and Applications*, vol. 72, pp. 1–29, 2019.

[168] S. Zheng and J. Kwok, "Stochastic variance-reduced admm," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[169] S. Pu and A. Nedich, "Decentralized gradient tracking for composite objectives: Finite-time analysis and admm connection," *IEEE Transactions on Automatic Control*, 2021.

[170] M. Zhao and R. Jin, "Differentially private admm-based federated learning with client-level privacy," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2225–2238, 2022.

[171] Z. Afroz, G. Shafiullah, T. Urmee, and G. Higgins, "Modeling techniques used in building hvac control systems: A review," *Renewable and sustainable energy reviews*, vol. 83, pp. 64–84, 2018.

[172] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[173] X. Yang, P. Zhao, X. Zhang, J. Lin, and W. Yu, "Toward a gaussian-mixture model-based detection scheme against data integrity attacks in the smart grid," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 147–161, 2016.

[174] D. Chou and M. Jiang, "A survey on data-driven network intrusion detection," *ACM Comput. Surv.*, vol. 54, no. 9, oct 2021. [Online]. Available: https://doi.org/10.1145/3472753

[175] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:209916786

[176] H. Zenati, C.-S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient gan-based anomaly detection," *ArXiv*, vol. abs/1802.06222, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:3305321

[177] H. Zenati, M. Romain, C. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International Conference on Data Mining (ICDM)*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2018, pp. 727–736. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICDM.2018.00088

[178] L. Ruff *et al.*, "A Unifying Review of Deep and Shallow Anomaly Detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, May 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9347460/

[179] C. Pascoal *et al.*, "Robust feature selection and robust pca for internet traffic anomaly detection," in *Proceedings IEEE INFOCOM*, 2012, pp. 1755–1763.

[180] T. Yu, X. Wang, and A. Shami, "Recursive principal component analysis-based data outlier detection and sensor data aggregation in iot systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2207–2216, 2017.

[181] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[182] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Computer Communications*, vol. 35, no. 7, pp. 772–783, 2012.

[183] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *Eai Endorsed Transactions on Security and Safety*, vol. 3, no. 9, p. e2, 2016.

[184] A. Belenguer, J. Navaridas, and J. A. Pascual, "A review of federated learning in intrusion detection systems for iot," *arXiv:2204.12443*, 2022.

[185] J. Zhang, G. Zhu, R. W. Heath Jr, and K. Huang, "Grassmannian learning: Embedding geometry awareness in shallow and deep learning," *arXiv preprint arXiv:1808.02229*, 2018.

[186] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[187] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. d. Hartog, "Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, 2022.

[188] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.

[189] M. Udell, C. Horn, R. Zadeh, and S. Boyd, "Generalized low rank models," *Foundations and Trends® in Machine Learning*, vol. 9, no. 1, pp. 1–118, 2016.

[190] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, 2011.

[191] T.-A. Nguyen, J. He, L. T. Le, W. Bao, and N. H. Tran, "Federated pca on grassmann manifold for anomaly detection in iot networks," 2023.

[192] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Computer Science*, vol. 167, pp. 636–645, 2020.

[193] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016. [Online]. Available: https://doi.org/10.1137/140990309

[194] L. Lin, B. Saparbayeva, M. M. Zhang, and D. B. Dunson, "Accelerated algorithms for convex and non-convex optimization on manifolds," *arXiv preprint arXiv:2010.08908*, 2020.

[195] S. Bubeck *et al.*, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[196] A. Paszke *et al.*, "Automatic differentiation in pytorch," in *NIPS 2017 Workshop on Autodiff*, 2017. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ

[197] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[198] C. Meng, S. Rambhatla, and Y. Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021.

[199] J. H. Stock and M. W. Watson, "Vector autoregressions," *Journal of Economic perspectives*, vol. 15, no. 4, pp. 101–115, 2001.

[200] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[201] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[202] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International journal of forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[203] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[204] S. L. Brunton and J. N. Kutz, *Modern Koopman theory for dynamical systems*. SIAM, 2022.

[205] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[206] H. Arbabi and I. Mezić, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 4, pp. 2096–2126, 2017.

[207] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "Data-driven approximations of dynamical systems operators for control," in *American Control Conference (ACC)*. IEEE, 2015, pp. 4606–4613.

[208] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[209] A. Wynn, D. Pearson, B. Ganapathisubramani, and P. J. Goulart, "Optimal mode decomposition for unsteady flows," *Journal of Fluid Mechanics*, vol. 733, pp. 473–503, 2013.

[210] S. Tirunagari, S. Kouchaki, N. Poh, M. Bober, and D. Windridge, "Dynamic mode decomposition for univariate time series: analysing trends and forecasting," 2017.

[211] S. Pan, E. Kaiser, B. M. de Silva, J. N. Kutz, and S. L. Brunton, "Pykoopman: a python package for data-driven approximation of the koopman operator," *arXiv preprint arXiv:2306.12962*, 2023.

[212] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 95–104.

[213] Facebook, "Prophet," 2020. [Online]. Available: https://facebook.github.io/prophet/

[214] M. Schonewille, A. Klaedtke, and A. Vigner, "Anti-alias anti-leakage fourier transform," in *SEG International Exposition and Annual Meeting*. SEG, 2009, pp. SEG–2009.

[215] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?" *Nature Electronics*, vol. 3, no. 1, pp. 20–29, 2020.

[216] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.

[217] M. Yao, M. Sohul, V. Marojevic, and J. H. Reed, "Artificial intelligence defined 5g radio access networks," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 14–20, 2019.

[218] J. Wang *et al.*, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *IEEE INFOCOM 2017-IEEE conference on computer communications*. IEEE, 2017, pp. 1–9.

[219] X. Gou and X. Zhang, "Telecommunication traffic forecasting via multi-task learning," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 859–867.

[220] G. P. Nason, "Stationary and non-stationary time series," 2006.

[221] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "5g backhaul challenges and emerging research directions: A survey," *IEEE Access*, vol. 4, pp. 1743–1766, 2016.

[222] Z. Md. Fadlullah, F. Tang, B. Mao, J. Liu, and N. Kato, "On intelligent traffic control for large-scale heterogeneous networks: A value matrix-based deep learning approach," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2479–2482, 2018.

[223] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, "Hethetnets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2252–2265, 2015.

[224] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *IEEE INFOCOM 2021-IEEE conference on computer communications*.

[225] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389–1401, 2019.

[226] H. Nan, X. Zhu, and J. Ma, "Mstl-gltp: A global–local decomposition and prediction framework for wireless traffic," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5024–5034, 2023.

[227] O. Azencot, N. B. Erichson, V. Lin, and M. Mahoney, "Forecasting sequential data using consistent koopman autoencoders," in *International Conference on Machine Learning*. PMLR, 2020, pp. 475–485.

[228] M. J. Colbrook, I. Mezić, and A. Stepanenko, "Limits and powers of koopman learning," *arXiv preprint arXiv:2407.06312*, 2024.

[229] V. Gallese, "The'shared manifold'hypothesis. from mirror neurons to empathy," *Journal of consciousness studies*, vol. 8, no. 5-6, pp. 33–50, 2001.

[230] A. N. Gorban and I. Y. Tyukin, "Blessing of dimensionality: mathematical foundations of the statistical physics of data," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2118, p. 20170237, 2018.

[231] M. Blondel and V. Roulet, "The elements of differentiable programming," *arXiv preprint arXiv:2403.14606*, 2024.

[232] Y. Shu, M. Yu, J. Liu, and O. Yang, "Wireless traffic modeling and prediction using seasonal arima models," in *IEEE International Conference on Communications*, vol. 3, 2003, pp. 1675–1679 vol.3.

[233] X. SHI *et al.*, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[234] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on computing*, vol. 15, no. 2, pp. 364–383, 1986.

[235] J. Song, X. Tan, D. Zhou, and L. Wang, "Efficient compression of language model embeddings with pca," *arXiv preprint arXiv:2104.05227*, 2021.

[236] E. Reif, A. Yuan, M. Wattenberg *et al.*, "Visualizing and measuring the geometry of bert," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[237] Z. Lan, M. Chen, S. Goodman *et al.*, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," in *International Conference on Learning Representations (ICLR)*, 2020.

[238] R. Jia, K. Gong, and Z. Liu, "Adversarial robustness of pre-trained transformers via noise filtering," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[239] A. Mauroy and J. Goncalves, "The koopman operator framework for analyzing and modeling dynamic systems," *Advances in Neural Information Processing Systems*, 2020.