Local & Federated Learning at the Network Edge for Efficient Predictive Analytics

Natascha Harth*, Hans-Joerg Voegel

BMW Group Research, New Technologies, Innovations, Parkring 19, 85748 Garching, Germany

Kostas Kolomvatsos

Department of Informatics and Telecommunications, University of Thessaly

Christos Anagnostopoulos

School of Computing Science, University of Glasgow

Abstract

The ability to perform computation on devices present in the Internet of Things (IoT) and Edge Computing (EC) environments leads to bandwidth, storage, and energy constraints, as most of these devices are limited with resources. Using such device computational capacity, coined as Edge Devices (EDs), in performing locally Machine Learning (ML) and analytics tasks enables accurate and real-time predictions at the network edge. The locally generated data in EDs is contextual and, for resource efficiency reasons, should not be distributed over the network. In such context, the local trained models need to adapt to occurring concept drifts and potential data distribution changes to guarantee a high prediction accuracy. We address the importance of personalization and generalization in EDs to adapt to data distribution over evolving environments. In the following work, we propose a methodology that relies on Federated Learning (FL) principles to ensure the generalization capability of the locally trained ML models. Moreover, we extend FL with Optimal Stopping Theory (OST) and adaptive weighting over personalized and generalized models to incorporate optimal model selection decision making. We contribute with a personalized, efficient learning methodology in EC environments that can swiftly select and switch models inside the EDs to provide accurate predictions towards changing environments. Theoretical analysis of the optimality and uniqueness of the proposed solution is provided. Additionally, comprehensive comparative and performance evaluation over real contextual data streams testing our methodology against current approaches in the literature for FL and centralized learning are provided concerning information loss and prediction accuracy metrics. We showcase improvement of the prediction quality towards FL-based approaches by at least 50% using our methodology.

Keywords: Federated Learning, Quality-Aware analytics, Local Learning, Personalization, Optimal Stopping Theory.

1. Introduction

1.1. Motivation & Challenges

Measurements of the surrounding environment and the continuous creation of contextual data are becoming pervasive in our daily lives. Most sensing and computing devices are connected over the internet to transfer the data to a Central Location (CL), e.g., the Cloud, for further analysis and processing. Sometimes, such collected data might also contain user and personal information that needs protection. Privacy has exhibited increased importance over the recent years with regulations defined for such purposes, such as General Data Protection Regulation (GDPR) [1] or California Consumer Privacy Act (CCPA) [2]. Applications have started to consider collecting fewer and processing less sensitive data of users. Consequently, sharing raw data over the network to train and deploy high-quality Machine Learning (ML) algorithms is constrained by privacy-preserving regulations and the concerns of users to share the data. In addition, the tremendous increase of computing devices raises bottlenecks of transmission and data collection due to limited bandwidth, storage, and connectivity losses. The Internet of Things (IoT) environments extend over time from simple sensors into small computers with the ability to perform computation at the source of the data, thus, overcoming the bottlenecks of legacy centralized architectures. The developing Edge Computing (EC) paradigm exploits such computational capacity of the Edge Devices (EDs) to enable ML raising, thus the possibility to achieve real-time local predictive analytics and actuation due to significantly reduced latency. However, most EC environments deploy the training and management of ML models in centralized locations, while inference is performed at the network edge. During the centralized model training, raw data are transferred from EDs to central collection points. This raw data transfer results in privacy concerns and issues towards

*Corresponding author

Email addresses: natascha.harth@bmwgroup.com (Natascha Harth), hans-joerg.voegel@bmwgroup.com (Hans-Joerg Voegel), kostasks@uth.gr (Kostas Kolomvatsos), christos.anagnostopoulos@glasgow.ac.uk (Christos Anagnostopoulos)

the collected and transferred data.

By pushing intelligence and computation to the EDs, coping with privacy and dealing with bottlenecks of bandwidth and real-time decision making, Federated Learning (FL) is introduced as an upcoming methodology that generates knowledge from data without sharing the actual raw data [3]. FL aims to build a global model by performing the learning (training) at EDs and transferring only the locally updated ML model parameters over the network. This method allows distributing the training process to the device level only. FL enables to build general knowledge at a CL without revealing individual device information, data, or context. FL also provides a solution to overcome the introduced constraints by transmitting only models over the network, which reduces the bandwidth. Furthermore, FL enables the EDs to perform analytics locally and autonomously decide on updating their local models. Without the involvement of a centralized decision-making process, the expected latency of analytics tasks is evidently reduced.

From the predictive analytics side, a major challenge is that many applications rely on contextual data streams and evolving data to support prediction and analysis tasks. Such data, by nature, change over time and usually involve irregular and unpredictable updates (a.k.a. concept drifts) in the underlying distribution. Concept drifts require continuous re-training of the developed ML model. Centralized online model learning techniques have been presented by mainly adopting the Gradient Descent mechanism to build a generalized model incrementally over evolving data. Primarily, sliding window methodologies have been used to consider qualitative analysis over evolving data streams. However, as highlighted earlier, transferring the data to a CL is not feasible and efficient for analytics. Aiming to generate a global model that can be deployed in new EDs and represent the users' behavior is of most interest. In this context, FL has been used for converging ML algorithms without considering the need for continuous updating under these dynamically changing cases. This indicates the weakness of the FL paradigm to follow the progress of the model building phases during their training period. As highlighted, such weakness of not continuously monitoring and, thus, adapting to the specific user behavior leads to potential inaccuracies, user dissatisfaction and degradation of the anticipated QoS. The challenge already occurring in FL with the communication overhead of transferring updated models and/or merged models over the network is increasing within EC environments of constant learning, given that online monitoring and adaptation to the stationary/non-stationary trends of data is required. Even though FL is developed for non-independent and identically distributed (non-i.i.d.) data, deploying a generalized global model into EDs for prediction and actuating results in lower accuracy than having a local *personalized* model. The concept of personalized models in FL has been investigated in additional work and highlighted as an essential research question in [4]. We refer the interested reader to Section 2; however, we summarize here that the presented work of personalized FL primarily considers model convergence without the need for constant retraining and adapting due to possible underlying changes of contextual data or assumes a fully decentralized architecture without a central coordinator. In this context, we emphasize that personalized models are expected to overfit the local data and, evidently, produce poor prediction results in sudden changing environments compared to generalized models. The challenge and, evidently, the motivation lies in locally selecting the best and most accurate model to perform analytics tasks within the EDs monitoring in parallel the progression of the prediction capability of the models.

Overcoming these issues, sophisticated mechanisms that can optimally and swiftly decide on *fusing* and/or *swapping* between a generalized (federated) model and a personalized (local) model are deemed appropriate and critical to guarantee qualitative predictive analytics at the network edge. This decision making needs to be *efficient* in terms of computation, communication, and resource consumption as EDs are resource-constrained. And, this is the novelty in our paper, where we introduce two efficient mechanisms that cope with the aforementioned challenges: (i) an adaptive federated-personalized model weighting (aggregation) mechanism and (ii) a time-optimized federated-personalized model selection/swapping mechanism at the network edge. Both mechanisms achieve a relatively low memory footprint by introducing low computation and communication-aware algorithms for predictive analytics at the edge. These mechanisms support model learning and inference by combining a personalized FL methodology with traditional FL, thus, adapting to local data distribution shifts and providing highly accurate predictions over contextual data streams.

1.2. Use-Cases & Applications

We elaborate on the importance of personalized and generalized learning paradigms on certain application domains, notably on the automotive domain. The adoption of such learning paradigms are essential for *predictive maintenance* and *driver behavior identification* services. The management of vehicle fleets can benefit from the combination of personalized and FL paradigms, given the contextual 'health' information generated by certain monitored components in each vehicle. Real-time maintenance platforms cater to just-in-time supply chains, reducing stocking costs and lowering maintenance times by aggregating localized trained ML models from vehicle fleets. Estimation of State-of-Health and Remaining Useful Lifetime of individual components, e.g., HGV Batteries or combustion engines [5], [6], can be initially predicted in-vehicle and fused in a CL for further predictive maintenance analytics for the entire fleet. Moreover, localized fault diagnosis for detecting abnormal data merged with global outlier detection models has the potential of reducing maintenance expenditure of complex failures.

The automotive domain has seen emerging interest in the classification of driving behavior and driver identity recognition [7]. The latter has seen commercial adoption by automotive manufacturers (e.g., Subaru's Driver Monitoring System[1]), by employing driver-facing cameras for distraction and drowsiness warning or driver personalization [8]. Local ML models

[1] https://www.subaru.com.au/driver-monitoring-system

provide facial identification under different lighting conditions and driver positions and gaze estimation to determine driver attentiveness, which enables proactive safety mechanisms within vehicles. Applications based on classification of general driving behavior using ML have gained significant attention in the last few years, benefiting from enhanced personalized local ML models utilizing the vehicular sensor data with significant features extracted in a CL after merging local ML models. By providing accurate driver identification via behavioral patterns, vehicles can provide customized experiences for drivers but also infer further knowledge, e.g., theft detection [9]. The classification of the local environment, e.g., identification of road complexity [10], can be used to optimize vehicle performance and provide adaptive safety systems tuned for different conditions by aggregating local models from different vehicles in a FL context. Using this federated and tailored/personalized knowledge, vehicles can become proactive elements in providing traffic safety mechanisms (e.g., alerting the driver of their performance) or used by authorities and/or insurance companies to assess the risks involved in driving styles. By intelligently combining local models in light of building powerful federated models, which in turn can be tailored to, e.g., vehicles, is the key for high quality and accurate applications. Personalized models that provide a unique driving experience are essential. However, with changes occurring in the environment (e.g., car-sharing with multiple drivers, weather change, personal emotion) that the personalized model never experienced, the prediction accuracy decreases, and the quality is not guaranteed. Generalized model learning across all drivers via local models aggregation can increase quality and accuracy, as it is equipped with more situations and data and fits even towards unseen data well. Therefore, predictive analytics at the edge enhanced with federated-personalized model selection strategies is essential to provide quality-aware services.

1.3. Contribution & Organization

To the best of our knowledge, we are among the first to propose personalized edge-centric analytics mechanisms *fusing* the principles of FL for generalization with adaptive model weighting and the principles of Optimal Stopping Theory (OST) to achieve localized, efficient and time-optimized model selection. We summarize our contributions as follows:

- A local adaptive model re-training scheme for evolving contextual data streams in EDs aggregating the FL model towards global knowledge generation for high prediction accuracy;
- A novel, lightweight, and adaptive model weighting mechanism (coined Adaptive Selection Model-ASM) of the personalized local model and FL model across EDs, which improves the quality, accuracy, and robustness for data concept drifts;
- A novel and lightweight (computation and communication efficient) time-optimized model selection mechanism (coined Time-optimized Model Selection-TOSM) swapping between personalized local and federated generalized models over continuously evolving data streams using the principles of OST;
- We provide a theoretical analysis of the TOSM, proving its optimality and uniqueness of solution along with incremental algorithms for optimal decision making. We also provide computational, communication and storage complexity of the models showcasing their suitability for EDs;
- We perform a comprehensive evaluation and comparative assessment against baseline models, approaches, and mechanisms found in the literature [11], [12], [13], [14] using real-data sets.

The rest of the paper is structured as follows: Section 2 discusses related work in the field of personalized FL, while Section 3 elaborates on the rationale and problem fundamentals in personalized learning of EC environments. We introduce our personalized learning mechanisms in Section 4 and the time-optimized model selection in Section 5. Comprehensive comparative assessment with approaches and methods found in the literature is provided in Section 6. Section 7 concludes the article with a summary and our future research agenda.

2. Related Work

FL has engaged the EDs to locally learn over their local data a global ML model, without revealing and/or transferring any data towards a CL. The overarching aim is to train a global function placed at CL by pushing the training towards the EDs. FL has been extensively studied in scenarios where numerous IoT devices are present (e.g., smartphones). The CL aggregates the gradients of the locally trained models together into a new global function (coined as FedAvg) introduced in [11, 15].

This kind of distributed learning provides the possibility to use the computational capacity of EDs and access to local data achieving privacy, real-time actuation, and robustness. FL can perform a fast convergence of the global trained model on massively distributed, non-IID, and unbalanced data. In many applications, generating a global model, which generalizes the data is of utter importance. However, in many EC environments, the generated data are non-IID. In such a case, the non-IID provokes personalized and local models to outperform the generalized global model mostly. This effect has been evidenced in [16, 12] by performing localized edge-learning to increase the quality of predictive analytics. Additionally, the authors in [4] argue an open research question is on *when* to choose the global model, providing generalization, over the local one, providing individuality. Personalization and FL has recently attracted the attention of the research community. Specifically, [17] summarizes the combination of personalized processing and FL. The approaches in [18, 19, 20, 21] show the impact of personalization in FL environments and its significant improvement towards prediction accuracy. The methods in [18, 19, 20, 21] cope with a fully decentralized architecture

in which no global server coordinates the communication nor has knowledge about the model gradients or the overall generated global model. EDs collaborate to share information and improve their model through other EDs. The idea of decentralized and personalized FL assumes complete connectivity among EDs, which is hard to achieve even with current network communication channels such as 5G. Additional network traffic is needed for synchronizing. This communication is due to connectivity issues, failures, and communication costs, a disadvantage compared to architectures with a central coordinator. Therefore, a fully decentralized approach will be hard to implement in many domains. Moreover, the methods mentioned above do not explicitly deal with evolving contextual data and concept drifts. Methods proposed for personalized learning by using FL rely on a hierarchical system involving a central coordinator performing the model merging as introduced in [14]. We depart from [14] by proposing a variant being less computationally complex, which adapts the relation between global generalized and local personalized models. Our adaptive policy is performed on a discrepancy-aware process that considers rewards of historical accuracy and difference in the global and local model parameters.

Meta-learning for FL, as a possibility to overcome the issue of heterogeneous data, has been introduced in [22, 23]. These methods achieve improvement in accuracy using personalized models in non-IID environments. The model in [24] introduces a FL optimization technique that merges locally trained models to a global model providing a specialized gradient update procedure. Moreover, the approach in [25] combines a globally trained model and a locally fine-tuned model adopting an adaptive parameter in controlling the relationship between the global and the local model. Continuous efficient model selection and updating due to the underlying data changes are essential in EC dynamic environments, which are not explicitly considered in the mentioned literature.

Besides the personalization aspect of FL, we adopt the principles of FL over contextual data with evolving nature. Performing FL on developing data is still an open question where a few research methods have been proposed. The work in [26] introduces an online asynchronous version of FL. It suggests a convergence strategy of locally updating the FL model using feature representation learning at the CL. However, the model in [26] does not consider efficiency and resource-constrained environments. This aspect is considered in [27] which shows the importance of local distributed learning in resource constrained edge networks. The implementation of FL in a resource constrained environment presented in [28] provides a communication efficient compression technique. The recent work in [29] considers the essential requirement of using efficient communication in resource-constrained environments and the importance of personalization in a combined implementation. Yet, these methods assume the global convergence of a model without concept drifts and potential changes in the underlying data, which renders the necessity of deciding on the best model selection for predictive analytics. Finally, local and global models update and decision making on forwarding local models have been proposed in [16, 30]. Such methods consider model updates based on either accuracy discrepancy or the difference between the current and past versions. We depart from this strategy by introducing an optimal model selection mechanism based on the principles of the OST by updating the CL with continuously trained local models. In Table 1 we summarize the related work and its limitations for a better overview.

3. Rationale & Problem Fundamentals

3.1. Rationale

FL can be seen as a distributed ML optimization problem in a network of EDs. The aim is to locally optimize an objective function $\mathcal{J}$ over distributed datasets. Consider a network of K EDs indexed by $k \in \{1, \ldots, K\}$, where each ED contains a local subset of data D_k from the set $D = \cup_{k=1}^{K}\{D_k\}$. A dataset D_k is generated through Sensing and Actuating Nodes (SANs) sensing continuously d-dimensional contextual data vectors $\mathbf{x}_t$ with $\mathbf{x} \in \mathbb{R}^d$ and discrete time domain $t \in \mathbb{T} = \{1, \ldots, T\}$ with $T \in \mathbb{T}$. At time instance t, a new contextual vector $\mathbf{x}_t$ is received at ED k, which is stored locally in the dataset D_k. In centralized learning, this data vector is forwarded to a CL that minimizes the objective function $\mathcal{J}$ over the entire dataset D.

A loss function $\mathcal{L}(\mathbf{x}, y; \mathbf{w})$ of a parametric ML model can be approximated by adopting the Empirical Risk Minimization (ERM) instead of the expected loss over all training data. Specifically, an approximation of $\hat{\mathcal{J}}$ over a training set D with N training input-output pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ using a set of parametric ML models $f(\mathbf{x}; \mathbf{w}) \in \mathcal{F}$ with parameters $\mathbf{w} \in \mathbb{R}^d$ can find the optimal solution for the ERM. This is expressed as follows:

$$\arg\min_{\mathbf{w}\in\mathcal{F}} \mathcal{J}(\mathbf{w}) \approx \hat{\mathcal{J}}_N(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\mathbf{x}_i, y_i; \mathbf{w}). \tag{1}$$

In FL, the optimization of $\mathcal{J}$ is performed by locally optimizing the objective function $\mathcal{J}_k$ in the kth ED over D_k with n_k data vectors, so that $\sum_{k=1}^{K} n_k = N$. The CL is, then, aggregating the locally obtained objective functions $\mathcal{J}_k$:

$$\mathcal{J}(\mathbf{w}) = \sum_{k=1}^{K} \frac{n_k}{N}\mathcal{J}_k(\mathbf{w}) = \sum_{k=1}^{K} \frac{n_k}{N}\frac{1}{n_k}\sum_{i\in D_k} \mathcal{L}_k(\mathbf{x}_i, y_i, \mathbf{w}). \tag{2}$$

The kth loss function $\mathcal{L}_k(\mathbf{x}, y, \mathbf{w})$ corresponds to the kth local optimization function $\mathcal{J}_k(\mathbf{w})$. A widely adopted incremental method to solve the optimization problem in Equation (2) is Stochastic Gradient Descent (SGD). Specifically, at each iteration (step) t, the kth ED aims to converge towards the minimum function $\mathcal{J}_k$ over its local data D_k given a new incoming training pair $(\mathbf{x}_t, y_t)$. This is achieved by adjusting the local ML model parameters $\mathbf{w}_k^t$ with a given learning rate $\eta \in (0, 1)$ and the gradient of the previous model parameter $\nabla\mathcal{J}_k(\mathbf{w}_k)$ at step $t-1$. The kth ED is then performing the following update on its parameter at t:

$$\mathbf{w}_k^{t+1} \leftarrow \mathbf{w}_k^t - \eta\nabla\mathcal{J}_k(\mathbf{w}_k^t). \tag{3}$$

After multiple iterations, the updated model parameter $\mathbf{w}_k$ of the kth ED is sent to the CL for aggregation. The FedAVG

Reference	Proposed Methodology	Limitation
[17]	Survey of personalization and FL	
[18, 19, 20, 21]	Decentralized personalized FL	Communication overhead due to synchronization, assumption of convergence
[14]	Local Representation Model in ED with global features	No consideration of adapting the model in changing environments
[22, 23]	Meta-Learning with FL	No consideration of adapting the model in changing environments
[24]	Personalized models are aggregated to a global model	Loosing the generalization of a global model for controlling unseen events
[25]	Adaptive weighting of local and global model	Convergence assumption, weighting fixed on gradient difference
[26]	Online asynchronized FL through feature representation	No consideration of resource constraint environment or continous learning
[27]	Adaptive FL under resource constrains	No consideration of personalization or continuous updating
[28]	Resource efficient FL compression	No consideration of personalization or continuous updating
[29]	Communication efficient personalized FL	Computationally complex, convergence assumption
[16, 30]	Local model forwarding mechanism in edge environments	Meta-data sharing, accuracy discrepancy forwarding

Table 1: Summary of related work

algorithm introduced in [11] aggregates the received model parameters from all the EDs in a central coordinator towards a new generalized ML model. This aggregation is provided by the following weighted average:

$$\mathbf{w}^{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} \mathbf{w}_k^{t+1}. \quad (4)$$

After merging the local gradients of the kth ED at the CL, the final federated ML model, hereinafter notated as f_{FL} is then distributed back to the EDs, which can be used locally for inference/prediction. The kth ED is selected at a random time to update the distributed federated model f_{FL} with its local data stored in a sliding window of the most recent M data vectors $\mathcal{W}_k^t = \{\mathbf{x}_{t-M+1}, \ldots, \mathbf{x}_t\}$. Then, the newly updated model parameter $\mathbf{w}_k$ is sent back to the CL.

3.2. Problem Definition

As highlighted in Section 2, the basic implementation of FL introduces two fundamental challenges. **Challenge 1**: the first challenge is the adaptation of the generalized model to constantly changing environments (where the underlying data over the EDs might be changing), in which the assumption of converging to a global minimization of the objective function $\mathcal{J}$ does not always hold. This implies that the minimization of $\mathcal{J}$ needs to be revised if, for instance, the underlying data distribution changes, which is not so rare especially in contextual data streams (e.g., concept drifts). **Challenge 2**: The second issue arises as the EDs are about to use, for further training, the newly received generalized model f_{FL} from the CL. This, evidently, renders the locally adapted model $\hat{f}_{FL}$ obsolete, since this knowledge cannot be re-used in the future. This means that, the very local statistical parameters learned and captured from the local data of the EDs are aggregated with other EDs' parameters, via the aggregation principle of the FL (refer to Equation (4)). This, potentially results in losing the local statistical dependencies of the local data, since all the local parameters are aggregated in light of concluding on a generalized model, which lacks of the individual statistical dependencies across all the EDs. This fundamentally derives from the generalization of the models over non-IID data, as highlighted in Section 2, showcasing the importance of keeping also the local learned statistical dependencies of each individual ED, coined as 'personalization' for the predictive analytics tasks at CLs.

In order to cope with the Challenges 1 and 2, i.e., dealing with concept drifts and keeping and transferring local data dependencies to future updated models, we introduce a dual model deployment mechanism inside EDs. Specifically, the kth ED develops an evolving personalized (local) model $f_k(\mathbf{x})$ *in parallel* to the generalized federated model $f_{FL}(\mathbf{x})$. An overview of the envisaged EC architecture is provided in Figure 1. The major challenge, therefore, of each ED with the deployed dual model mechanism is to efficiently and effectively decide on the *correct* ML model for predicting $\hat{y}$ with either $\hat{y} = f_{FL}(\mathbf{x})$ or $\hat{y} = f_k(\mathbf{x})$ upon receiving a new incoming input $\mathbf{x}$ at time t. In addition, EDs are limited with their resources, which does not allow complex algorithms to be implemented for timely decision making locally; recall that this decision should be taken at every time t, when a new input $\mathbf{x}$ is coming to the ED for prediction. Furthermore, the timely decision making of efficiently selecting and/or aggregating personalized and federated models is influenced by the nature of many applications like autonomous cars and smart homes involving supervised learning models (e.g., classification tasks for driver identification behavior). In this context, checking the performance of the (supervised) learning models cannot always be achieved since labeled data are not always provided nor being available at the time of prediction. Hence, we are in need of introducing computationally lightweight methods for fusing the personalized and generalized federated models inside each ED for predictive analytics tasks by exploiting both the local statistical features learned from the personalized statistical learning and the generalization/predictability capability of the federated model taking into consideration the dynamical changes of the underlying data.

The idea of training personalized FL models inside EDs has started to gain significant interest in the last years to overcome issues related to heterogeneous non-IID data, continuously changing data and concept drift adaptation. The authors in [16, 12] highlighted the importance of ML models building at EDs by adopting centralized ensemble learning over these models at the CL for supporting predictive analytics. However, the adoption of ensemble learning within EDs involving complex pruning strategies per incoming input for timely prediction is considered prohibitive due to evident restrictions of sharing user-specific models and meta-data with the CL via the network. On the other hand, the general FL method is purely based on generating a global model on the CL being trained over dis-

tributed datasets without accessing the raw data on each ED, so achieving data privacy *by design*. The drawback of generalized FL models, in the first place, can be overcome by building and fusing local models. This, however, results in another problem where by only having a local personal model at the ED, the adaptation towards unseen data induced by concept drifts will evidently generate inaccurate and wrong predictions.

Our work focuses on using the combined power of the inherent generalization of the FL-generated model via a group of EDs and the advantage of the personalization local model inside each ED. This is showcased to provide qualitative prediction over changing environments at the network edge, while being resource-efficient by reducing the computational complexity per prediction request. By introducing our parallel model deployment of the federated model f_{FL} and the personalized model f_k inside each k ED deals with two fundamental challenges: (**R1**) definition of a mechanism that online chooses to use either f_{FL} or f_k, or an aggregation over both models for making prediction per request, and (**R2**) the definition of a mechanism that determines *how* and *when* to decide on which of both models each ED should choose to cope with changes in the underlying data in dynamic environments.

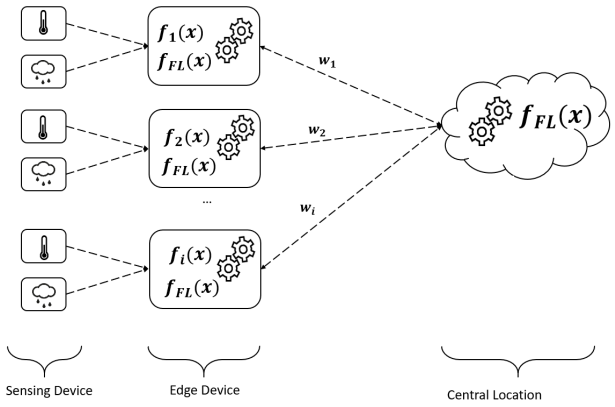


Figure 1: Generating global knowledge f_{FL}using FL via distribution of local models to the CL while keeping locally derived knowledge f_k at the EDs.

4. Personalized Learning at the Edge

In this section, we contribute to the requirement **R1**, as presented in Section 3.2, by introducing two strategies, coined Evolving Federated Model (EFM) and Local Federated Model (LFM), for personalized and efficient learning at the network edge based on the fundamentals of FL. We then propose a methodology for aggregating the local and federated models in each ED based on the statistically recent prediction capability of the models.

4.1. Strategies for Personalized Learning at the Edge

Consider the ED k, which receives its d-dimensional contextual data vector $\mathbf{x}_t$ from the SANs at each time $t \in \mathbb{T}$. Data vectors are stored in the local window $\mathcal{W}_k$ of fixed size M. Only at a pre-defined selected epoch these data in the window will be used to update the newly received model f_{FL} from the CL, as it will be explained later. Initially, the ED's personalized model f_k is set to be equal to the centrally received federated model f_{FL}. The choice of using a pre-built model as a starting point for prediction lies in accordance to the transfer learning theory [31, 32]. Specifically, the principle of the transfer learning highlights that using a baseline model for retraining achieves an earlier convergence and higher accuracy than starting the learning process from scratch. If the ED k needs to perform a prediction, it uses its current cached ML model. At each pre-defined epoch, the ED k is selected to receive the most recently trained generalized FL model f_{FL} from the CL. Then, the ED k incrementally calculates over its locally stored data in its window $\mathcal{W}_k$ the model gradient updates using Equation (3). In this context, the (asymptotic) computational complexity for these updates requires $O(Md)$ time, with $M \ll n_k$ with $O(Md)$ storage. Note: the interested reader could also refer to [16] for resource-efficient SGD processes regarding local ML training, which are beyond of the scope of this paper. The updated parameter $\mathbf{w}_k$ is then forwarded to the CL. The CL (FedAVG) can then use the new parameters received only from the *selected* EDs to obtain the new model f_{FL}; refer to Equation (4). All the EDs in turn receive the updated federated model f_{FL} from CL, which can be used in conjunction with the local f_k model for performing predictions locally on each ED k.

4.1.1. Evolving Federated Model

The Evolving Federated Model (EFM) incorporates personalization into an edge environment by advancing on the basic functionality of the FL paradigm. The EFM introduces a communication-efficient strategy that discards the updating communication *from* CL *to* the ED after generating the new generalized model f_{FL}. Specifically, the ED k instead of receiving the new updated federated model f_{FL}, it keeps the updated model as the f_k and continuously evolves that model each time a new input-output pair $(\mathbf{x}_t, y_t)$ is collected by the SANs. When the ED k is selected at a specific update epoch, it sends to CL the updated local model parameter $\mathbf{w}_k$ and receives the recently updated federated model f_{FL}. Then, the model f_k is replaced with the f_{FL}. Hence, in EFM the communication is reduced and controlled by the update epoch frequency s. The ED can adapt to the evolving data and concept drifts by continuously learning and retraining the generalized model to its personalized environment. The computational, storage and communication (number of messages sent between ED and CL) is provided in Remark 1. The EFM process for the ED k is provided in Algorithm 1.

4.1.2. Local Federated Model

The Local Federated Model (LFM) limits the communication further between EDs and the CL compared to EFM. The idea is to initialize the local model f_k with the received federated model, ie., $f_k = f_{FL}$ for each ED k. Then, the ED k continuously updates its model using SGD. Instead of the CL requesting each epoch the updates for the f_{FL}, the ED k regularly sends its local model f_k to CL with update epoch s. Within the CL, the parameters of each f_k are merged into a global model f_{FL}.

The model aggregation part of the LFM process is based on the method introduced in [14], where no distribution of merged models is required. However, in our context, we enforce LFM to distribute the federated model f_{FL} only on those EDs, whose local models f_k are significantly deviated from the current federated model in CL, or, when a new ED joins the network. Specifically, given a d-dimensional discrepancy model threshold $\boldsymbol{\xi} \in \mathbb{R}^d$ between the parameters of the f_{FL} and f_k models, the CL sends the f_{FL} model to the ED k, which replaces its local model. With this selective model update, the EDs obtain the generalizability of the federated model adapting, in turn, their local models for personalization. The computational, storage and communication (number of messages sent between ED and CL) is provided in Remark 1. The LFM process for the ED k is provided in Algorithm 2.

Algorithm 1 Evolving Federated Model (EFM)

receive federated model f_{FL} from CL
$f_k \leftarrow f_{FL}$
for each time $t = 1, \ldots,$ **do**
  receive input-output $(\mathbf{x}_t, y_t)$ from SANs
  update local model f_k using Equation (3)
  update $\mathcal{W}_t \leftarrow \{\mathcal{W}_{t-1} \setminus \{(\mathbf{x}_{t-M+1}, y_{t-M+1})\}\} \cup \{(\mathbf{x}_t, y_t)\}$
  if ED k is selected by CL **then**
    send updated $\mathbf{w}_k$ to CL for aggregation
    receive new federated model f_{FL} from CL
    replace $f_k \leftarrow f_{FL}$
  end if
end for

Algorithm 2 Local Federated Model (LFM)

receive federated model f_{FL} from CL; $f_k \leftarrow f_{FL}$
for each time $t = 1, \ldots,$ **do**
  receive input-output $(\mathbf{x}_t, y_t)$ from SANs
  update local model f_k using Equation (3)
  update $\mathcal{W}_t \leftarrow \{\mathcal{W}_{t-1} \setminus \{(\mathbf{x}_{t-M+1}, y_{t-M+1})\}\} \cup \{(\mathbf{x}_t, y_t)\}$
  if $t \mod s = 0$ (update CL with frequency s^{-1}) **then**
    send updated $\mathbf{w}_k$ to CL for aggregation
  end if
  if ED k is selected by CL for update **then**
    receive new federated model f_{FL} from CL
    replace $f_k \leftarrow f_{FL}$
  end if
end for

4.2. Adaptive Model Weighting

The personalization adoption of FL so far deals with the use of a single model within a ED, which adapts to the data and continuously evolves given new data. In the following, we propose an adaptive model selection methodology launched within each ED. This methodology uses a statistical reward mechanism over the recent historical prediction accuracy to weight the relation between local and federated models. By considering the frequent appearance of concept drifts and distribution shifts over contextual and evolving data, it is important to maintain, besides a personalized model, also another model which represents a generalization of the data, especially when focusing on the quality of the prediction. We propose a combination of keeping the two models in parallel updated and selected in the prediction tasks within each ED. As LFM and EFM only store one model and, continuously, retrain it, the Adaptive Selection Model (ASM) exploits both the generalized model f_{FL} and the local model f_k inside each ED k. The importance of keeping and maintaining both models up-to-date inside each ED is mainly to overcome the issue of losing generalization and adaption to unseen data by only deploying personal models. Moreover, previously unknown relationships within data in EDs can be changing from IID to non-IID data relationships. This cannot be identified at the start of the application deployment, which could have potentially achieved better prediction accuracy from either the generalized or the personalized model.

The reward mechanism in ASM weighs the local and federated model into a combined predicted $\hat{y}$ given an input $\mathbf{x}$. The predicted $\hat{y}$ is calculated locally in ED k by using the local prediction $f_k(\mathbf{x})$ and the federated prediction $f_{FL}(\mathbf{w})$ with an adaptive balancing weight $\alpha \in (0, 1)$. Firstly, when the ED k receives from its SANs a new contextual vector $\mathbf{x}_t$ at time t, the absolute prediction error ϵ_k from the local model f_k and the absolute prediction error ϵ_{FL} for the federated model f_{FL} w.r.t. actual prediction y_t are calculated, respectively:

$$\epsilon_L = |y_t - f_k(\mathbf{x}_t)|, \tag{5}$$

$$\epsilon_{FL} = |y_t - f_{FL}(\mathbf{x}_t)|. \tag{6}$$

Given these two errors in ED k, we introduce a binary *reward value* $\theta \in \{0, 1\}$, which is used as the factor for balancing the two models. Specifically, in ED k at each time t, based on ϵ_{FL} and ϵ_L, the reward $\theta = 0$, if the local model is performing better than the generalized one, and $\theta = 1$ otherwise, i.e.,

$$\theta_t = \begin{cases} 0, & \epsilon_{FL} > \epsilon_L \\ 1 & \epsilon_{FL} \leq \epsilon_L. \end{cases} \tag{7}$$

As shown in Equation (7), a positive reward is given to the federated model f_{FL} if the absolute error ϵ_{FL} is smaller than the absolute error of the local mode f_k noted as ϵ_L. The reward mechanism deployed inside the ED k should not only incorporate the current performance of f_k and f_{FL} but, also, the historical performance, i.e., the most recent reward values θ_t stored in a sliding window $\mathcal{U}_k^t = \{\theta_{t-L+1}, \ldots, \theta_t\}$ of size L for the last $t-L$ predictions (time instances). Note: the use of the sliding window is chosen towards the adoption of lightweight processing methods for handling continuous evolving data and adaptation to changing environments able to act immediately on concept drifts [33, 34]. As the window $\mathcal{U}_k^t$ consists of the most recent rewards, at each time t, we can calculate a ratio that represents the performance of both models over the time horizon $t - L$, which refers to the adaptive weight α, i.e.,

$$\alpha = \frac{1}{L} \sum_{\tau=1}^{L} \theta_\tau. \tag{8}$$

The α value represents the most recent prediction performance of both models in ED k. Each time the ED k is performing prediction, the two predictions of $f_k(\mathbf{x})$ and $f_{FL}(\mathbf{x})$ are weighted to the final prediction $\hat{y} = f_{ASM}(\mathbf{x})$, i.e.,

$$f_{ASM}(\mathbf{x}) = \alpha f_{FL}(\mathbf{x}) + (1-\alpha) f_k(\mathbf{x}). \quad (9)$$

Algorithm 3 Adaptive Selection Model (ASM)

Central Location:

initialize federated parameter $\mathbf{w}_{FL}$
for each update epoch **do**
  select a random subset $\mathcal{K} \subset \{1, \ldots, K\}$
  for each ED $k \in \mathcal{K}$ **in parallel do**
    receive parameter $\mathbf{w}_k$ from ED k
    $\mathbf{w}_{FL} \leftarrow \sum_{k \in \mathcal{K}} \frac{n_k}{N} \mathbf{w}_k$
  end for
end for

Edge Device k**:**

receive federated model f_{FL} from CL ($t = 0$)
for each time $t = 1, \ldots,$ **do**
  receive input-output $(\mathbf{x}_t, y_t)$ from SANs
  update local model f_k using Equation (3)
  calculate local error $\epsilon_L \leftarrow |y_t - f_k^t(\mathbf{x}_t)|$
  calculate federated error $\epsilon_{FL} \leftarrow |y_t - f_{FL}^t(\mathbf{x}_t)|$
  calculate reward θ_t using Equation (7)
  update window $\mathcal{U}_t \leftarrow \{\mathcal{U}_{t-1} \setminus \{\theta_{t-L+1}\}\} \cup \{\theta_t\}$
  calculate weight α using Equation (8)
  if prediction of $\hat{y}_t$ is requested **then**
    $\hat{y}_t = f_{ASM}(\mathbf{x}_t)$ using Equation (9)
  end if
  if ED k is selected for updating **then**
    update f_k using Equation (3) given $((\mathbf{x}_t, y_t))$
    return updated $\mathbf{w}_k$ to CL.
  end if
end for

The α value combines f_{FL} and f_k predictions towards $\hat{y}$ for every input $\mathbf{x}$ using exponential smoothing [35]. An $\alpha \to 1$ indicates that more influence comes from the federated model f_{FL}, whereas, $\alpha \to 0$ places more importance on the local model f_k predictions. Algorithm 3 shows the calculations in the CL and the ED k towards the ASM mechanism.

Remark 1. *We elaborate on the computational, storage and communication complexity of the ASM, EFM, and LFM models along with the base FL mechanism (FedAVG). The communication complexity is expressed in terms of the messages sent from CL to ED and vice versa given a specific model. Firstly, let us provide the communication complexity of FedAVG, which involves all the m EDs at predefined epochs $s < T$ for models aggregation and updates during a specific time horizon T. This indicates that the communication (messages sent) of CL and EDs within T is $O(m\frac{s}{T})$. Within two consecutive update epochs, each ED updates its local model in $O(Md)$ time based on SGD storing the data vectors in a sliding window of size M. This also requires $O(Md)$ storage. In the EFM process provided in Algorithm 1, the CL at time t selects a subset $\mathcal{K}$ of m EDs from all the connected K EDs for being updated with the local models' parameters and then distributing the updated federated model. Let ED k be the ED selected to update CL with its local model f_k and then receive the updated federated model f_{FL} at model update epoch s. Given a finite time horizon T, with $s < T$ epochs, the probability that the ED k is selected $P\{k \in \mathcal{K}\}$, i.e., the probability that $k \in \mathcal{K}$ of m EDs is $\frac{\binom{m-1}{K-1}}{\binom{m}{K}} = \frac{m}{K}$. Hence, the communication (messages sent) of CL and ED k within horizon T is $O(\frac{m}{K}\frac{s}{T})$. The EFM updates the local model f_k in $O(Md)$ time adopting SGD over the sliding data window $\mathcal{W}$ of size M, which requires $O(Md)$ storage.*

In the LFM provided in Algorithm 2, the communication for updating the CL with the local model f_k is: $O(\frac{s}{T})$ given a fixed model update epoch s. The LFM updates the local model f_k in $O(Md)$ time adopting SGD over the sliding data window $\mathcal{W}$ of size M with $O(Md)$ storage. Moreover, given a discrepancy threshold vector $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_d]^\top$ between the federated model parameter $\mathbf{w}$ and the local model parameter $\mathbf{w}_k$, under a L_p norm: $\|\mathbf{w} - \mathbf{w}_k\|_p$, the CL sends to ED k the federated model f_{FL} for replacing the local f_k with probability: $P\{\|\mathbf{w} - \mathbf{w}_k\|_p \geq \boldsymbol{\xi}\}$. Based on the Markov inequality, this probability is bound by $\frac{\mathbb{E}[\|\mathbf{w}-\mathbf{w}_k\|]}{\xi} = O(\min_{j=1,\ldots,d} \frac{1}{\xi_j})$. Hence, the overall communication complexity (messages sent from CL to ED k regarding model discrepancy and messages sent from ED k to CL for updating regularly the federated model at every epoch s) given a fixed time horizon T is $O(\frac{1}{T}(\min_{j=1,\ldots,d} \frac{1}{\xi_j} + s))$.

In the ASM process provided in Algorithm 3, the communication complexity is the same as EFM. Furthermore, the ASM requires $O(Md)$ time for local model update and storage $O(Md+L)$ for the data window $\mathcal{W}$ and rewards window $\mathcal{U}$, with sizes M and L, respectively.

5. Time-Optimized Model Selection

In this section, we deal with the challenge **R2**, as presented in Section 3.2, by introducing a time-optimized model selection (TOSM) scheme for optimally determining *when* to use the local f_k model or the federated f_{FL} model based on their prediction performance and *how* the ED k can decide on selecting one of them by maximizing an error-oriented reward, thus, securing high quality of predictive analytics tasks. TOSM copes with a time-based stochastic optimization problem whose optimal and unique solution relies on the principles of the Optimal Stopping Theory (OST).

The ASM model exploits the recent prediction performance of the local and federated models via a reward mechanism to swiftly determine on aggregating their predicted outcomes upon a prediction request. This is processed timely in $O(1)$ time (calculating the α ratio) requiring $O(L)$ storage (storing the recent θ values). In this section, we further introduce a time-optimized mechanism on finding *when* and *which* model should be used per prediction request overcoming the problem of aggregating the predictions of two models f_k and f_{FL} inside the ED. Such

mechanism, coined hereinafter as Time-Optimized Selection Model (TOSM) attempts to choose one of the two models reducing the probability of losing potentially local statistical dependencies learned from the local model and/or the generalizability prediction capability of the federated model. TOSM achieves an optimal scheduling of involving either f_k or f_{FL} based on accumulating sequential prediction errors that reflect the actual performance of each model individually. Hence, the right model is scheduled to be used for prediction at any given prediction request time. TOSM, instead of weighting the predictions, which potentially filters out with this aggregation the inherent variance of the predicted outcome y, optimally selects one of the two models for the prediction $\hat{y}$. This is achieved by identifying the *optimal* model at prediction time t. TOSM calculates locally in ED k at time t the prediction errors ϵ_L and ϵ_{FL} of the local f_k and federated model f_{FL}, respectively. Initially, at time $t = 0$, the chosen model is the f_{FL} ensuring generalizability of the initial prediction. Given the two prediction errors ϵ_L and ϵ_{FL}, TOSM decide *when* is the best time t^* to switch from the federated model f_{FL} to the local model f_L given cumulative comparative error evaluations. Fundamentally, TOSM at time t compares the prediction errors of f_k and f_{FL} transforming them to a reward binary variable $Z \in \{0, 1\}$ as follows:

$$Z_t = \begin{cases} 0 & \text{if } \epsilon_L > \epsilon_{FL}, \\ 1 & \text{if } \epsilon_L \leq \epsilon_{FL}. \end{cases} \tag{10}$$

Then, TOSM accumulates the rewards within an unknown finite time horizon to decide on switching to the f_k (f_{FL}) models. As past behavior and prediction quality are of high importance to the applications and predictive analytics tasks, the cumulative comparison of the predictability of both models, including the history divergence of these two models, are exploited for an optimal switching decision in TOSM. The history of the rewarded prediction performance comparison is represented by the cumulative sum of Z_t values, defined as R_t:

$$R_t = \sum_{\tau=0}^{t} Z_\tau. \tag{11}$$

Evidently, an instantaneous decision at t only over the current comparison between ϵ_{FL} and ϵ_L would not effectively demonstrate a robust historical behavior. On the other hand, a fixed time horizon observing the prediction behavior of both models cannot be determined beforehand, as proposed in the ASM model by adopting a L fixed-size window $\mathcal{U}$).

Hence, the problem that arises here is to find which is the optimal time horizon where TOSM can switch from f_k (f_{FL}) to f_{FL} (f_k) such that we achieve the best prediction performance of both models. In order to formally define this problem as a stochastic optimization problem (recall that Z and R are random variables), we introduce a delay factor $\beta \in (0, 1)$ over the cumulative sum of error comparisons in R_t, such that, our objective function becomes:

$$Y_t = \beta^t R_t = \beta^t \sum_{\tau=0}^{t} Z_\tau. \tag{12}$$

Y_t is a random variable whose maximization of its expectation across the time domain $t \in \mathbb{T} = \{1, 2, \ldots\}$ will indicate the *optimal model switching time* to decide when to switch from the federated model f_{FL} to the local model f_k. The delay factor $\beta \in (0, 1)$ indicates a (delay) tolerance level in observing the prediction performance of both models. If $\beta \to 1$, the tolerance is increased. We then formulate our problem for the TSOM model as follows:

Problem 1. *Given the local model f_k and federated model f_{FL} in an ED k, with prediction errors ϵ_L and ϵ_{FL} at time t, respectively, find the optimal model switching time t^*, where ED k switches from f_{FL} to f_k such that the following supremum of the expectation of Y_t is attained:*

$$\sup_{t \geq 0} \mathbb{E}[Y_t]. \tag{13}$$

In order to find the unique solution of the Problem 1, i.e., to find the optimal model switching time t^*, we cast this problem as an Optimal Stopping Time problem, which will be solved based on the principles of the OST. Before elaborate on our solution, we provide some preliminaries of the OST in Section 5.1 (the reader can skip this section should they be familiar with the OST fundamentals).

5.1. Optimal Stopping Theory

The fundamentals of OST [36, 37] lay in choosing the best time to take an action that maximizes an expected return or reward. The optimal decision making rule of stopping at the best time instance (a.k.a. optimal stopping rule) is determined over a sequence of the realizations of the random variables $Z_1, Z_2, \ldots$ and a sequence of the corresponding rewards that depend on the observed value of Z until time t. Such sequence of return functions can be defined as $(Y_t(Z_1, \ldots, Z_t))_{t>1}$. The decision maker (in our case, the ED k), is observing the sequence of Y_t, reflecting the error comparison values, and decides to either *stop* and switch the local/federated models, or *continue*. The ED k, by determine an optimal stopping rule, can then decide to switch the model between the federated and the local and vice versa maximizing the expected return defined in Equation (12). Our aim is to maximize the expected return or reward of the function Y_t when we decide to switch the models.

5.2. Optimal Model Selection based on OST

The ED k, by monitoring the prediction errors of both local and federated models, evaluates the reward function Y_t defined in Equation (12) and seeks for an optimal stopping rule that maximizes the expectation of Y_t, $\mathbb{E}[Y_t]$, given a fixed tolerance β value. In order to proceed with an optimal solution of the Problem 1, where the supremum of the expectation of Y_t can be attained, we need first to prove that the optimal stopping time t^* that maximizes (13) exists.

Theorem 1 (Optimal Model Switching Time Existence)**.** *The optimal model switching time t^* of the Problem 1 exists.*

Proof. We prove that the optimal model switching time exists, thus, the ED k can optimally switch between the models for achieve maximization of the expected reward. Based on the principles of the OST, to prove its existence, two conditions should be true: (C1) $\limsup_t Y_t \leq Y_\infty = 0$ is surely true, and (C2) $\mathbb{E}[\sup_t Y_t] < \infty$.

The condition C1 implies that with the elapse of time ($t \to \infty$), the reward should go to zero, i.e., $Y_\infty = 0$. Since no change of the model over an indefinite horizon is useless due to placement in constantly changing environments, $Y_\infty = 0$ represents the reward of an endless non-model switch phase. The supremum limit of Y_t is notated by $\limsup_t Y_t$, i.e., the limit of $\sup_t Y_t$ as $t \to \infty$ or $\lim_{t\to\infty}(\sup\{Y_j : j \geq t\})$. As Z_t is non-negative and using the strong law of numbers ($\frac{1}{t}\sum_{j=1}^{t} Z_j) \to \mathbb{E}[Z]$, we can derive that:

$$Y_t = t\beta^t(R_t/t) = t\beta^t(1/t)\sum_{j=1}^{t} Z_j \sim t\beta^t\mathbb{E}[Z] \overset{\text{a.s.}}{\to} 0. \tag{14}$$

This results to $\lim_{t\to\infty}\sup_t Y_t = 0$. As $Y_\infty = 0$ is by definition true, we then declare that C1 is satisfied.

The condition C2 implies that the expected reward under any policy (model switching rule) is finite. Therefore, C2 can be shown as:

$$\sup_t Y_t = \sup_t \beta^t \sum_{j=1}^{t} Z_j \leq \sup_t \sum_{j=1}^{t} \beta^j Z_j \leq \sum_{j=1}^{\infty} \beta^j Z_j. \tag{15}$$

This results into satisfying C2 with,

$$\mathbb{E}[\sup_t Y_t] \leq \sum_{j=1}^{\infty} \beta^j \mathbb{E}[Z] = \mathbb{E}[Z]\frac{\beta}{1-\beta} < \infty. \tag{16}$$

As both conditions C1 and C2 are satisfied, then the optimal model switching time t^* exists for the Problem 1. □

The next step in finding now the optimal model switching time is to prove that the existed t^* is also unique. This will guarantee the ED k that a unique optimal stopping rule can be implemented locally and be swiftly used for deciding when to switch the local to the federated model and vice verse.

Theorem 2 (Uniqueness of the Optimal Model Switching Time)**.** *The existing optimal model switching time t^* of the Problem 1 is unique.*

Proof. By proving the existence of the optimal time t^* in Theorem 1, we desire to find that this optimal stopping time in unique inside the ED k, which enables ED to decide on switching the models by maximizing the trade-off between their accuracy. Since Y_t is non-negative, the Equation (13) turn to be monotone [37]. Hence, the optimal model switching time t^* is then obtained by the one-stage look-ahead optimal rule (1-sla):

$$t^* = \inf\{t \geq 1 | Y_t \geq \mathbb{E}[Y_{t+1}]\}. \tag{17}$$

The adoption of 1-sla is optimal since $\sup_t Y_t$ has a finite expectation (equal to $\mathbb{E}[Z]\frac{\beta}{1-\beta}$) and $\limsup_t Y_t = 0$, as proved in Equation (14). Consequently, t^* is unique and can then be estimated through the principle of optimality. □

We have proved in Theorems 1 and 2 that the optimal model switching time exists and is unique for the Problem 1. Now, we elaborate on a methodology where ED k can define the optimal stopping time to estimate the optimal time t^*.

Theorem 3 (Optimal Model Switching Rule)**.** *Given the reward random variables $Z_1, Z_2, \ldots$, observed in ED k, the ED decides on switching the f_{FL} model to the f_k model at the first time instance t^* such that:*

$$t^* = \inf\{t \geq 1 | \sum_{i=1}^{t} Z_i \geq \frac{\beta}{1-\beta}\mathbb{E}[Z]\}. \tag{18}$$

Proof. Assume that the cumulative reward $R_t = r > 0$, when the ED k decides that it is optimal to switch the models. Then, the current reward of $\beta^t r$ is at least as large as any expected $\mathbb{E}[(\frac{\beta}{1-\beta})^{t+\tau}(r+R_\tau)]$. This means that $r(1-\mathbb{E}[(\frac{\beta}{1-\beta})^\tau]) \geq \mathbb{E}[(\frac{\beta}{1-\beta})^\tau R_\tau]$ for all times τ. This must hold true for all $r' \geq r$, so that the optimal time t^* for some r_0 must be of the form $t^* = \inf\{t \geq 1 | R_t \geq r_0\}$. Especially when the ED switches the first time t for which $R_t \geq r_0$, then the tolerance for forwarding r_0 must be the same as the tolerance for continuing using the 1-sla, therefore the sum of tolerances is positive. That is, r_0 must satisfy the equation

$$r_0 = \mathbb{E}[(\frac{\beta}{1-\beta})^\tau (r_0 + R_\tau)], \tag{19}$$

with $\tau = \inf\{t \geq 1 | R_\tau > 0\}$. Since Y is non-negative, it is possible to obtain $\tau \equiv 1$ and $R_\tau \equiv Y$ [37] and, then, replacing with $r_0 = \frac{\beta}{1-\beta}\mathbb{E}[Y]$. This will finally result in the definition of the optimal time t^* for switching from the federated model f_{FL} to the local model f_k defined as:

$$t^* = \inf\{t \geq 1 | \sum_{i=1}^{t} Z_i \geq \frac{\beta}{1-\beta}\mathbb{E}[Z]\}. \tag{20}$$

□

As it can be observed from Equation (18), the ED k needs to incrementally estimate the expectation $\mathbb{E}[Z]$ over the random variables Z_t in order to evaluate the optimal model switching rule. Specifically, we can write the expectation of the reward based on the conditional expectations given an event where the local model performs better than the federated model and vice versa. That is, we express the expectation $\mathbb{E}[Z]$ using the conditional expectations $\mathbb{E}[Z|\epsilon_L > \epsilon_{FL}]$ and $\mathbb{E}[Z|\epsilon_L \leq \epsilon_{FL}]$, i.e.,

$$\begin{aligned} \mathbb{E}[Z] \quad = \quad & \mathbb{E}[Z|\epsilon_L > \epsilon_{FL}]P(\epsilon_L > \epsilon_{FL}) + \\ & \mathbb{E}[Z|\epsilon_L \leq \epsilon_{FL}]P(\epsilon_L \leq \epsilon_{FL}). \end{aligned} \tag{21}$$

The expectation of Z, given by Equation (10), is then $\mathbb{E}[Z|\epsilon_L > \epsilon_{FL}] = 0$ and $\mathbb{E}[Z|\epsilon_L \leq \epsilon_{FL}] = 1$, which is:

$$\mathbb{E}[Z] \quad = \quad 1 - P(\epsilon_L < \epsilon_{FL}) = 1 - F_{\epsilon_L}(\epsilon_{FL}), \tag{22}$$

where F_{ϵ_L} is the Cumulative Distribution Function (CDF) of the ϵ_L prediction error $F_{\epsilon_L}(e) = P(\epsilon_L \leq e)$. Upon an incremental

estimation F_{ϵ_L} over sequential ϵ_L errors (as will be discussed later), the ED k is led to determine the optimal stopping time t^* switching the federated model f_{FL} to the local model f_k as follows:

$$t^* = \inf\{t \geq 1 | \sum_{\tau=1}^{t} Z_\tau \geq \frac{\beta}{1-\beta}(1 - F_{\epsilon_L}(\epsilon_{FL}^t)\}. \quad (23)$$

The ED k based on the Equation (23) switches the models at the first time instance t such that the cumulative sum of the rewards up to t exceeds the current quantity $(1 - F_{\epsilon_L}(\epsilon_{FL}^t))$, which depends on the current prediction error of the federated model ϵ_{FL}^t scaled by the odds factor $\beta(1-\beta)^{-1}$.

The TOSM model further considers an optimal rule for the *reverse* model switching, i.e., from the local model f_k to the federated model f_{FL}. Therefore, once the model is switched from the federated model to the local one, the cumulative reward sum R is set to 0. Then, the problem of finding the optimal model switching time t^* to switch back to the federated model is solved as explained in the above-mentioned methodology. In a similar way of solving the Problem 1, the optimal time to switch from the local model f_k to the federated model f_{FL} is based on the reverse reward random variable Q, in which the prediction errors ϵ for both models are monitored, i.e., similar to Z_t, we define Q_t:

$$Q_t = \begin{cases} 0 & \text{if } \epsilon_{FL} > \epsilon_L, \\ 1 & \text{if } \epsilon_{FL} \leq \epsilon_L. \end{cases} \quad (24)$$

From Equation (24), the expectation of Q, $\mathbb{E}[Q]$ is then similarly calculated via the two conditional expectations, i.e.,

$$\begin{aligned} \mathbb{E}[Q] &= \mathbb{E}[Q|\epsilon_{FL} > \epsilon_L]P(\epsilon_{FL} > \epsilon_L) + \\ &\quad \mathbb{E}[Q|\epsilon_{FL} \leq \epsilon_L]P(\epsilon_{FL} \leq \epsilon_L) \\ &= 1 - F_{\epsilon_{FL}}(\epsilon_L). \end{aligned} \quad (25)$$

From the expectation of Q in Equation (25), it is possible to derive the optimal time t^* to switch from the local f_k back to the generalized f_{FL}.

For reasons of completeness, we provide the optimal time t^* for the reverse switch in Equation (26), which is analogously determined as in Equation (23).

$$t^* = \inf\{t \geq 1 | \sum_{\tau=1}^{t} Q_\tau \geq \frac{\beta}{1-\beta}(1 - F_{\epsilon_{FL}}(\epsilon_L^t)\}. \quad (26)$$

Once the model is switched back to the federated model f_{FL}, the reward summation Q is set to 0, and the Z-values accumulation starts off. This method is performed until a new federated model f_{FL} is sent from the CL to the EDs, thus, replacing their old one.

Note, the decision making at time t, which is based on the optimal model switching rules in (23) and (26), is computationally lightweight requiring only $O(1)$ time (refer to Section 5.3). Specifically, the ED k needs only the CDF of the error values ϵ of the models f_k and f_{FL}, which can be incrementally estimated as elaborated in the Section 5.3.

The Algorithm 4 illustrates the information flow of the TOSM model regarding the optimal model switching decisions in the ED k from State `FL` (from f_{FL} to f_k) to State `L` (from f_k to f_{FL}) in lines 1 and 13, respectively, as long as the federated model f_{FL} has been updated and received locally. In Remark 2, we provide the computational, storage and communication complexity for the TOSM model in an ED k.

Algorithm 4 Time-Optimized Selection Model (TOSM)

```
1: reward sum R ← 0; State = FL
2: while TRUE do
3:    receive input-output (x_t, y_t) from SANs
4:    update local model f_k using (3)
5:    calculate errors ε_L and ε_FL
6:    calculate reward Z_t using (10)
7:    R ← R + Z_t
8:    if criterion in (23) is TRUE then
9:       switch f_FL with f_k
10:      break
11:   end if
12: end while
13: reward sum R ← 0; State = L
14: while TRUE do
15:   receive input-output (x_t, y_t) from SANs
16:   update local model f_k using (3)
17:   calculate errors ε_L and ε_FL
18:   calculate reward Q_t using (24)
19:   R ← R + Q_t
20:   if criterion in (26) is TRUE then
21:      switch f_k with f_FL
22:      break
23:      go to Line:1
24:   end if
25: end while
```

5.3. Incremental Estimation of CDFs F_{ϵ_L} and $F_{\epsilon_{FL}}$

The ED k incrementally estimates the probability density functions (or densities) $p_L(\epsilon)$ and $p_{FL}(\epsilon)$ based on the error prediction values ϵ_L and ϵ_{FL}, respectively. For simplicity of notation, we use ϵ for any prediction error (local or federated) in the following analysis since the methodology for estimating both CFDs F_{ϵ_L} and $F_{\epsilon_{FL}}$ is the same, thus, dropping the subscript of the density $p(\epsilon)$. Let us focus on estimating in an incremental way the CDF F_{ϵ_L}; the same holds true for $F_{\epsilon_{FL}}$. This is achieved by adopting the non-parametric Kernel Density Estimation (KDE) method over $n > 0$ recent error values $\{\epsilon_{t-n+\tau}\}_{\tau=1}^{\tau=n}$, such that $p(\epsilon)$ is estimated as:

$$\hat{p}(\epsilon) = \frac{1}{n \cdot h} \sum_{\tau=1}^{n} K\left(\frac{|\epsilon - \epsilon_{t-n+\tau}|}{h}\right), \quad (27)$$

where $h > 0$ is the bandwidth of the symmetric kernel $K(u)$ (integrating to unity). One of the most frequent adopted kernel function is the Gaussian, i.e., $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$. We rely on an incremental estimation of $\hat{p}(\epsilon; t)$ for $\tau = 1, \ldots$, by previous estimate $\hat{p}(\epsilon; t-1)$ and the current value ϵ_t. We recursively

obtain for $t = 1, \ldots$:

$$\hat{p}(\epsilon; t) = \frac{t-1}{th}\hat{p}(\epsilon; t-1) + \frac{1}{th}K\left(\frac{|\epsilon - \epsilon_t|}{h}\right) \quad (28)$$

Upon capturing ϵ_t at t, $\hat{p}(\epsilon; t)$ is incrementally estimated by $\hat{p}(\epsilon; t-1)$, thus, there is no need to store all the previous values for estimating $\hat{p}(\epsilon; t)$. Hence, at time t, in order to examine the criterion in the optimal model switching rules in Equations (23) and (26), we obtain that $F(\epsilon_t) = \int_0^{\epsilon_t} \hat{p}(u; t)\, du) \approx \hat{F}(\epsilon_t; t) = \frac{t-1}{th}\hat{F}(\epsilon_t; t-1) + \hat{I}(\epsilon_t)$. That is, we obtain an incremental estimation of the CDF $\hat{F}(\epsilon) = \int_0^{\epsilon} \hat{p}(u)\, du$, which requires at the time instance t the calculation of the integral quantity over the current error ϵ_t:

$$\hat{I}(\epsilon_t) = \frac{1}{th}\int_0^{\epsilon_t} K(\frac{|u - \epsilon_t|}{h})\, du. \quad (29)$$

The calculation of $\hat{I}(\epsilon_t)$ is $O(1)$ time by adopting the Gaussian kernel function. Such kernel function is mostly used due to its convenient mathematical properties and especially when dealing with estimation of a probability density function. It is shown in B. W. Silverman[2] that the optimal value of h for a Gaussian kernel is:

$$h^* = 1.06 \min(\hat{\sigma}, \frac{\hat{r}}{1.34})n^{-\frac{1}{5}}, \quad (30)$$

where $\hat{\sigma}$ is the standard deviation of the error samples, $\hat{r}$ is the interquartile range, and n is the number of training prediction error values.

Remark 2. *We elaborate on the complexity of the TOSM process provided in Algorithm 4. The TOSM updates the local model in $O(Md)$ time adopting SGD over the sliding data window $\mathcal{W}$ of size M, with storage $O(Md)$. The calculation of the optimal model switching rules in (23) and (26) is $O(1)$ due to incremental update of the KDE-based CDF in (28) and (29) of the prediction errors of the local and federated models. The communication complexity is the same as EFM given the model updates to/from CL (refer to Remark 1).*

6. Performance Evaluation

6.1. Experimental Setup

In order to assess the performance of our methods, we performed experiments on a multivariate dataset (DS) that contains 415 weather stations around the United Kingdom (UK) measuring sequential contextual data streams of the surrounding environment. These data streams have been collected over the time horizon of December 2017 until March 2018 using the API of Wunderground [38]. Each weather station represents the kth ED with $k \in \{1, \ldots, K\}$ with $K = 415$. Each time t, EDs received a d-dimensional input data vector $\mathbf{x}_t$. The DS provides a 9-dimensional data streams in the form of $\{(\mathbf{x}_t, y_t)\}$ including temperature, dew point, humidity, wind-speed, wind-gust, wind direction, pressure, windchill, and precipitation. The output y_t is set with the DS's measurement of temperature, while the remaining measurements are used for the input $\mathbf{x}_t$. The function $y = f(\mathbf{x})$ is a multivariate linear regression with $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}; \mathbf{w} \in \mathbb{R}^{d+1}$ resulting in minimizing the objective of Equation (1) into:

$$\mathcal{J}(\mathbf{w}) = \min_{\mathbf{w}\in\mathbb{R}^{d+1}} \frac{1}{T}\sum_{t=1}^{T}\left(y_t - (\mathbf{x}_t)^\top\mathbf{w}\right)^2 + \lambda, \|\mathbf{w}\|^2 \quad (31)$$

where λ is the regularization parameter. The data collection frequency is every 5 minutes over the time horizon of 100 days, resulting in a dataset size of $N = 9,044,683$, assembling roughly 250 values measured per ED and per day. All data are normalized and scaled, i.e., each dimension $x \in \mathbb{R}$ is mapped to $\frac{x-\mu}{\sigma}$ with mean value μ and variance σ and scaled in the unity interval, thus, $\mathbf{x} \in [0, 1]^d$.

A training period of 1 month is considered before testing the proposed methods. This corresponds to the splitting of DS of size $N = N_T + N_M$, which results in $N_T = 1,500,000$ data points for training and around $N_M = 7,500,000$ data points for testing the prediction capability of each approach. Converting this into a percentage, only 15.8% of the collected values are used during training to overcome the cold-start problem using transfer learning, and over 84% is used for testing the different approaches.

The starting date is located on the 1st of January 2018 and presents the time $t = 0$. During the training period, each ED k generates a local model f_k. At time $t = 0$, each ED sends its local model f_k towards the CL. Inside the CL, these models are merged using the baseline FedAVG model [11] (see Equation (4)) for comparison towards the first federated central model f_{FL}. At time $t = 0$, f_{FL} is distributed across all the EDs as their first federated model.

6.2. Performance Metrics

We assess our methods, the baseline models and the models under comparison with respect to two categories of performance metrics for *accuracy* and *information loss*, which are widely used for assessing predictive analytics tasks.

For assessing accuracy, we use three metrics acknowledged in the literature: (1) Mean Absolute Error (MAE) $= \frac{1}{N}\sum_{n=1}^{N}|\hat{y}_n - y_n|$; (2) Root Mean Squared Error (RMSE) $= [\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2]^{1/2}$; and (3) Symmetric Mean Absolute Percentage Error (SMAPE) $= \frac{100}{T}\sum_{t=1}^{T}\frac{|\hat{y}_t - y_t|}{|y_t| + |\hat{y}_t|}$, because of its unbiased properties and ability to compare the results representing in percentage with values in $[0, 100]$.

For information loss in predictive analytics, we use the Kullback-Leibler (KL) divergence. The KL divergence from $p(\mathbf{x})$ to $p(\tilde{\mathbf{x}})$ denotes the information loss when attempting to reconstruct sequential data (data streams) $\tilde{\mathbf{x}}$ for the actual data stream $\mathbf{x}$, using $p(\tilde{\mathbf{x}})$ and $p(\mathbf{x})$ as the probability distribution functions, respectively. KL is defined as:

$$KL(p(\tilde{\mathbf{x}})\|p(\mathbf{x})) = \int_{\mathcal{X}\subset\mathbb{R}^d} p(\tilde{\mathbf{x}}) \log \frac{p(\tilde{\mathbf{x}})}{p(\mathbf{x})} d\mathbf{x}. \quad (32)$$

[2] B. W. Silverman. 1986. Density Estimation for Statistics and Data Analysis. Chapman and Hall, London, 1986.

Parameter	Notation	Value/Range	Optimal Setting
$(\mathbf{x}_t, y_t)$	Data input vector and output at time t		
$t, \tau, t^* \in \mathbb{T}$	Discrete time instances		
ϵ	Absolute prediction error		
$p(x)$; $F(x)$	Density function; Cumulative Distribution Function		
$\mathbf{w}$	Parameters of ML model		
f_{FL}, f_k	Federated model, Local model at ED k		
$\mathcal{U}$	Sliding window of rewards θ in EDs		
$\mathcal{W}$	Sliding window for data vectors $\mathbf{x}$ in EDs		
$D = \{D_{k=1}^K\}$	Complete dataset, local datasets at EDs		
Z, Q	Rewards of local and FL models (TOSM)		
K	Number of EDs	415	
d	Data dimensionality	$\{2, \ldots, 9\}$	
N	Size of complete dataset D	9,000,000	
N_T; N_M	Training data; testing data	1,500,000;7,500,000	
α	Model weighting between f_{FL} and f_k (ASM)	(0,1)	
θ	Reward (ASM)	$\{0, 1\}$	
s	Model update epoch (local and/or federated model)	$\{1, 2, 4\}$	1
M	Size of data sliding window $\mathcal{W}$	$\{250, 500, 1000\}$	1000
L	Size of reward sliding window $\mathcal{U}$	$\{50, 100, 250, 500\}$	50
β	Delay tolerance factor (TOSM)	(0,1)	0.3
η	Learning rate in SGD	0.1	0.1

Table 2: Notations and ranges/values of basic parameters.

6.3. Baseline Models & Models under Comparison

A baseline model for comparison is required to evaluate the performance of our methodologies. Specifically, during the assessment, the performance of the **Global Model (G)**, which transmits raw data streams from the EDs to the CL, at time instance t is used as a baseline comparison model towards the others. Furthermore, focusing on the distributed nature of generating a federated global model in the CL based on the local ML models, we compare our methods with the basic deployment of FL, coined here as the **Federated Model (FM)** proposed in [11]. In addition, we compare our methods with the locally trained model **Local Model (L)** proposed in [12], which does not involve any central coordinator or generalized model. The model L is built from scratch without any previous model(s) received from the CL. The **Evolving Federated Model (EFM)** uses as the initial local model f_k at time $t = 0$ the received generalized model f_{FL}. At each selected epoch, the CL requests the local models of the selected EDs and merges the generalized f_{FL} over them. Only the model f_k is stored inside the ED. Moreover, the proposed **Local Federated Model (LFM)** extends the EFM and partially extends the functionality of learning methodology in [14] for comparative assessment reasons. Specifically, the LFM merges the local models in the CL provided in [14] by updating the generalized model f_{FL} locally at each time t until the new update from the CL is sent at a predefined epoch. The **Adaptive Selection Method (ASM)** introduces a dual parallel model implementation inside each ED. A local model f_k is initially set to $f_k = f_{FL}$ and is continuously updated each time t. The federated model f_{FL} is received and updated as the model FM. The final prediction is generated through a reward function with respect to the historical performance of each model and balancing the f_k and f_{FL} model through the parameter α. We also compare with the **Smoothed Model (SM)** in [13], which is based on the same concept as the ASM but with a fixed value for the balancing weight α. Finally, the **Time-Optimized Selection Method (TOSM)** provides a selecting mechanism of the optimal model at any given time. The fundamental concept is based on finding the optimal model switching time t^* to switch between the two models f_k and f_{FL} and vice versa using OST.

6.4. Parameters Configuration

In order to assess the accuracy, multiple parameters for the different models and ED storage capacities have to be set. After the starting point ($t = 0$), at each time t, the models EFM, LFM, L are updated using SGD. The values for the reward θ are inserted into the window $\mathcal{U}$ each time t for the model ASM. The adaptive weighting α is generated through the ratio over $\mathcal{U}$ of size $L \in \{50, 100, 250, 500\}$. Moreover, for the model TOSM, the Z_t and Q_t values are determined at each time t and the respective current optimal model flagged inside each ED k. The delay tolerance factor for TOSM β takes values from the range $\beta = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. If the CL requests at a specific epoch an update of the model FM, the values inside the window $\mathcal{W}$ are used for the SGD update. The size M of the window $\mathcal{W}$ takes values in the range $M = \{250, 500, 1000\}$. The model update epoch is set for the assessment to be every day, every other day and every fourth day, with $s \in \{1, 2, 4\}$, respectively.

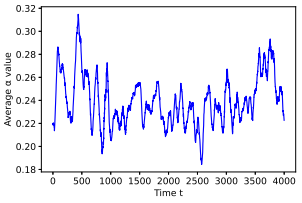
Assessing the overall performance of each model, a type of cross-validation has been deployed to guarantee independent validation of the results. The application is stopped at 24 random time points t. At the selected time t, the methodology implemented is stopped and the next 250 values (representing one day) of each ED k is used as prediction input to analyze the performance of each model.

6.5. Performance & Comparative Assessment

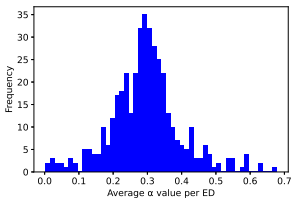
6.5.1. Personalized & Efficient Local Learning

We first analyze the impact of α on weighting between local model prediction using f_k and the federated model f_{FL} for the

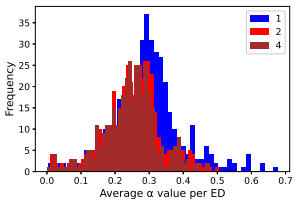
weighted prediction $\hat{y}$ using the model ASM. The influence of the model update epoch s, where the CL requests and updates the f_{FL} model, and the size L of the rewards value window $\mathcal{U}$ on the model weighting parameter α using the introduced performance metrics is shown in Figure 2.



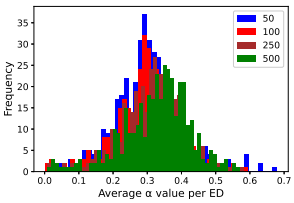
(a) Average value of α over time t.



(b) Histogram of α in EDs.



(c) Histogram of α in EDs conditioned on the federated update frequency (epoch) s.

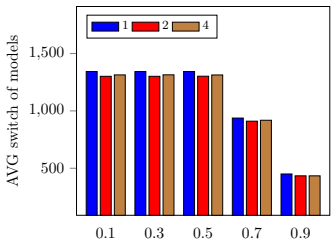


(d) Histogram of α in EDs conditioned on the window $\mathcal{U}$'s size L.

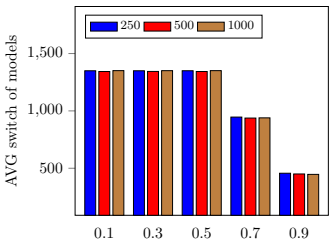
Figure 2: Parameter setting influence on α for the model ASM.

In Figure 2 (a) the average value of α over all EDs until $t = 4000$ is presented. It can be observed that the value of α is greatly changing over time but lies in the range of $\alpha \geq 0.18$ and $\alpha \leq 0.32$. This figure has been using the setting of $L = 50$, $M = 1000$, and update epoch s to be every day with $s = 1$. The same settings are used for the frequency analysis and α distribution over each ED illustrated in Figure 2 (b). In this figure, one can observe the distribution of average α values for each ED, which can be approximated as a normal distribution with the mean around $\alpha = 0.3$. This correlates with the findings of Figure 2 (a) for the average α-values per time instance t over all the EDs lying in the highlighted range. To identify the influence of the update epoch s and window size L towards the weighting parameter α, Figure 2 (c) and Figure 2 (d) highlight this, respectively. In Figure 2 (c), the setting of $s = \{1, 2, 4\}$ indicating the update frequency of the federated model f_{FL} to be each day, every second day, and every fourth day. From this figure, the influence of the update frequency towards the model weighting α indicated that increasing the frequency is decreasing the mean of α and increases the variance. This is in opposite to the influence of window size L to α as shown in Figure 2 (d). In this figure, an increase of the mean is presented by increasing the window size of rewards. The increase in L only influences the mean but not the variance for the value α in each ED k.

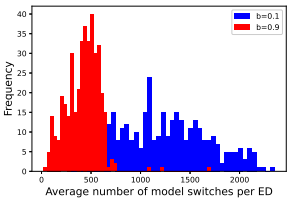
We further investigate the behavior of the model TOSM by experimenting with the factor β for the delay tolerance of switching the models f_k and f_{FL} to use the optimal model at the prediction time. This value of β is analyzed in Figure 3 with the number of model switches that occur during the runtime. In Figure 3 (a), the influence of switching the models dependent on β with increasing the update frequency of model epoch s.



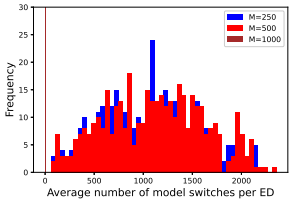
(a) Average switching time for different β and federated model update frequency s.



(b) Average switching time for different β and size M of data window $\mathcal{W}$.



(c) Histogram of model switching times for $\beta = \{0.1, 0.9\}$; $M = 250$.



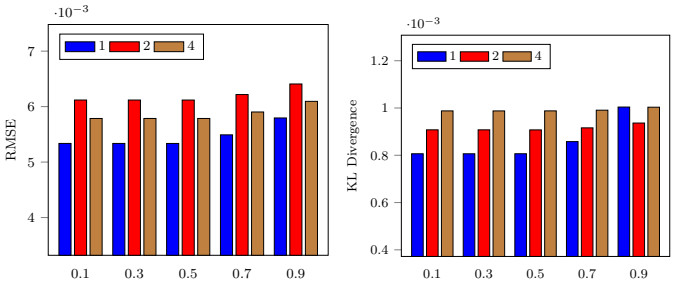
(d) Histogram of model switching times for $\beta = \{0.5\}$; $M = \{250, 500, 1000\}$

Figure 3: Parameter setting influence on β for the model TOSM.

In this figure, no difference between the variation of model update epoch s can be seen influencing the model switching in the model TOSM. However, a decrease of average switches can be seen when the value of $\beta \geq 0.7$. Similar results are highlighted in Figure 3 (b). This figure indicates the influence of increasing the size M of the (data) window $\mathcal{W}$ with respect to the factor β and the number of times the model is switched inside the ED. Figure 3 (c) shows the distribution of $\beta = 0.1$, the number of switches inside each ED and the distribution of $\beta = 0.9$ with delaying the switching. With a $\beta = 0.9$, the variation of average switches is higher than with $\beta = 0.1$, in which the density is around the mean of 500. In Figure 3 (d), the distributions of $\beta = 0.1$ using different values of M are presented. As already shown in Figure 3 (b), no change of the frequency by differing the size of M can be observed.

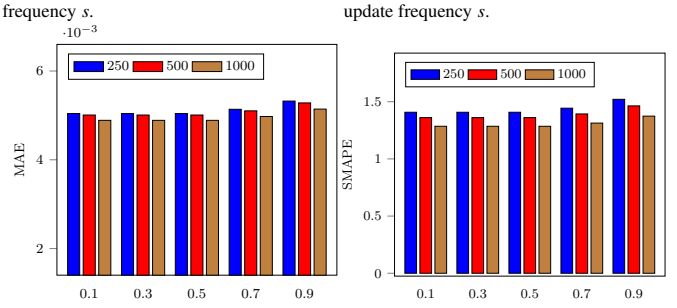
In Figure 4, the performance metrics RMSE, MAE, SMAPE and KL divergence over different β values are investigated. Highlighted in the previous figure, increasing the delay of switching between the models f_k and f_{FL} inside the ED indicated through the value of β shows that $\beta \rightarrow 1$ increases the tolerance and, thus, less often the models are switched.

In Figure 4 (a), the analysis of RMSE with respect to the model update frequency s is illustrated. One can observe that the update frequency s is only slightly increasing the RMSE. Similar results have been obtained using MAE and SMAPE, which are not shown due to space limitations. Moreover, in this figure, the behavior of update epoch s towards the accuracy is illustrated. Using $s = 1$ indicating an update frequency of the f_{FL} model every day, results in the lowest accuracy error. However, by assuming that increasing the frequency is rising the error does not hold true as the performance of $s = 4$ (update frequency every fourth day) generates lower prediction errors than $s = 2$. The KL divergence highlighted in Figure 4 (b) indicates a clear decrease of information loss by increasing the update frequency s. Additionally, the influence of β towards the



(a) RMSE for different β and model update frequency s.

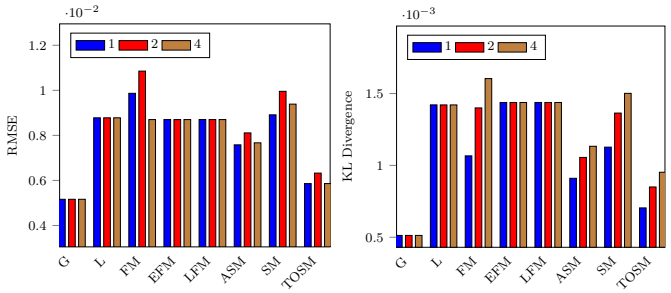
(b) KL divergence for different β and model update frequency s.



(c) MAE for different β and (data) window $\mathcal{W}$ size M.

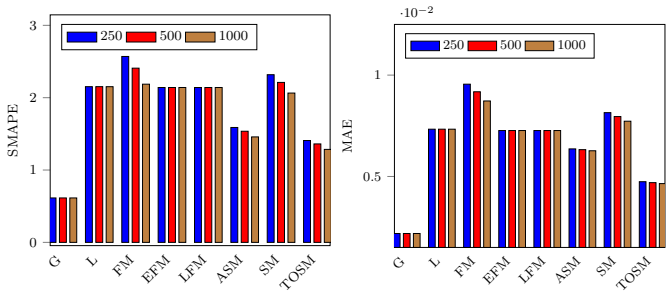
(d) SMAPE for different β and (data) window $\mathcal{W}$ size M.

Figure 4: Comparison of the influence of delay tolerance β for model TOSM against model update epochs s and window size M towards the metrics KL, RMSE, MAE, and SMAPE.



(a) RMSE per comparative model with $L = 250$ and $M = 250$.

(b) KL divergence per comparative model with $L = 50$ and $M = 1000$.



(c) SMAPE per comparative model with $L = 50$ and $s = 1$.

(d) MAE per comparative model with $L = 250$ and $s = 4$.

Figure 5: Comparative assessment over all models against different settings for model update epoch s and data window size M using the performance metrics KL divergence, RMSE, MAE, and SMAPE.

information gain shows that using more frequent switching of models results in higher entropy, presented by the KL metric, inside each ED.

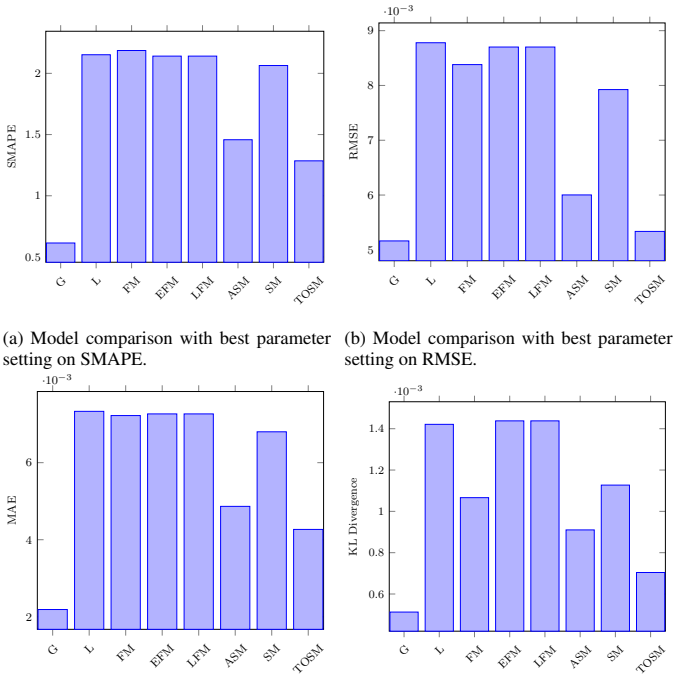
Figure 4 (c) and Figure 4 (d) investigate the behavior of increasing the window $\mathcal{W}$'s size M representing the data used for performing SGD at the model update epoch in order to update the f_{FL} to the CL. In Figure 4 (c), the influence is presented via the MAE metric. Increasing the size M to $M = 1000$ results in decreasing the error. The dependency of β with respect to the accuracy over different window sizes M does show a slight increase of MAE, when increasing the delay tolerance of $\beta \to 1$. In Figure 4 (d), similar behavior and dependency are observed. By increasing $\beta \to 1$, results in increasing the SMAPE, while by increasing the window $\mathcal{W}$'s size M to $M = 1000$, results in decreasing the prediction error.

In Figure 5, the performance of all models under comparison introduced in Section 6.3 and the influence of the parameters model update epoch s and data window size M are illustrated for comparative assessment. In Figure 5 (a), the RMSE for size $M = 250$ and rewards window size $L = 250$ over all epoch values s is highlighted. In this figure, the influence of epoch s towards the models FM, ASM, SM and TOSM is illustrated. The models G, L, EFM, and LFM do not change their performance by changing the parameters of M and s as their learning and adaptation to the data input is continuously and independent of the FM. Increasing the frequency of model epoch s results in increasing the prediction error for the methods SM and ASM. Whereas, the accuracy is increasing for the FM method. The TOSM model is, as evidenced in Figure 4, only slightly increasing and performing worst for epoch $s = 2$, representing updating every other day. TOSM depends only on s with respect to the update frequency of the updated model f_{FL} from the CL on the ED. That is, in the TOSM, the behaviors of the previous version of f_{FL} and of the new version of f_{FL} are both captured by the prediction accuracy compared with that of the local model. Even if, a new/updated f_{FL} is received by ED, the OST mechanism for deciding on the optimal model switching time is not enforced to proceed with changing the local model with the new f_{FL} model. The arbitrary model epoch parameter s, which is either application specific or dependent on the data, does not enforce TOSM to swap models in ED. Instead, TOSM assesses the prediction capability of the currently selected model given its recent history along with its counterpart model, which optimally decides on a model change event maximizing the expectation of the objective function in (13). And, this is the superiority of the TOSM compared to the models depending on the arbitrary parameter s. Notably, this figure shows that, evidently, the G model generates the best accuracy as expected, and the TOSM model provides accuracy closest to the G model. In Figure 5 (b), the information loss over changing the model update epochs s over all models is highlighted. Using the setting of $M = 1000$ and $L = 50$ with the performance metric KL divergence, the results show that for the models depending on these parameters (FM, ASM, SM and TOSM) the KL divergence increases with the increase of the model epoch s. A greater information loss can be observed for all these four methods comparing $s = 1$ and $s = 4$ with each other. Moreover, the information loss is the smallest when transmitting raw data and generating a centralized model (G). Given the information loss

performance metric, it is worth mentioning here, the behavior of TOSM against the model epoch s. As explained above, the prediction capability of TOSM is based on the model swap decision either involving an updated f_{FL} model or not. Nonetheless, once the model update frequency is getting higher (i.e., a relatively small s value), then an ED receives updated versions of the f_{FL} with high frequency. These recently updated models, evidently, are more up-to-date and capture the current data trends (as aggregated by the edge nodes in the network). Hence, statistically, their corresponding prediction accuracy, recorded in the TOSM during the OST mechanism, triggers the ED to swap to the most accurate model (given the relative ranking of the prediction errors between local and FL model versions). This is reflected in Figure 5 (b) where TOSM achieves less information loss with smaller s value, providing evidence of this parameter's influence. Figure 5 (c) and Figure 5 (d) highlight the performance towards changing the data window $\mathcal{W}$ size M and the corresponding behavior of each model. Setting the parameters for Figure 5 (c) with epoch $s = 1$ and the rewards window $\mathcal{U}$ size $L = 50$ using SMAPE shows the following: over all the assessed models, an increase in the training dataset size in window $\mathcal{W}$ inside each ED k for SGD presented by M results in decreasing the prediction error. In Figure 5 (d), similar behavior is illustrated with model epoch $s = 4$ and $L = 250$ using MAE. However, it should be highlighted that TOSM, even with a relatively high s parameter reflecting a smaller model update frequency from the CL, shows overall better performance than any other comparable model. Again, as above-mentioned, TOSM given any arbitrary parameter s, which can be e.g., application specific or tailored to stationary/non-stationary data, optimally decides on the most accurate model to be invoked for predictions at any given time under our OST-based objective function.

After identifying the influence of the data window $\mathcal{W}$ size M, the rewards window $\mathcal{U}$ of size L for the ASM model, and the centrally defined model update epoch s, towards the models in Section 6.3 with respect to their performance, the following best settings have been adopted over the four performance metrics KL divergence, RMSE, MAE, and SMAPE and the results are shown in Figure 6. Figure 6 uses the setting of $M = 1000$ as shown in Figure 5 performing the lowest prediction error over all models, $L = 50$ as highlighted in Figure 2 for minor variance in the average α value, model epoch $s = 1$ indicating an update of the FM model of each day, and $\beta = 0.3$ analyzed in Figure 4 and Figure 3 to be the most accurate values with respect to the performance metrics.

In Figure 6 (a), the SMAPE over all the models under comparison is provided using the best parameter settings. As it is evidenced from the experiments, the aim is to identify which model is closest to the G with respect to performance metrics. Even though the G is not suitable for IoT, EC applications, and distributed environments, the aim is to have quality-aware and efficient models inside the EDs performing as accurate as of the G. Having this aspect in mind, the TOSM model based on the OST, which identifies the optimal time to switch between the local generated model f_k (L) and the generalized model f_{FL} (FM), shows the best performance. However, TOSM still increases



(a) Model comparison with best parameter setting on SMAPE.

(b) Model comparison with best parameter setting on RMSE.

(c) Model comparison with best parameter setting on MAE.

(d) Model comparison with best parameter setting on KL divergence.

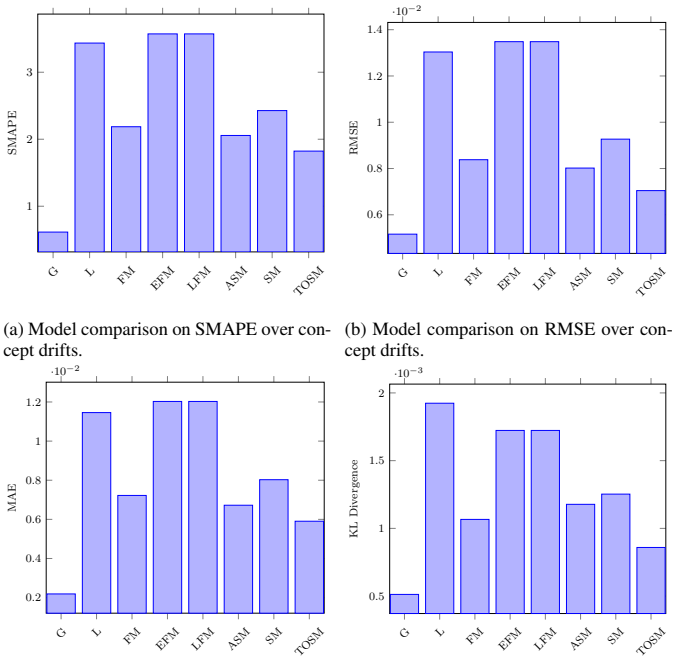
Figure 6: Model Comparison using the best parameter settings towards the performance metrics KL divergence, RMSE, MAE, and SMAPE

the error by 1%. The second best with regards to the performance metric SMAPE is the ASM model. The other models do not show a significant difference. Figure 6 (b) illustrates the analysis using RMSE. In this figure, the proposed models differ from each other. Highlighted in this figure is the great performance of TOSM with RMSE very close to the G. Moreover, the prediction accuracy performance of the FM model is better than that of the L model. The ASM model provides next to TOSM the best accuracy for predictive analytics. Figure 6 (c) and Figure 6 (d) support these findings for the performance metrics MAE and KL divergence. Especially, regarding the KL divergence, we can observe that the adaptive weighting of ASM and the optimal model switching of TOSM are the closest to the G model.

6.5.2. Models Behavior on Concept Drifts

In this section, we investigate the models' capability of being adaptive and evolving during learning inside EDs, which supports qualitative predictive analytics in data streaming environments. We experiment with concept drifts over the multi-dimensional data streams to examine the ability of each model under comparison towards changing environments and quality-aware predictions. In Figure 7, the performance across all metrics: KL divergence, RMSE, MAE, and SMAPE is illustrated using the identified best parameter settings of Figure 6. The values are compared against each other, showing the change of accuracy and information loss with concept drift appearances towards regular predictive tasks.

Figure 7 (a) illustrates the performance of all models with respect to SMAPE. The illustration shows a clear improvement



(a) Model comparison on SMAPE over concept drifts.

(b) Model comparison on RMSE over concept drifts.

(c) Model comparison on MAE over concept drifts.

(d) Model comparison on KL over concept drifts.

Figure 7: Model Comparison using the best parameter settings over concept drifts against the performance metrics KL divergence, RMSE, MAE, and SMAPE

of the FM performance when concept drift occurs. However, the adaptive and parallel model adaption of ASM and TOSM perform equal or better than the FM model. This indicates the ability to adapt to changing environments when using the generalized model and incorporating the local individualized model towards predictions. Moreover, it should be noted that the models L, EFM, and LFM highly increase their prediction error. In Figure 7 (c), similar behavior towards the approaches is illustrated assessed by the MAE metric. Figure 7 (b) shows the RMSE during the concept drift appearances and highlights the relatively satisfactory adaptation of the ASM model with similar performance to the presented results in Figure 6 (b) for familiar data inputs inside the EDs. The FM and ASM models generate similar prediction results, which indicates that the adaptive parameter α places more importance on the FM as the L model cannot adapt fast to concept drifts. Figure 7 (d) provides insights into the information loss by showing the KL divergence. The value of KL divergence for the FM model improves through the concept drifts, showing the importance of generalization inside EDs. Some improvement of the information loss value is also achieved by the ASM model, as it highly depends on the accuracy of the FM or LM (depending on the weighting factor α). However, the method of TOSM provides constantly low information loss independent of the occurrence of the concept drift, indicating its applicability in edge computing settings over data streaming environments.

7. Conclusions

The focus on personalized and efficient predictive analytics in resource constraint environments has been investigated. It has been shown through related work on local edge learning, that Federated Learning introduced local learning over local data by design. The research community leaves open questions towards the quality and efficiency of Federated Learning under changing environments. In this paper, two fundamental strategies have been proposed that enable the ability to centrally learn a predictive model and enhancing the quality of local inferred and predictive results. Quality of analytical results through enabling the local individuality of heterogeneous devices provided the fundamentals of these approaches. The first model, Adaptive Selection Model, uses the local model and generalized model to provide a new prediction outcome by weighting these two models based on historical rewards. The second strategy (TOSM) introduces the optimization to find the best (optimal) time to switch between the local model and the generalized federated model by using Optimal Stopping Theory. Theoretical and complexity analysis of the optimality and uniqueness of the solution is provided showcasing the time-optimized and lightweight process suitable for the EDs capacity. Comparative and performance evaluation has been comprehensively provided across different methods found in the literature. Furthermore, it was possible to provide evidence that a switching strategy between models inside EDs enables qualitative predictive analytics for continuous changing and evolving environments (including concept drifts). The highlighted strategy is included in our research projects GNFUV[3] in which we introduced an edge-centric communication reduction mechanism and local learning methodologies applied and tested over a swarm of intercommunicated Unmanned Surface Vehicles (USVs) in [13]. In this context, USVs equipped with humidity and temperature sensors were acting as mobile edge nodes locally building and updating ML models for environmental monitoring applications [16], specifically monitoring the sea surface in a coastal area in Athens.

Our future research agenda focuses on hierarchical structuring of Federated Learning models to enable not only personalized models at each ED individually, but enable group-based models based on their *similarities*. Moreover, the combination of active learning for continuous changing environments and Federated Learning is highly interesting to provide qualitative analytics for real-time applications over semi-supervised learning.

CRediT authorship contribution statement

Natascha Harth: Conceptualization, Methodology, Formal analysis, Writing- original draft, Visualization, Investigation, Software, Validation. **Christos Anagnostopoulos:** Conceptualization, Formal analysis, Writing - review & editing, Supervision. **Kostas Kolomvatsos:**Writing - review & editing,

[3] http://www.dcs.gla.ac.uk/essence/funding.html#GNFUV

Supervision. **Hans-Joerg Voegel:** Writing - review & editing, Supervision.

References

[1] REGULATIONS REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), Tech. rep.

[2] California Consumer Privacy Act (CCPA), AB-375, Tech. rep.

[3] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, H. B. McMahan, Adaptive federated optimization, in: International Conference on Learning Representations, 2021.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. B. et al., Advances and Open Problems in Federated Learning, Foundations and Trends® in Machine Learning 14 (1).

[5] A. F. Dakhil, W. M. Ali, A. A. Abdulredah, Predicting prior engine failure with classification algorithms and web-based iot sensors, in: 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), 2020, pp. 1–6.

[6] M. Al-Zeyadi, J. Andreu-Perez, H. Hagras, C. Royce, D. Smith, P. Rzonsowski, A. Malik, Deep learning towards intelligent vehicle fault diagnosis, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–7.

[7] M. Martínez, J. Echanobe, I. del Campo, Driver identification and impostor detection based on driving behavior signals, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 372–378.

[8] J. Hu, X. Zhang, S. Maybank, Abnormal driving detection with normalized driving behavior data: A deep learning approach, IEEE Transactions on Vehicular Technology 69 (7) (2020) 6943–6951.

[9] B. I. Kwak, J. Woo, H. K. Kim, Know your master: Driver profiling-based anti-theft method, in: 2016 14th Annual Conference on Privacy, Security and Trust (PST), 2016, pp. 211–218.

[10] A. Bichicchi, R. Belaroussi, A. Simone, V. Vignali, C. Lantieri, X. Li, Analysis of road-user interaction by extraction of driver behavior features using deep learning, IEEE Access 8 (2020) 19638–19645.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: 20th International Conference on Artificial Intelligence and Statistics, Vol. 54, PMLR, 2017, pp. 1273–1282.

[12] C. Anagnostopoulos, Edge-centric inferential modeling and analytics, Journal of Network and Computer Applications 164 (September 2019) (2020) 102696.

[13] N. Harth, C. Anagnostopoulos, Quality-aware aggregation amp; predictive analytics at the edge, in: 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 17–26.

[14] P. P. Liang, T. Liu, L. Ziyin, R. Salakhutdinov, L.-P. Morency, Think Locally , Act Globally : Federated Learning with Local and Global Representations, in: Workshop on Federated Learning (NeurIPS), 2019, pp. 1–17.

[15] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: AISTATS, 2017.

[16] N. Harth, C. Anagnostopoulos, Edge-centric efficient regression analytics, in: IEEE International Conference on Edge Computing, EDGE 2018 - Part of the 2018 IEEE World Congress on Services, IEEE, 2018, pp. 93–100.

[17] V. Kulkarni, M. Kulkarni, A. Pant, Survey of personalization techniques for federated learning, in: 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2020, pp. 794–797.

[18] V. Zantedeschi, A. Bellet, M. Tommasi, Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs, in: International Conference on Artificial Intelligence and Statistics (AISTATS), Vol. 108, 2020, pp. 864–874.

[19] P. Vanhaesebrouck, A. Bellet, M. Tommasi, Decentralized collaborative learning of personalized models over networks, in: International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.

[20] A. Bellet, R. Guerraoui, M. Taziki, M. Tommasi, Personalized and private peer-to-peer machine learning, International Conference on Artificial Intelligence and Statistics (AISTATS) 84 (2018) 473–481.

[21] X. Yuan, X. Ma, L. Zhang, Y. Fang, D. Wu, Beyond class-level privacy leakage: Breaking record-level privacy in federated learning, IEEE Internet of Things Journal (2021) 1–1.

[22] A. Fallah, A. Mokhtari, A. Ozdaglar, Personalized federated learning: A meta-learning approach, in: Conference on Neural Information Processing Systems (NeurIPS), 2020.

[23] Y. Jiang, J. Konečný, K. Rush, S. Kannan, Improving Federated Learning Personalization via Model Agnostic Meta Learning, arXiv preprint arXiv:1909.12488.

[24] F. Hanzely, P. Richtarik, Federated learning of a mixture of global and local models, in: International Conference on Learning Representations (ICLR 2021), 2021.

[25] Y. Deng, M. M. Kamani, M. Mahdavi, Adaptive personalized federated learning, arXiv preprint arXiv:2003.13461.

[26] J. Chen, X. Ran, Deep Learning With Edge Computing: A Review, Proceedings of the IEEE 107 (8).

[27] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, K. Chan, Adaptive Federated Learning in Resource Constrained Edge Computing Systems, IEEE Journal on Selected Areas in Communications 37 (6) (2019) 1205 – 1221.

[28] F. Sattler, S. Wiedemann, K. R. Muller, W. Samek, Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data, IEEE Transactions on Neural Networks and Learning Systems 31 (9) (2020) 3400–3413.

[29] A. Li, J. Sun, B. Wang, L. Duan, S. Li, Y. Chen, H. Li, Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets, in: ACM/IEEE Symposium on Edge Computing (SEC), 2021.

[30] C. Anagnostopoulos, Edge-centric inferential modeling and analytics, Journal of Network and Computer Applications 164, 2020.

[31] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, IGI global, 2010, pp. 242–264.

[32] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359.

[33] A. Bifet, R. Gavaldà, Learning from time-changing data with adaptive windowing, SIAM International Conference on Data Mining (2007) 443–448.

[34] T. G. Dietterich, Machine learning for sequential data: A review, in: Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), Springer, 2002, pp. 15–30.

[35] J. Durbin, A simple and efficient simulation smoother for state space time series analysis, Biometrika 89 (3) (2002) 603–616.

[36] A. N. Shiryaev, Optimal stopping rules, Vol. 8, Springer Science & Business Media, 2007.

[37] H. Robbins, D. Sigmund, Y. Chow, Great expectations: the theory of optimal stopping, Houghton-Nifflin 7 (1971) 631–640.

[38] Weather Underground.
URL `https://www.wunderground.com/api/`