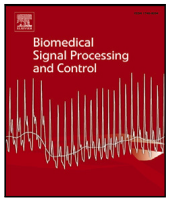
Biomedical Signal Processing and Control 87 (2024) 105416

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc



Model-Free-Communication Federated Learning: Framework and application to Precision Medicine

I. De Falco [a], A. Della Cioppa [b,a,*], T. Koutny [c], U. Scafuri [a], E. Tarantino [a]

[a] ICAR-National Research Council of Italy, Via P. Castellino, 111, Naples, 80131, Italy
[b] NCLab, DIEM, University of Salerno, Via Giovanni Paolo II 132, Fisciano (SA), 84084, Italy
[c] Department of Computer Science and Engineering, New Technologies for Information Society, University of West Bohemia, Technicka 18, Pilsen, 330 01, Czech Republic

ARTICLE INFO

Keywords:
Precision medicine
Federated learning
Explicit models
Grammatical evolution
Glucose forecasting

ABSTRACT

The problem of executing machine learning algorithms over data while complying with data privacy is highly relevant in many application areas, including medicine in general and Precision Medicine in particular.

In this paper, an innovative framework for Federated Learning is proposed that allows performing machine learning and effectively tackling the issue of data privacy while taking a step towards security during communication. Unlike the standard federated approaches where models should travel on the communication networks and would be subject to possible cyberattacks, the models proposed by our framework do not need to travel, thus moving in the direction of security improvement. Another very appealing feature is that it can be used with any machine learning algorithm provided that, during the learning phase, the model updating does not depend on the input data.

To show its effectiveness, the learning process is here accomplished by an Evolutionary Algorithm, namely Grammatical Evolution, thus also obtaining explicit knowledge that can be provided to the domain experts to justify the decisions made. As a test case, glucose values prediction for a number of patients with type 1 diabetes is considered and is tackled as a classification problem, the goal being to predict for any future value a possible range. Finally, a comparison of the performance of the proposed framework is performed against that of a non-Federated Learning approach.

1. Introduction

Precision Medicine (PM) is defined as "prevention and treatment strategies that take individual variability into account" [1,2] and is becoming more and more relevant to the medical community. In it, attention is given to the personalization of the treatment for each patient. PM is highly important in managing diseases, and this holds especially true in the predictive part. In fact, PM aims at the detection and treatment of perturbations in patients as early as possible, even prior to the appearance of symptoms. This early detection and treatment can allow the optimization of the treatment for every patient and the possibility to quickly act to circumvent the progression of the disease. As a consequence, PM is becoming more and more important in nowadays' medicine, and among its many benefits we may recall here at least improvement in disease detection, preemption of disease progression, customization of disease-prevention strategies, and prescription of more effective drugs.

Artificial Intelligence can prove useful in PM [3–5], and applications of Machine Learning (ML) methodologies are more and more often being reported successful in several studies pertaining to the use of PM in different medical areas [6–9]. Good and very recent reviews on the use of ML for PM are, among others, [10–12].

Although the potential benefits in the use of ML for medicine in general and for PM in particular are obvious, these application areas pose some difficult practical problems, such as data privacy [13], that must be carefully dealt with. In fact, medical data is highly sensitive and strictly personal to the patient, so it should never be revealed to anybody else, be it either another patient involved in the clinical trial or any subject responsible for data management or for the learning task. This problem becomes manifest whenever an ML algorithm should be executed over medical data from more subjects or groups of people that are either collected and made available in real time or situated in different repositories. In this case, an ML algorithm aiming at learning

* Corresponding author at: NCLab, DIEM, University of Salerno, Via Giovanni Paolo II 132, Fisciano (SA), 84084, Italy.
E-mail addresses: ivanoe.defalco@icar.cnr.it (I. De Falco), adellacioppa@unisa.it (A. Della Cioppa), txkoutny@kiv.zcu.cz (T. Koutny), umberto.scafuri@icar.cnr.it (U. Scafuri), ernesto.tarantino@icar.cnr.it (E. Tarantino).

https://doi.org/10.1016/j.bspc.2023.105416
Received 31 May 2023; Received in revised form 28 July 2023; Accepted 12 September 2023
Available online 22 September 2023

from the data of all the subjects should see all the data, so this latter should be gathered in a single-placed data set. Unfortunately, to do so, data should travel on a communication network, which can expose it to the leakage of sensitive personal information, thus impacting data privacy.

To tackle this problem, in 2015, a new approach to ML, called Federated Learning (FL) [14,15], was designed. In the basic form of FL, an ML program placed on a server starts by generating an initial global model and sends it to a collection of client programs, each of which refers to a subject or a group of subjects participating in the study. Learning from the data of any (group of) subject(s) only occurs on the specific client related to that particular subject/group, so local data need not be transmitted. The result of learning on a client is that the model is there modified locally to best abide by the data stored locally. At the end of the local learning, the client transmits the locally modified model to the server. Once this server has received all the modified models from the clients, it aggregates them; this means that a new global model is generated that takes into account all of the local learning. At the end of this aggregation step, the server transmits again this newly obtained global model to the clients, and the process iterates until a termination criterion is reached on the server. Interested readers may refer to [16] for a recent survey on advances in FL.

In general, a problem related to FL is that of security, meaning with this a variety of possible attacks on the client, server, and communication sides, ranging from model/data poisoning to adversarial attack [17,18]. Given that, at the origin of model/data poisoning and adversarial attacks, there are intrusions at the server/client level and that this problem is a general security issue, here we are interested in the specific FL security aspect related to the threats that could arise during the communication steps: in the classical FL, both local and global models need to travel and, consequently, are prone to cyberattacks while they are on the communication network. This aspect is particularly relevant from the security point of view, especially if we consider that the number of clients involved in FL can be large and the models' sharing requires sending a high number of parameters over the communication network.

Keeping this problem in mind, in this paper we introduce a new model to perform FL ensuring privacy while taking a step towards security, i.e., MFC-FL, which stands for Model-Free-Communication Federated Learning. In our proposal, neither subjects' data nor models, be they global or local, travel through the communication network. Rather, the only items that are exchanged between the server and the clients are the values computed during the evaluation of the local and the global quality of any solution; these latter are represented by real values. Therefore, the whole communication process can be considered secure in the sense that neither models nor data can be stolen while traveling through the network. This is possible in that the key feature of our approach is that both the server and the clients run a copy of the same algorithm with exactly the same parameter setting, each client on its own local data. As a consequence, our approach does not perform any explicit aggregation among local models. Rather, aggregation is an emergent feature in that, by combining the local performance into a global one, the learning process pushes the model towards a general behavior. Of course, this novelty does not cancel out the security issue because model/data poisoning and adversarial attacks can still take place, yet it can mitigate this threat by eliminating a very important possible source of lack of security linked to model transmission. This greatly reduces model poisoning attacks, while avoiding the models' theft and possible reconstruction of local data (model reconstruction attacks) and, consequently, ensuring data privacy. Finally, another appealing feature of the framework proposed is its generality given that it can work with any ML algorithm, provided that, during the training phase, the updating procedure the model undergoes does not depend on the input data.

The specific FL implementation we propose here is based on the application of an Evolutionary Algorithm (EA), a well-known ML methodology [19–21], and involves the employment of a client/server version of an EA [22,23]. Consequently, we call the specific approach presented here MFC-FL with EA (MFC-FLEA). As the EA, we avail ourselves here of Grammatical Evolution [24,25]: it is able to obtain explicit models for the problem to be tackled, hence making explicit knowledge available to users [26]. This latter can be provided to the domain experts to justify the decisions made, which is a very intriguing feature in general, and is of even higher interest in the medical domain. In fact, on the one hand, physicians wish to be confident on the motivations for a decision made by an artificial system and, on the other hand, patients are happy to receive, and often ask for, justifications of the diagnoses they receive. This framework represents the main novelty of our paper: as far as we know, just one paper [27] exists in which an EA was used to perform FL; in that case, nonetheless, the models found are local to each node, and travel through the network from the slaves to the master and vice versa, which makes the algorithm prone to cyberattacks and to possible loss of private data.

The proposed MFC-FLEA framework is tested on a problem related to diabetes, i.e., the prediction of the future glucose values starting from historical data. In particular, we transform the original multivariable regression problem into a classification one: the glucose range is divided into seven intervals, which gives origin to a seven-class classification task, where the goal is to predict the classes of the future glucose values. In this way, the aim of MFC-FLEA consists in the achievement of a global explicit model that can perform reasonably well on all the subjects participating in the trial, and, at the same time, can show good performance on new subjects not considered during the learning phase. For any patient, the use of this model allows the prediction of possible dangerous hypoglycemic or hyperglycemic events and their advanced treatment, which are both important goals for Precision Medicine, as mentioned earlier.

Finally, to assess the effectiveness of our approach, the widely-used and publicly-accessible Ohio T1DM data set [28] has been taken into account.

The rest of this paper is structured as follows: Section 2 describes the relevant state of the art, while Section 3 reports on the framework and the algorithm. Section 4 contains a description of the Ohio T1DM data set as well as of the preprocessing phase, and Section 5 accounts for the specific grammar used and for the transformation of the prediction problem into a classification task. Section 6 contains the experiments performed, the comparison between the framework proposed and another in which only local models are obtained, a statistical analysis, and a discussion. Finally, Section 7 reports the conclusions and the future work.

2. State of the art

The use of FL for medicine is becoming more and more widespread and the trend is strongly increasing over the years. Interested readers may refer to some good and recent review papers on the application of FL to medicine and healthcare [29–33]. A review focusing on security issues when using FL in medicine is [18].

The use of FL for precision or personalized medicine, instead, has been in these last years less common, although with an increasing trend as well, and has led to the publication of several papers on this topic in the last couple of years or so. In the following, we provide some examples of the most recent of such papers.

In [34] (2019), the description takes place of the five-year effort of the European Medical Informatics Framework aiming at developing an infrastructure in which federated access could be scaled up and the importance of its utilization in PM is assessed.

In [35] (2021), an extension to FL is proposed, in which personalized layers are used in neural networks. The weights for such layers are stored on client nodes so that learning patterns can be obtained from the data of the users.

In [36] (2021), the authors start from the concept of the Patient Similarity Network (PSN) recently introduced in PM to group patients

on the basis of their similarities, and introduce a federated model for it, termed as FPSN. This latter is applied to a COVID-19 data set and is tested against other models.

In [37] (2021) Swarm Learning is introduced, meaning with this a decentralized machine-learning approach that takes advantage of some of the most recent ideas, as edge computing, blockchain-based peer-to-peer networking and coordination. The authors claim that their approach maintains confidentiality while a central coordinator is not needed. The approach is tested on four data sets making reference to COVID-19, tuberculosis, leukaemia and lung pathologies, respectively.

In [38] (2022), the importance of feature selection on medical data sets is considered, and a federated approach is proposed of a privacy-preserving Cox proportional hazards regression model with LASSO regularization as feature selector. This model is evaluated, among others, on real world data related to the observations of lung cancer patients, for each of which 106 radiomic features were extracted.

In [39] (2022), a description is given of the ATHENA project funded by the Flemish Government. Its main objectives include accelerating the use of PM for bladder cancer and multiple myeloma, and applying distributed and privacy-preserving methods to the stratification of patients.

In [40] (2022), the use of federated Random Forests models is proposed, and their evaluation is effected with a particular focus on the heterogeneity within and between data sets.

In the review [41] (2023), the role of machine learning in several fields, among which PM, is thoroughly discussed.

In [42] (2023), it is shown that FL is useful to the improvement of the robustness of artificial intelligence algorithms, especially when big data are to be faced, as it is often the case for PM. This positively impacts the trust in these algorithms from users.

From these papers, it can be concluded that FL is seen as a promising tool for PM both at a medical level and from a strategic viewpoint.

Yet, in all these papers, models, be they local or global, must travel through the communication network. Some very recent notable examples are, e.g., [43–45]. This is, we feel, a weakness of those papers with reference to the issue of security: should a model be intercepted, it could either be predated by attackers, so that personal information could be stolen, or it could be modified, which could lead to alteration in the learning process and in the final model obtained. This is the first issue where our proposal takes a step forward: if no model travels, no model can be stolen or modified.

Our paper, instead, introduces in the literature the fact the only performance values of the models need to travel to perform FL. Should such a piece of information be stolen, it would be useless to attackers; only possible attack is of the *malicious* type, so the fitness value intercepted could be modified, which could slightly modify the evolution but not strongly alter the model.

An interesting intermediate case, in which both models and fitness values are transmitted, is reported in [46] (2021): in it, a client–server architecture is considered and ANNs are used for ML. The declared goal is the reduction in the communication costs. To achieve this goal, a Federated Particle Swarm Optimization (FedPSO) was designed to replace the classical FedAvg scheme: on each client, the ANN weights are locally updated thanks to a PSO algorithm. Then, each client sends the local fitness value to the server; this latter determines the client with the best fitness and requests to it the values of the weights. Then, these weights are sent to all the clients and the process iterates. The PSO algorithm is, actually, distributed among the clients: on each of them, just one individual is present, rather than a whole population.

As far as we know, so far no other paper in the literature has introduced a general framework able to work with any ML algorithm as we do here. Of course, it is well known that the classical FL approach can work well with a set of some ML algorithms as, e.g., regression, K-means, Radial Basis Functions, Artificial Neural Networks. Yet, the use of population-based ML algorithms is cumbersome in that framework. Our novelty here is that we have widened the set of ML algorithms that can be used: every ML algorithm, for which the updating procedure for the model during learning does not depend on the input data, is eligible to be used here.

Table 1
Comparison of the approach proposed here against the current FL literature.

	Models travel?	Fitnesses travel?	General framework?	Use of EA?
Current literature	yes	no	no	no
[46] (2021)	yes	yes	no	yes/no
[27] (2023)	yes	no	no	yes
This paper	no	yes	yes	yes

As concerns the use of EAs in FL, instead, up to now they have been in some cases used as an optimization tool, the learning duty being left to other ML algorithms as, e.g., multi-layer perceptrons [47,48], convolutional neural networks [47–51], dual adversarial generative networks [52], or radial basis functions [53]. This appears to be a limitation of the current FL because Swarm Intelligence Algorithms and EAs are well known in the scientific literature to be good optimizers, capable of obtaining noticeable results in several application field. To the best of our knowledge, only one paper exists [27] in which an EA, namely a Grammatical Evolution algorithm, is used by us to directly perform FL and achieve explicit knowledge. Yet, in that paper, models migrate through the connection network, which makes that approach unsafe. This being the state of the art, our approach is a breakthrough, because it easily allows Swarm Intelligence Algorithms and EAs to be utilized as the ML engine.

Table 1 summarizes the contents of this section, and shows the advantages of our approach with respect to the current literature on FL. In it, four aspects are taken into account, one per column, respectively: whether or not models travel, whether or not performance values travel, whether or not the framework is general (meaning with this if it is applicable to many ML algorithms), and whether or not an EA is used to perform ML. The first row in the table represents the state of the art, the second the FedPSO approach reported in [46], the third our recent paper [27] (2023) and the fourth the current paper. For [46], the cell "use of EA" contains "yes/no", because, at a general level, a population-based PSO scheme is used, yet, at the level of any client, just one solution is present on it.

3. Model-free-communication federated learning

Before discussing our method, we briefly sketch out the canonical FL approach as introduced by McMahan et al. in [14,15]. According to it, the aim is to identify a general model $\mathcal{M}$ by aggregating a set of local models $\mathcal{M}_i$ trained by a number m of different clients $\{C_1, \ldots, C_m\}$ on their local data $\{D_1, \ldots, D_m\}$, whose communication can be arranged either centrally by a server S or peer-to-peer. In the following we will refer to the centralized FL both for sake of conciseness and in that little changes apply to the peer-to-peer one.

Centralized FL consists of three steps:

1. **model initialization**: the server initializes a model $\mathcal{M}$ and, then, distributes it to all the clients participating in FL;
2. **local model training**: each client performs a learning phase on its own local data, thus obtaining its own updated version $\mathcal{M}_i$ of the model. Then, the updated model $\mathcal{M}_i$ of each client is sent to the server;
3. **aggregation**: once the server has received all the local updated models $\mathcal{M}_i$, aggregates them (typically this is accomplished by averaging the values of the models' parameters, e.g., in case of a neural network, the weights of the neurons) thus obtaining a new global model $\mathcal{M}$. Then, such a model is sent to all the clients and the process continues from the step 2 until a stopping criterion is met.

By doing so, the FL goal becomes to train separately a general model $\mathcal{M}$ on all the local data sets of the different clients to optimize the performance function $\psi(\mathcal{M})$:

$$\psi(\mathcal{M}) = \frac{1}{m} \cdot \sum_{i=1}^{m} \phi(\mathcal{M}_i) \tag{1}$$

where m is the number of clients and $\phi(\mathcal{M}_i)$ the local performance function value of the solution $\mathcal{M}_i$ on the ith client.

It should be noted here that, while it is possible to evaluate $\psi(\mathcal{M})$ each time at the end of the step 3, actually it is computed by the server only at the end of the whole learning process. In fact, before stopping, the clients send the last model update to the server along with its local performance, the server aggregates all the updates into the final global model, computes $\psi(\mathcal{M})$ and sends the final global model to the clients.

In the following sections, we first describe our proposed general MFC-FL framework, without any reference to the specific ML algorithm employed, and then specialize the description to the MFC-FLEA framework in which an EA is utilized.

3.1. The MFC-FL framework

In order to get a general FL framework in which neither clients' data nor models, be they global or local, travel through the communication, we need that:

- the aggregation step should not be performed by the server, but it should be done on the clients side. Obviously, this means that no explicit aggregation can be accomplished, but it should only take place in an implicit way;
- the local learning algorithm that performs the model updating should not depend on the local data. For example, Artificial Neural Networks, be they shallow or deep, when endowed with backpropagation, cannot work. Anyway, replacing backpropagation with, for example, Particle Swarm Optimization [54,55], makes them perfectly suitable for our approach.

In particular:

- the server and all the clients should generate exactly and autonomously the same initial solution $\mathcal{M}$ to a specific ML problem;
- the updating procedure the model undergoes on each client should perform the same action at the same time and should produce exactly the same updated model;
- the implicit mechanism the updating makes use of should be related to the model's local performance on the data of all the clients. For example, the global performance value could be computed by averaging the local performance values, thus pushing the learning process to induce the emergence of an effective global model. As a consequence, the performance of the global model should be evaluated at each iteration of the FL process.

The general MFC-FL framework, detailed in Fig. 1, is an architecture composed by a server and a set of clients and consists of three steps:

1. **model initialization**: the server and all the clients generate exactly the same initial model $\mathcal{M}$ for the specific ML problem;
2. **performance communication**: any client i measures the local performance $\phi^i(\mathcal{M})$ and sends it to the server; this latter, once received the local performance values from all the clients, computes the global performance, i.e., the global quality of this model on the entire data set:

   $$\langle\psi\rangle \equiv \psi(\mathcal{M}) = \frac{1}{m} \cdot \sum_{i=1}^{m} \phi^i(\mathcal{M}) \tag{2}$$

   and sends it to all the clients;

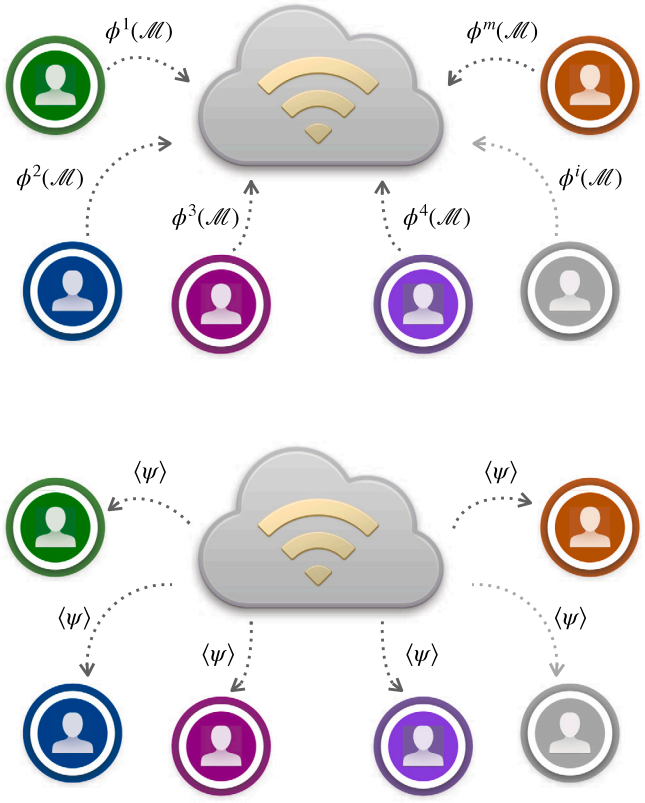


Fig. 1. Model-Free-Communication Federated Learning framework: the performance communication step. The top pane of the figure sketches the dispatch from the clients to the server of the solution's local performance ($\phi^i(\mathcal{M})$ represents the performance value of the current solution $\mathcal{M}$ computed on the local data present on the ith client), while the bottom one traces the dispatch from the server to the clients of the solution's global performance.

3. **local model training**: each client performs a learning phase based on the global performance just received from the server, thus obtaining an updated version of the model $\mathcal{M}$, which implicitly exploits the information on the performance on the other data sets. This updated solution will be the same on all the nodes, be they the server or a client.

Steps 2 and 3 are repeated until a stopping criterion is met.

As stated above, in our approach, neither subject's data nor models are sent through the communication network, the only items being sent are the values of the local and the global performance measure, as evidenced in Fig. 1. Therefore, the whole communication process is secure in the sense that relevant information such as the models and personal data cannot be stolen on the network.

3.2. The MFC-FLEA framework

In this section, we detail MFC-FL when an EA is adopted as a local machine-learning procedure (see Fig. 2). In such a case, as before, we have a server S and a set of m clients $\{C_1, \ldots, C_m\}$, each of which has its own local data. All of them generate the same initial set (population) of n possible models (individuals) $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ to face the specific ML problem. In this situation, the same general mechanism takes place as in MFC-FL, the only difference being that, on each client, we have a population of n models whose quality is evaluated on the local data thanks to a fitness function (performance measure) ϕ based on some typical metric for the ML problem. The steps MFC-FLEA undergoes are the following:

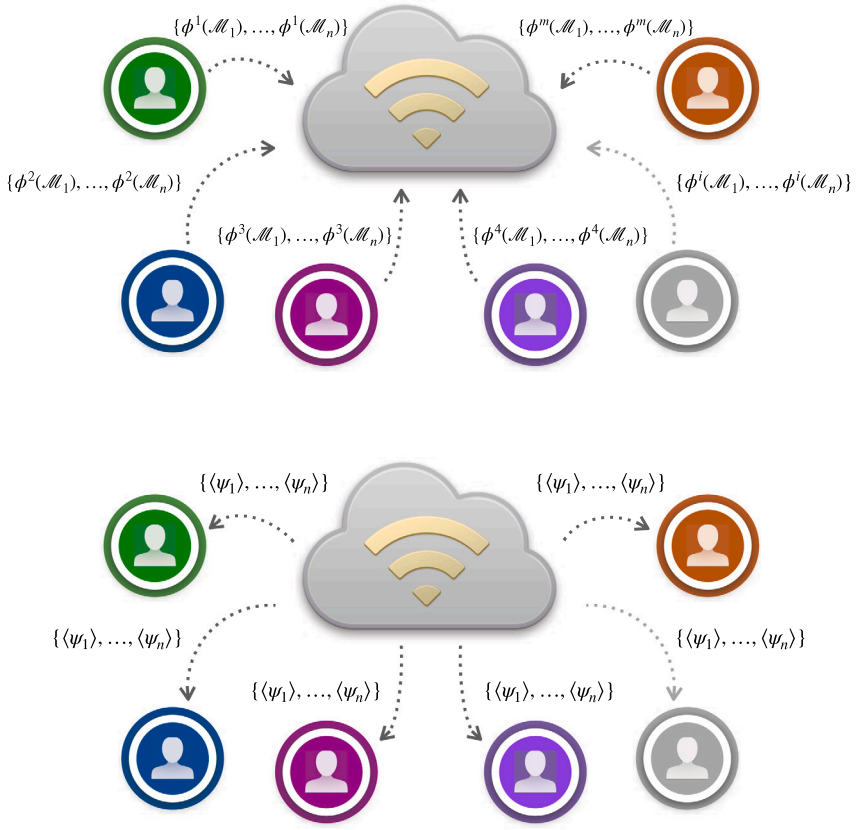


Fig. 2. MFC-FLEA framework: the performance communication step. The top pane of the figure sketches the dispatch from the clients to the server of the models' local performance ($\phi^i(\mathcal{M}_j)$ represents the fitness value of the solution $\mathcal{M}_j$ computed on the local data present on the ith client), while the bottom one traces the dispatch from the server to the clients of the models' global performance.

- **model initialization**: the server and all the clients generate exactly the same initial population of models $\boldsymbol{\mathcal{M}}$ for the specific ML problem;
- **performance communication**: each client evaluates each model in the population on its local data and sends the set of computed performance (fitness) of each individual to the server, i.e., $\boldsymbol{\Phi}^i = \{\phi^i(\mathcal{M}_1),\ldots,\phi^i(\mathcal{M}_n)\}$, where $\phi^i(\mathcal{M}_j)$ represents the fitness value of the generic jth solution $\mathcal{M}_j$ computed on the local data present on the client i; once the server has received the set $\boldsymbol{\Phi}^i$, $\forall i \in \{1,\ldots,m\}$, computes the average of the local performance for each individual as follows:

$$\langle\psi_j\rangle \equiv \psi(\mathcal{M}_j) = \frac{1}{m}\cdot\sum_{i=1}^{m}\phi^i(\mathcal{M}_j), \quad j\in\{1,\ldots,n\} \tag{3}$$

  which represents the global fitness value of that solution on the whole distributed data set. Finally, the server sends the set $\langle\Psi\rangle = \{\langle\psi_1\rangle,\ldots,\langle\psi_n\rangle\}$ to all the clients.
- **local model training**: each client replaces the local values with the global values just received. Then, starting from the currently available solutions and from their global fitness values, on each node, the typical actions of the chosen EA are taken and a new set of possible solutions is generated.

The algorithm iterates effecting these steps until a termination condition is met. A typical condition is fulfilling a given maximum number of generations g_{max}, which is also used in this paper. Finally, at the end of the whole training, the model with the best global performance is taken as the final global model.

It should be remarked here that, similarly to the general MFC-FL framework, there is no communication of data and models; rather, just the fitness values, typically real numbers, travel and, consequently, the approach is secure as no model or personal data can be stolen on the network.

Algorithm 1 Pseudocode of MFC-FLEA on the server

```
set the maximum number of generations g_max
set the current generation g equal to 0
generate the initial population with n models (the same as in all the clients)
while g < g_max do
  for each slave do
    receive the local fitness values
  end for
  compute the global fitness values
  for each slave do
    send the global fitness values
  end for
  generate the new population according to the chosen EA
  g = g + 1
end while
```

As the specific EA, in this paper we exploit Grammatical Evolution (GE) [24,25]: this is because in GE each proposed solution is an explicit model consisting in an expression containing (some of) the problem variables. This is a very positive feature because it allows us to extract explicit knowledge that can be provided to physicians so that they can understand the reasons for the algorithm to take its decisions, and can also check if they are sensible from a medical viewpoint. This is highly desirable in general and in the specific medical domain.

Algorithm 1 shows the pseudocode for the server, while Algorithm 2 does the same for a generic client.

Algorithm 2 Pseudocode of MFC-FLEA on a client

set the maximum number of generations $g_{\max}$
set the current generation g equal to 0
generate the initial population with n models (the same in all the clients and in the server)
while $g < g_{\max}$ **do**
  evaluate the local fitness of each model
  send the local fitness values to the server
  receive from the server the global fitness values
  replace the local fitness values with the received global ones
  generate the new population according to the chosen EA
  $g = g + 1$
end while

4. The data set

In this paper, the proposed MFC-FLEA framework is tested on a problem related to diabetes, i.e., the prediction of the future glucose values for participating patients. To this end, the widely-used and publicly-accessible Ohio Type 1 Diabetes Mellitus (T1DM) data set [28] is taken into account. Its gathering was carried out at the Ohio University; the contents come from 12 patients, each of which was monitored through the use of a continuous glucose monitoring (CGM) system for a total span of around two months while they were on insulin pump therapy.

For each such subject, this data set contains a large number of parameters. Some of these parameters were gathered through different sensors, so they were taken at regular intervals, for example glucose sampling frequency is equal to $\Delta t = 5$ minutes; others, instead, were saved through self-declared recordings, leading to irregular intervals. Since the Ohio T1DM data set became publicly available, lots of papers were published in which the authors have tried to find a good parameter subset allowing the best possible performance [56–63], in some cases only considering the glucose data themselves [64–71]. More in general, the recent review paper [72] evidences in its Table 3 that, when data-based algorithms and models are considered for blood glucose and hypoglycemia prediction, the vast majority of the papers takes into consideration as the parameters measured subcutaneous glucose, injected insulin (basal plus boluses), and carbohydrates ingested during the day (time and estimated size of all meals) with few papers also taking exercise into account. Similarly to these papers, here too, the decision has been made to only consider these three parameters. Going back to the Ohio data set, this is the same choice made in, e.g., [57,58,62,70].

Starting from the present and recent measurements for them, MFC-FLEA should predict future values for glucose.

We have decided to associate to each considered patient a client node, where the private data of that subject is stored. To perform learning in a supervised mode, a training set and a testing set are obtained for each subject by suitably partitioning the data series yielded by their monitoring. The former set is utilized for the learning phase, at the end of which a model is extracted, while the latter is utilized for assessing the quality of that model over unseen data.

This supervised learning step takes place on the data of six subjects out of the twelve contained in the Ohio T1DM data set, namely on those that were collected in 2018 [73]. A subsequent phase of validation of the achieved global model is carried out over the testing sets of the other six subjects who were monitored in 2020.

The division of each subject's data into training and testing sets has been effected in the same way as reported in [28], where interested readers can find information on the number of items assigned to either set.

4.1. Data preprocessing

To effect the preprocessing of the data, several decisions have been made:

- as concerns glucose data, samples in which glucose readings are not present are discarded; this takes place in both training and testing sets and has the goal to circumvent the possibility that values artificially introduced to replace them could negatively influence the prediction model to be obtained;
- for data related to insulin and carbohydrates, these are aligned to the nearest time at which a glucose reading takes place through the use of the CGM;
- for outliers, no detection is effected;
- for all data, no normalization is carried out, rather the actual values are used.

Aiming at estimating the influence on the glucose levels of the discrete signals representing administered insulin, meaning with this both insulin boluses and basal insulin, these latter should be converted into continuous signals. The same should take place for the assumed carbohydrates.

A detailed description of how the above steps are taken is too lengthy and would go beyond the scope of the current paper. Interested readers may refer to [27].

Suffice it to say here that, at the end of this preprocessing, two new discrete signals are obtained, each showing an item every Δt minutes. These two signals are denoted with $I(t)$ and $C(t)$ and represent, respectively, the values of the absorbed insulin and of the carbohydrates at the corresponding time steps.

5. MFC-FLEA to forecast future glycemic trends

Predicting future glycemic trends for T1DM patients may be considered a regression problem involving multivariate time series data [74–79]. This falls under the domain of data-driven models that leverage information gathered by CGM systems. To address this problem using the MFC-FLEA framework, we utilize GE's capabilities of automatically evolving explicit regression models.

Unlike other EAs, GE explicitly incorporates context-free grammars that are useful to tailor desired forms for the obtainable solutions representing models.

In order to achieve this goal, an appropriate grammar and a fitness function must be defined. Additionally, instead of aiming to the exact prediction of future glucose values, we change the time series regression task into a classification one.

5.1. The grammar

Fig. 3 presents the context-free grammar that outlines the syntax of the expressions obtainable using GE. Here, $\langle gluc \rangle$ indicates past glucose levels, $\langle ins \rangle$ and $\langle cho \rangle$ represent future insulin and carbohydrate absorption, and $\langle dg \rangle$ represents the difference between the actual value of the glucose and one of its past values.

The first line defines the general structure of any model: by using constants and suitable signs deriving from physiology, it sums an expression on the glucose values, an expression on the insulin values, and an expression on the carbohydrates values. This resulting expression is modified by means of a further expression on the derivatives of the glucose values, the two being tied by one arithmetical operator (+ or - or *).

Table 2 provides an overview of the protected functions used and of their meaning.

By taking into account the $G(t)$ values at intervals of Δt minutes within a time window of $k\Delta t$ minutes prior to the current time t, and those of $I(t)$ and $C(t)$ every Δt minutes within a time window of $h\Delta t$ minutes after the current time t, the aim is to obtain an explicit

⟨*glucose*⟩ ::= ((⟨*e_gluc*⟩) + ⟨*d*⟩.⟨*d*⟩ * abs(⟨*e_cho*⟩) - ⟨*d*⟩.⟨*d*⟩ * abs(⟨*e_ins*⟩)) ⟨*op*⟩ (⟨*e_dg*⟩)

⟨*e_gluc*⟩ ::= (⟨*e_gluc*⟩⟨*op*⟩⟨*e_gluc*⟩) | aq(⟨*e_gluc*⟩,⟨*e_gluc*⟩) | ⟨*func*⟩(⟨*e_gluc*⟩) | ⟨*gluc*⟩ | ⟨*number*⟩

⟨*e_ins*⟩ ::= (⟨*e_ins*⟩⟨*op*⟩⟨*e_ins*⟩) | aq(⟨*e_ins*⟩,⟨*e_ins*⟩) | ⟨*func*⟩(⟨*e_ins*⟩) | ⟨*ins*⟩ | ⟨*number*⟩

⟨*e_cho*⟩ ::= (⟨*e_cho*⟩⟨*op*⟩⟨*e_cho*⟩) | aq(⟨*e_cho*⟩,⟨*e_cho*⟩) | ⟨*func*⟩(⟨*e_cho*⟩) | ⟨*cho*⟩ | ⟨*number*⟩

⟨*e_dg*⟩ ::= (⟨*e_dg*⟩⟨*op*⟩⟨*e_dg*⟩) | aq(⟨*e_dg*⟩,⟨*e_dg*⟩) | ⟨*func*⟩(⟨*e_dg*⟩) | ⟨*dg*⟩ | ⟨*number*⟩

⟨*op*⟩ ::= + | - | *

⟨*func*⟩ ::= plog | psqrt | exp

⟨*gluc*⟩ ::= G(t) | G(t-Δt) | ... | G(t-$k\Delta t$)

⟨*ins*⟩ ::= I(t) | I(t+Δt) | ... | I(t+$h\Delta t$)

⟨*cho*⟩ ::= C(t) | C(t+Δt) | ... | C(t+$h\Delta t$)

⟨*dg*⟩ ::= G(t) - G(t-Δt) | ... | G(t) - G(t-$k\Delta t$)

⟨*number*⟩ ::= ⟨*d*⟩.⟨*d*⟩ | -⟨*d*⟩.⟨*d*⟩

⟨*d*⟩ ::= [0, ..., 99]

Fig. 3. The grammar for the glucose forecasting model (Eq. (4)).

Table 2
The protected functions contained in the chosen grammar and their meaning.

Protected function	Meaning
plog(x)	$\log(1+\lvert x\rvert)$
psqrt(x)	$\sqrt{\lvert x\rvert}$
aq(x, y)	$\frac{x}{\sqrt{1+y^2}}$

Table 3
The seven classes utilized to effect the glucose classification task.

Class	Class ID	Range (mmol/L)	Range (mg/dL)	Action required
very low	0	<3.0	<54	hypo: immediate action
low	1	[3.0–3.9[	[54–70[	hypo: alert and monitor
normal-to-low	2	[3.9–5.0[	[70–90[	'towards hypo' warning
normal	3	[5.0–7.8[	[90–140[	none
normal-to-high	4	[7.8–10.0[	[140–180[	'towards hyper' warning
high	5	[10.0–13.9[	[180–250[	hyper: alert and monitor
very high	6	≥13.9	≥250	hyper: immediate action

regression model assessing $\widehat{G}(t+h\Delta t)$, i.e., the predicted glucose value at time $t+h\Delta t$:

$$\begin{aligned}\widehat{G}(t+h\Delta t) = \Big(& \Gamma\big(G(t), G(t-\Delta t), \ldots, G(t-k\Delta t)\big) \\ & - \Theta\big(I(t), I(t+\Delta t), \ldots, I(t+h\Delta t)\big) \\ & + \Omega\big(C(t), C(t+\Delta t), \ldots, C(t+h\Delta t)\big) \Big)\end{aligned}$$

$$\diamondsuit\, \Phi\big(dG(t, t-\Delta t), \ldots, dG(t, t-k\Delta t)\big) \tag{4}$$

here, the symbol $\diamond$ stands for algebraic operations within the set: $\{+,-,\cdot\}$, and Γ, Θ, Ω, and Φ are expressions involving, respectively, G, I, C, and dG.

5.2. From regression to classification

Traditionally, glucose prediction is effected as a multiseries regression to accurately forecast glucose values. However, in recent literature, there is a growing trend towards using classification for the prediction of glucose ranges instead of exact values [64,66,80,81]. This approach proves valuable when predicting with sufficient lead time high-risk situations; in diabetes, such situations make reference to hyperglycemic or hypoglycemic events. In these cases, predicting the occurrence of such events is more critical than precisely forecasting glucose values.

To enable classification, the continuous glucose values are categorized into seven intervals, resulting in a seven-class classification task. Specifically, two classes represent hypoglycemia, three classes represent euglycemia (normal values), and two classes represent hyperglycemia.

The inclusion of two hypo- and two hyper-classes aligns with the 2017 international consensus outlined in [82]. We utilize the same boundaries specified in that document, which are presented in Table 3.

Regarding the euglycemic range, contrary to [82], we opt for dividing it into three classes as described in Table 3. This decision is motivated by the need to track potentially dangerous glucose developments that deviate from the target range. With a single normal/target range, there would be no indication of glucose deviating towards hypoglycemic or hyperglycemic levels before the occurrence of an event.

Table 4
The amount of items in the training (Tr) and testing (Ts) sets for each subject ID and for each of the seven classes ($C0 \div C6$).

ID	C0	C1	C2	C3	C4	C5	C6
	Tr/Ts	Tr/Ts	Tr/Ts	Tr/Ts	Tr/Ts	Tr/Ts	Tr/Ts
559	60/31	342/42	969/172	2889/549	1763/559	2718/683	1281/263
563	33/0	252/15	1015/88	4646/648	2968/758	2536/869	278/120
570	14/0	202/11	518/90	1908/355	2119/325	4096/902	1747/882
575	217/35	734/104	1110/227	4010/884	2152/432	1877/555	461/159
588	25/0	107/4	474/55	3676/605	3648/800	3915/1123	607/150
591	113/14	297/117	851/252	3386/969	2416/561	2592/695	717/62

With the use of three classes, the middle of which is wider while the two bordering towards hypo- and hyper-areas are narrower, we can provide forewarnings prior to hypoglycemic or hyperglycemic events. For hypoglycemia, this approach entails a sequence of normal values after which a sequence follows containing values transitioning from normal to hypo.

Table 3 displays for each class the ID assigned to that class, the corresponding glucose value range in mmol/L and mg/dL, and the actions needed for that class while monitoring.

This transforms the original task into a classification one, where the objective is the prediction of the class of any future glucose value based on available data for glucose, insulin, and carbohydrates.

Fig. 4 illustrates, for the training set of each patient considered, the conversion of the original continuous glucose signals into a set of instances for the classification into seven classes.

Table 4 provides the amount of items in the training (Tr) and testing (Ts) sets for each subject and class. In general, for each patient, the Ohio T1DM data set contains the data resulting from eight weeks of monitoring, the last ten days of which were kept for the testing phase. Yet, the number of significant glucose items is different for the different patients; this is because every now and then a patient may have decided to momentarily stop monitoring, or the device may have stopped functioning. All of this considered, for each patient, the training set contains about the former 80% of the significant data and the testing set about the latter 20%.

The numbers reported in the table evidence that, for all the patients investigated, the three classes referring to normal values and the two classes containing hyperglycemic values have much higher numbers of items than the two classes for 'very low' and 'low' glucose values, that often consist in only a few values. Particularly, it is interesting to note that, for the test sets, the 'very low' class is empty for the patients 563, 570, and 588, which indicates a highly unbalanced distribution in these data sets; in general, this poses challenges for classification [83].

5.3. Fitness function

To assess the quality of the found solutions, an appropriate fitness function should incorporate metrics specific to the classification task, such as accuracy, sensitivity, specificity, precision, recall, area under the ROC curve, F_1 score, Matthews correlation coefficient, etc.

In this study, we chose to use the F1 score [84], given the highly unbalanced nature of the datasets for the six subjects investigated. Particularly, classes 0 (very low glucose values), 1 (low glucose values), and, for some subjects, 6 (very high glucose values) consist of very few instances compared to the remaining four classes.

In general, a metric like accuracy is unsuitable for unbalanced datasets [85–87]: achieving good performance in the majority class(es) could imply achieving numerically good results without effective learning occurring in the minority class(es). Such a situation means that, when classifying an item of one of the minority classes, the algorithm may incorrectly assign it to one of the majority classes.

Other metrics such as, among others, specificity, sensitivity (or recall), and precision, are better suited than accuracy when data is imbalanced; in fact, they take into account the different types of errors (false positives or false negatives) made by a given model. Often, more such metrics should be considered at the same time. Unfortunately, some of these measures exhibit a trade-off; moreover, it is impractical to simultaneously monitor several measures [85]. As a consequence, in the case of unbalanced data sets, new metrics like F_1 score [85,86], Matthews correlation coefficient [88], or Kohen's Kappa [89] have been developed that can effectively address this issue. In particular, the F_1 score combines precision and recall into a single metric; more precisely, it is their harmonic mean.

Given a classification problem over two classes where one is considered as the positive and the other as the negative, the F_1 score is computed as follows:

$$F_1 = \frac{2 \cdot tp}{2 \cdot tp + fp + fn} \tag{5}$$

where:

- *tp*: amount of true positives (items in the positive class exactly allotted to that class);
- *fp*: amount of false positives (items in the negative class wrongly assigned to the positive class);
- *fn*: amount of false negatives (items in the positive class wrongly assigned to the negative class).

For multiple classes, like in the problem faced here, the definition of the F_1 score can be generalized in several ways. We have decided to utilize here the weighted averaging, where the resulting F_1 score value takes into account the contributions of the F_1 scores computed for each class, weighted using the number of instances belonging to the class. The formula is as follows:

$$F_{1_w} = \frac{\sum_{n=1}^{c} p_i \cdot F_{1_i}}{c} \tag{6}$$

here, c represents the number of classes present in the data set, p_i symbolizes the percentage of instances in the ith class, and F_{1_i} stands for the F_1 score calculated for the ith class. In this case, we use the subscript w to represent the fact that this definition should only be used when more than two classes are present in the data set.

Given that, for the data set at hand, seven classes are considered, in the remainder of this paper we will denote by F_1 the value defined in Eq. (6), i.e., $F_1 = F_{1_w}$.

The F_1 score ranges in $[0.0 - 1.0]$, with higher values indicating higher-quality classifications. By selecting this metric, the classification task is seen as one of maximization.

6. Experimental results

6.1. Experimental framework setting

We have implemented our framework starting from PonyGE2, a freely available version of a GE implemented in Python [90]. To run PonyGE2, the values for a set of parameters must be chosen; [90] reports the parameters and their meaning. At the end of a preliminary parameter tuning phase, the following values have been chosen: population size equal to 100, g_{max} equal to 1000, codon size equal to 100,000, tournament selection with size 4, mutation events equal

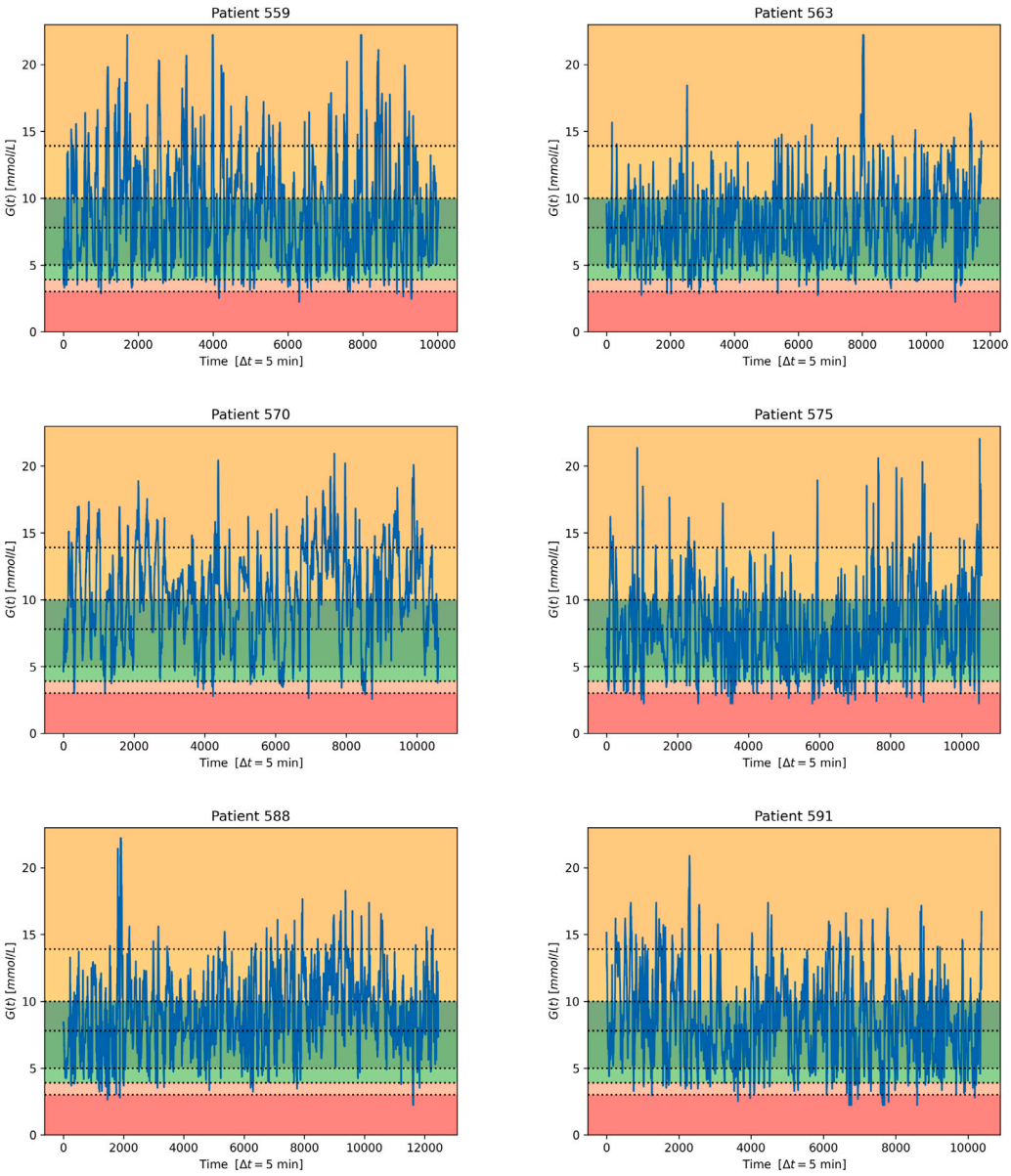


Fig. 4. Classification bands on the training set for each patient in the Ohio T1DM data set [73].

to 1, one-point crossover probability equal to 75%, int flip per codon mutation with one mutation event, and Position Independent Grow method for the individual initialization. For all the nodes, carrying out a number of generations equal to the maximum has been set as the termination condition.

In order to reduce the randomness deriving from the initialization of a GE population, throughout this section, for each experiment, 20 runs have been effected with 20 different random seeds.

The forecasting horizon $h\Delta(t)$ has been set equal to 30 min; in fact, it is known from the literature [78,91] that the prediction quality decreases as long as the forecasting horizon increases. A value larger than 30 min only makes sense when almost steady-state situations are considered, as it is the case, for example, when the subject is sleeping at night. This is because, in general, the occurrence of external events may give rise to unforeseeable and even large variations in glucose levels.

The width of the past time window $k\Delta(t)$, instead, has been set equal to 60 min. This value has been chosen because in the literature [76] it is shown that even a value as low as 30 is sufficient to achieve good prediction.

As in Ohio T1DM data set the interval between successive measurements is equal to 5 min, this yields that $k = 12$ and $h = 6$, respectively. Hence, at any given time step t, 12 glucose values before t and six insulin and carbohydrate values after t are to be considered both in the grammar and in Eq. (4).

To investigate the effectiveness of MFC-FLEA both on its own and with reference to absence of FL, two tests have been carried out:

1. MFC-FLEA has been executed: in each run, the global model is found. The best such model over the runs is considered;
2. a non-FL approach has been executed: in each of its runs, we perform separate optimizations for the six subjects, which yields a personalized local model for each of them. At the end of the runs, we collect all these local best models and pick from among them the model showing the highest average performance over the six subjects.

6.2. Findings and discussion

Table 5 shows the results obtained by the MFC-FLEA. Namely, it reports, in the Average column, the score of the global model with the

Table 5
Results of the MFC-FLEA algorithm in terms of F_1 score. For each patient, the related column reports the local quality of the best global model over the 20 runs. The Average column shows the average quality of the best global model; the std. dev. column displays the standard deviation of the performance over the six subjects.

ID	Patient						Average	std. dev.
	559	563	570	575	588	591		
F_1	0.7558	0.7241	0.8180	0.6964	0.7320	0.6829	0.7349	0.0483

Table 6
Results of the non-FL approach. The best average F_1 score is reported in bold.

Best	Patient						Average	std.
model	559	563	570	575	588	591		dev.
559	0.7687	0.5185	0.6763	0.7108	0.7301	0.6795	0.6807	0.0865
563	0.7441	0.7343	0.8199	0.6875	0.7331	0.6863	**0.7342**	0.0488
570	0.7511	0.7166	0.8155	0.7021	0.7291	0.6784	0.7321	0.0476
575	0.7424	0.5941	0.7585	0.7237	0.7239	0.6700	0.7021	0.0607
588	0.7468	0.5492	0.7769	0.6943	0.7559	0.6794	0.7004	0.0830
591	0.7495	0.4453	0.6341	0.6954	0.7314	0.6882	0.6573	0.1112

highest F_1 value over the 20 runs. Moreover, it also reports the value of that global model over each of the six subjects, and the standard deviation of the performance over the six subjects.

To express the best global model found by MFC-FLEA, we recall here the shape of the Eq. (4), where the several components are, respectively:

$$\begin{cases} \Gamma(G) = G(t) \\ \Theta(I) = 0.04e^{0.25(I(t+20))} \\ \Omega(C) = 0.02e^{0.25(C(t+20))} \\ \Phi(dG) = e^{0.25\left(\frac{dG(t-5)}{(dG(t-5)^{2.0}+1.0)^{0.5}}\right)} \end{cases} \tag{7}$$

The variables involved are: for the glucose, the current value $G(t)$; for the insulin, the value estimated in the future at time $t+20$; for the carbohydrates, the value estimated in the future at time $t+20$; for the derivative of the glucose, the past value at time $t-5$.

This general model contains parameters the values of which vary from subject to subject, depending on the dynamics specific to each patient. Otherwise said, the model is the same, but the dynamics, and the related data, are different from subject to subject. Hence, for each of them, this model will be able to predict the occurrence of dangerous and even life-threatening hypo- or hyper-glycemic events in the near future, and generate in advance the necessary warnings or requests for immediate action. Hence, it fits well the Precision Medicine framework.

Table 6, instead, is a bit more complicated because it deals with the local models. Namely, in this table, for the generic ith row, the cell (i,i) contains the F_1 score value obtained by the best local model, i.e., that with the highest F_1 score, found on the ith node when applied to the local data of the ith subject. Instead, each other cell (i,j) $(i \neq j)$ in the ith row reports the value obtained by that model when applied to the data of the jth subject. Moreover, the last two columns report, respectively, the averages and the standard deviations of the F_1 score values computed for the same local model over the different subjects.

Among the six local models, the one that obtains the higher average performance over the six subjects is that found over the node associated to patient 563; hence, from the point of view of FL, it is the preferable model, and we consider it as the best local model. With reference to Eq. (4), the explicit form of this model is made up of the following constituents:

$$\begin{cases} \Gamma(G) = G(t) \\ \Theta(I) = 0.12\,|I(t+20)| \\ \Omega(C) = 0.96\log\Big(\log\Big(\log\big(|C(t+20)+0.1|^{0.25}+1.0\big)+1.0\Big)^{0.5} \\ \qquad +\ 1.0)+1.0) \\ \Phi(dG) = 1.86\cdot dG(t-5) \end{cases} \tag{8}$$

Table 7
The results achieved by the two best models on the testing sets of the new patients.

Model	Patient						Avg	std.dev.
	540	544	552	567	584	596		
MFC-FLEA	0.6771	0.7591	0.7097	0.6857	0.7145	0.7064	0.7088	0.0287
non-FL	0.6760	0.7358	0.6989	0.6704	0.7256	0.6883	0.6992	0.0265

In this case, the variables involved are: for the glucose, the current value $G(t)$; for the insulin, the value estimated in the future at time $t+20$; for the carbohydrates, the value estimated in the future at time $t+20$; for the derivative of the glucose, the past value at time $t-5$.

A comparison between the models in terms of simplicity shows that the best local model is slightly less compact, especially in the part related to carbohydrates. As concerns the variables involved, instead, they are the same in the two models, both in number and in the variables themselves: $G(t)$, $I(t+20)$, $C(t+20)$, and $dG(t-5)$.

When, instead, the comparison between the two best models is effected from the performance point of view, the results in Tables 5 and 6 show that the global model found by MFC-FLEA is slightly better than the best local one provided by non-FL in terms of higher F_1 score over the whole data set composed by the six patients.

For comparison purposes, Fig. 5 reports the confusion matrices obtained for the patients 559 (top), 563 (middle), and 570 (bottom): on the left, the results for the best global model evolved by MFC-FLEA algorithm are shown, whereas, on the right, those for the best local model, i.e., model 563, are displayed. Similar information for the patients 575, 588, and 591 is presented in Fig. 6.

A very interesting question is related to the investigation of the generalization capability provided by the two best models found by the two different approaches when new patients are considered whose data has not been used until this moment. To answer this question, the other six subjects contained in the Ohio T1DM data set are taken into account. Table 7 shows the F_1 score values achieved by the two best models on the testing sets of these subjects. The table also shows, in its last column, the average and the standard deviation of these F_1 values computed over the six subjects.

From the table, it can be appreciated that, also over these six subjects never seen before, the best global model found by MFC-FLEA achieves, on average, better performance than that provided by the non-FL approach in terms of higher F_1 score. Namely, the MFC-FLEA-derived model performs better over the five patients 540, 544, 552, 567, and 696, whereas the non-FL one has better performance on subject 584 only. Therefore, our results suggest that the MFC-FLEA approach can generalize better than the non-FL one, which is relevant when new, not examined before, patients should be monitored.

Similarly to what done for Figs. 5 and 6, here too, for comparison purposes, Fig. 7 reports the confusion matrices obtained for the new patients 540 (top), 552 (middle), and 584 (bottom): on the left, the results for the best global model evolved by MFC-FLEA algorithm are shown, whereas, on the right, those for the best local model, i.e., model 563, are displayed. Similar information for the patients 567, 584, and 596 is presented in Fig. 8.

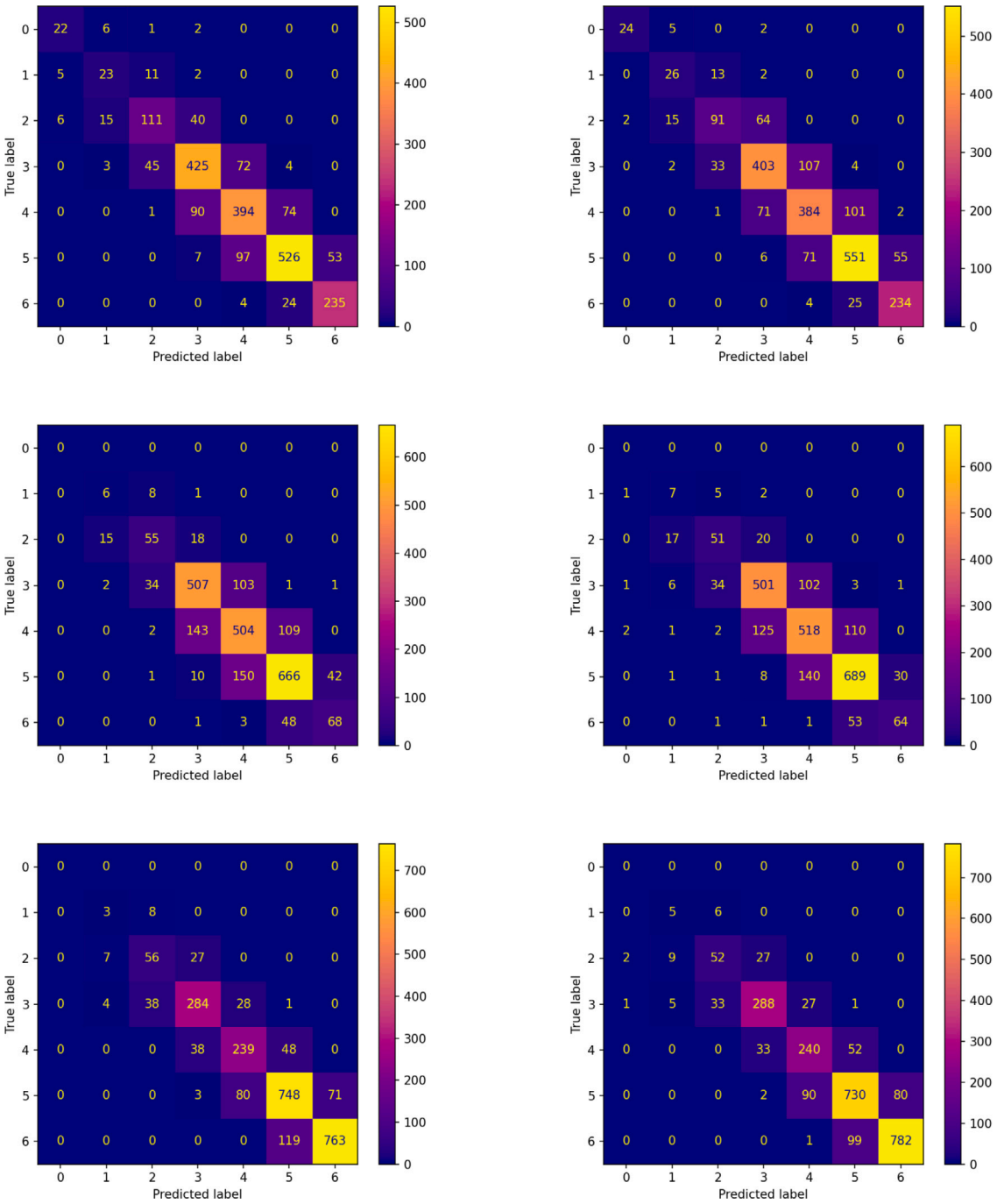


Fig. 5. Confusion matrices on the testing set for patients 559 (top), 563 (middle), and 570 (bottom): on the left, the results for the best global model evolved by MFC-FLEA algorithm, on the right, those for the best local model (model 563).

6.3. Statistical analysis

We have run a statistical analysis test to investigate if the best global model found by MFC-FLEA is statistically better than the best local model found by non-FL. High interest lies in checking this issue for the generalization ability of the two algorithms over unseen subjects, therefore we have considered the six patients that have not been used during the phase of the creation of the models, i.e., those accounted for in Table 7. To perform this analysis, we have availed ourselves of the online web platform 'Statistical Tests for Algorithms Comparison' [92] (STAC).[1] An important decision is that of choosing the most suitable statistical test. Our choice has fallen on the Quade test; in fact, this test does not consider all the problems as equal, rather it is able to account for the higher difficulty of some problems and for the larger differences in the results that the algorithms may show over them. This feature makes Quade test preferable to Friedman and Aligned Friedman tests, that can only consider all the problems as equally important. Readers

[1] https://tec.citius.usc.es/stac/

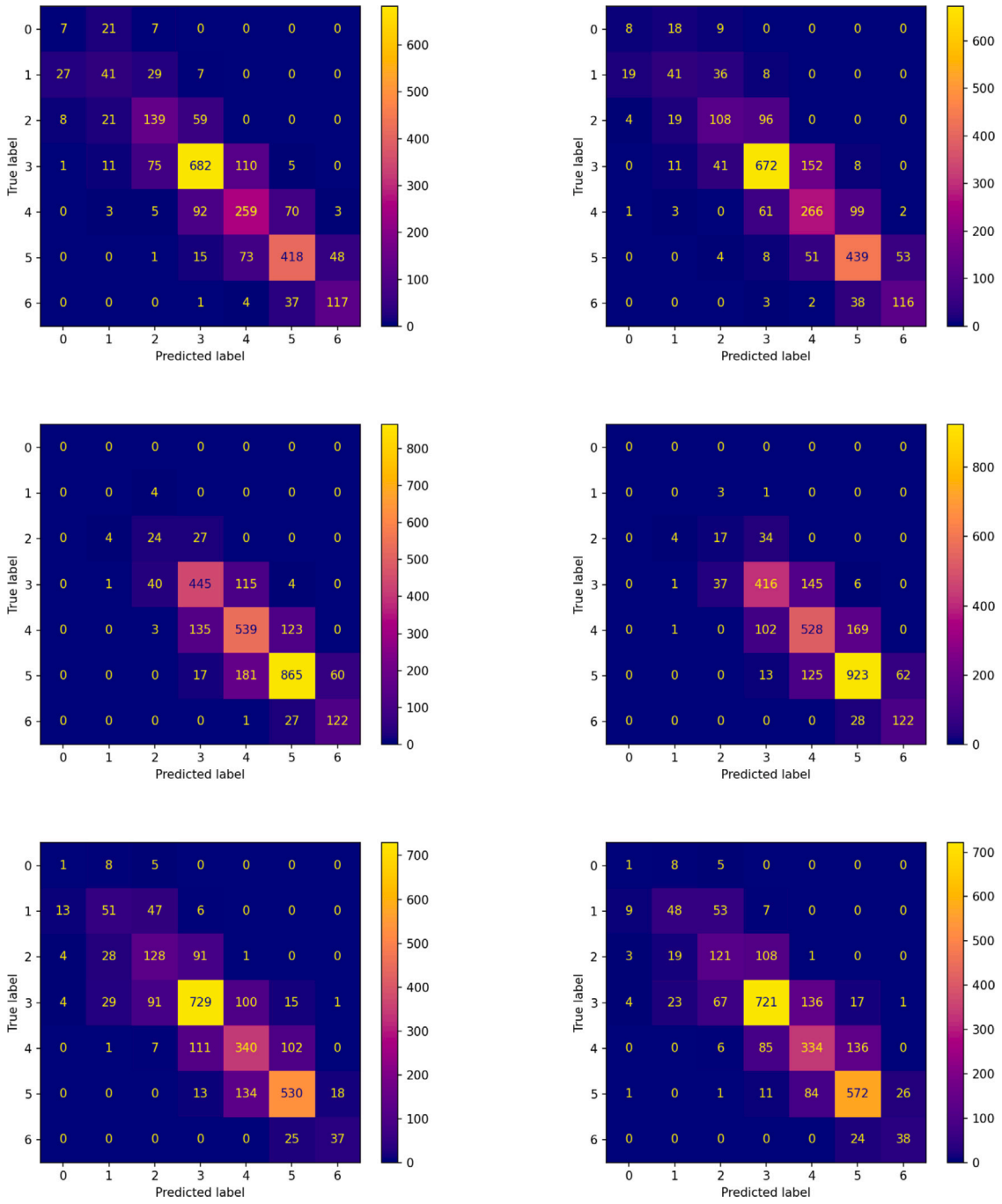


Fig. 6. Confusion matrices on the testing set for patients 575 (top), 588 (middle), and 591 (bottom): on the left, the results for the best global model evolved by MFC-FLEA algorithm, on the right, those for the best local model (model 563).

can make reference to [93] for details on the general statistical analysis reported in this paper and, specifically, on the Quade test.

Any statistical test requires the preliminary choice of a null hypothesis H_0; we have chosen it as the statistical equivalence of the two models found by MFC-FLEA and non-FL. Another choice concerns the significance level α; we have chosen this value as 0.05. This latter choice means, if the statistical test rejects the above null hypothesis H_0, its rejection may be incorrect with a 5% probability.

Table 8 contains the outcome of this statistical test: the second column contains the algorithms compared, while, in the first, their rank values are shown. In such tests, better algorithms exhibit lower ranking values.

This table displays that MFC-FLEA has a much lower rank than the non-FL approach, so it is supposed to be better. Yet, as the computed p-value is 0.12009, higher than the significance level equal to 0.05, this test cannot exclude the statistical equivalence between the two approaches.

The posthoc procedures [93] can be considered to further investigate this issue. Many such procedures exist in the literature and many of them are contained in STAC, i.e., Bonferroni–Dunn, Holm, Hochberg, Finner, and Li. The results, in terms of adjusted p-values,

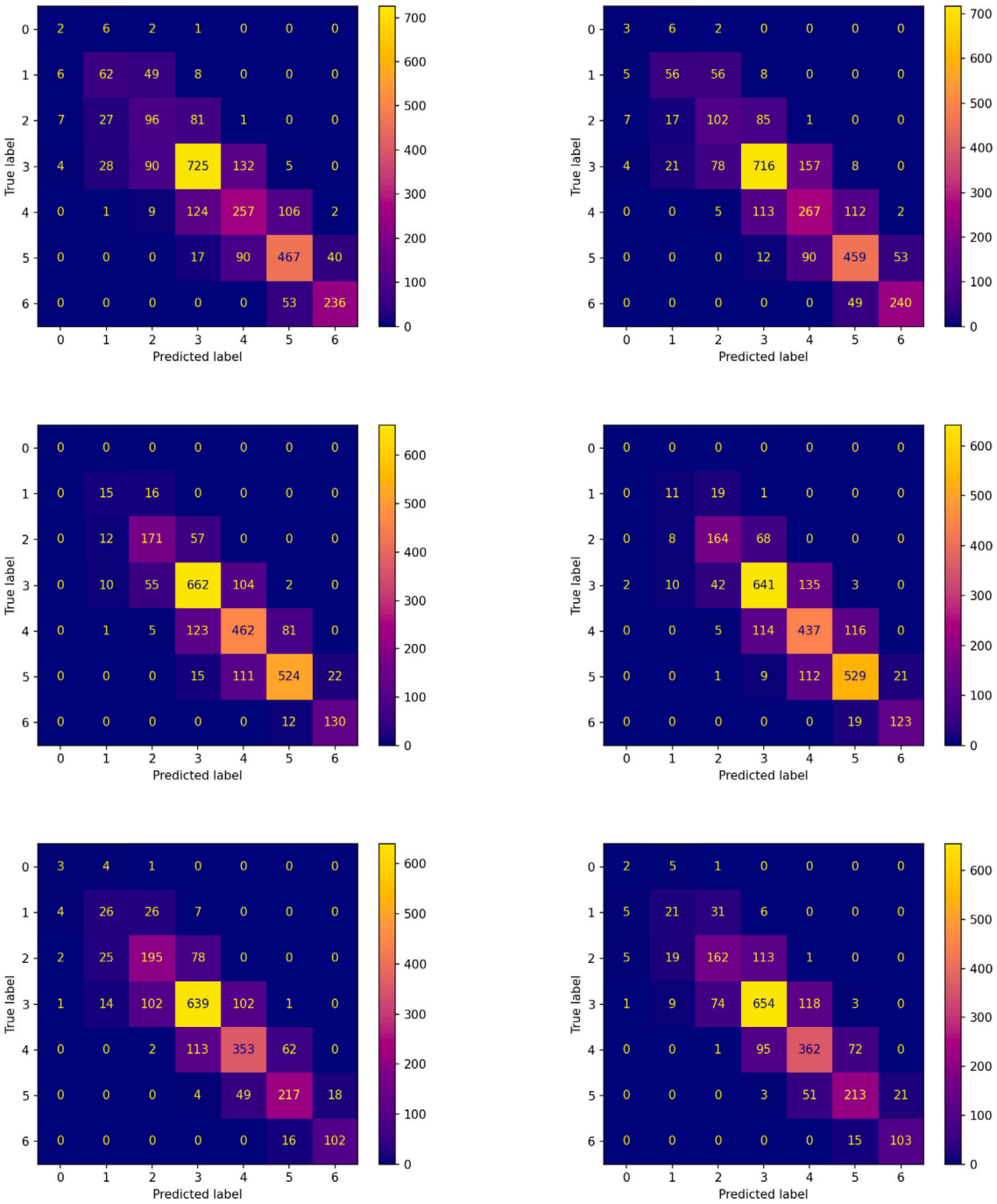


Fig. 7. Confusion matrices on the testing set for the new patients 540 (top), 544 (middle), and 552 (bottom): on the left, the results for the best global model evolved by MFC-FLEA algorithm, on the right, those for the best local model (model 563).

Table 8
Quade ranks test on the results obtained by the two models over the six unseen patients in the Ohio T1DM data set.

Rank	Algorithm
1.14286	MFC-FLEA
1.85714	non-FL
Statistic: 3.50467	p-value: 0.12009

for all the posthoc procedures present in STAC, are shown in Table 9. Each such procedure requires a control method, that is typically chosen as the method with the lowest ranking value provided by Quade; consequently, we have assigned this role to the MFC-FLEA algorithm.

Each posthoc procedure returns an adjusted p-value for non-FL. For each of them, this value is much lower than the α value chosen, i.e., 0.05. This yields that the null hypothesis H_0 of equivalence can be rejected by all the procedures. Therefore, it can be concluded that MFC-FLEA is statistically better than non-FL in terms of generalization ability when unseen patients are considered.

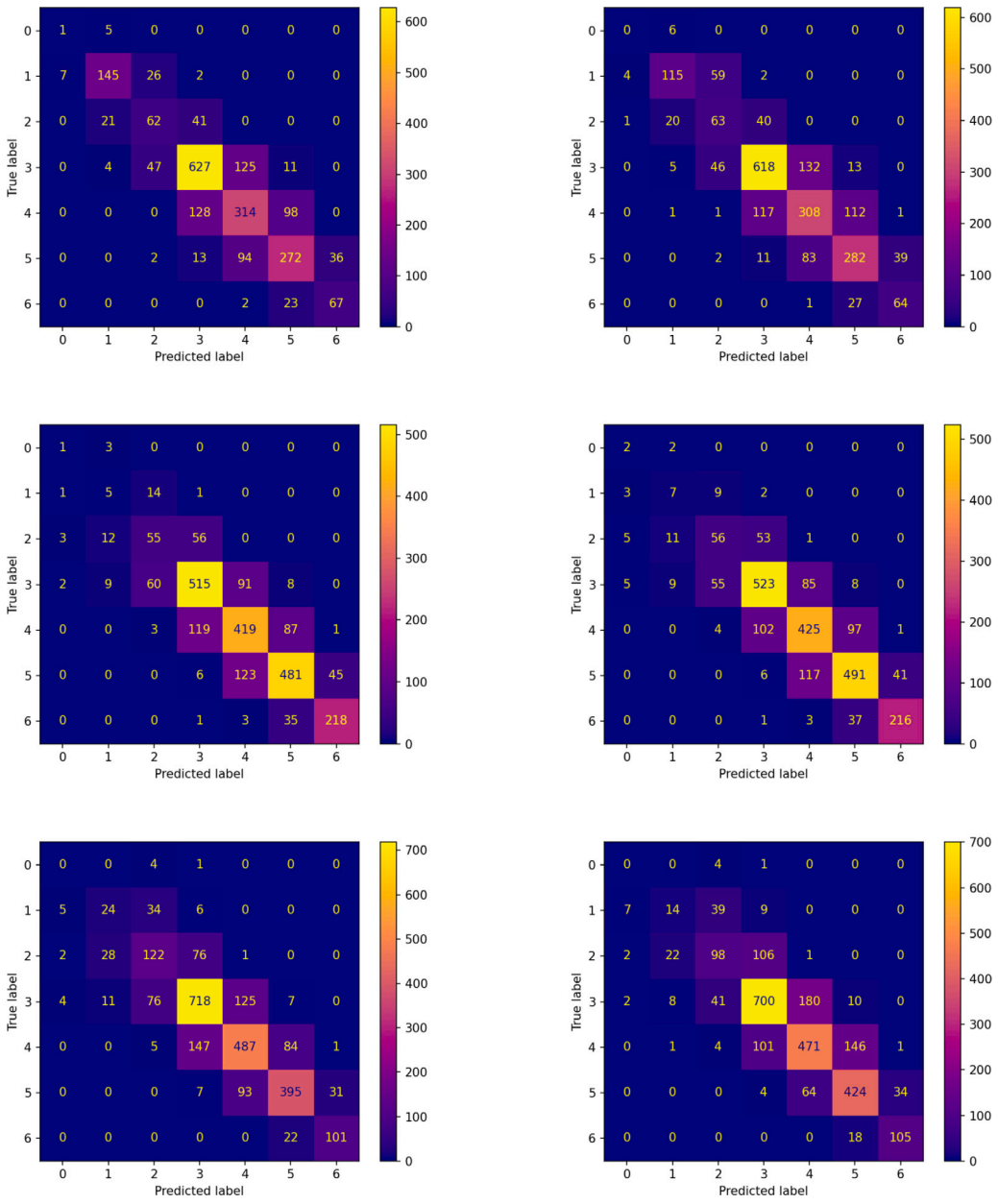


Fig. 8. Confusion matrices on the testing set for the new patients 567 (top), 584 (middle), and 596 (bottom): on the left, the results for the best global model evolved by MFC-FLEA algorithm, on the right, those for the best local model (model 563).

Table 9
Posthoc procedures for Quade test over the six unseen patients in the Ohio T1DM data set: the adjusted p-values. Control method: MFC-FLEA.

	Statistic	Bonferroni-Dunn	Holm	Finner	Hochberg	Li
MFC-FLEA vs non-FL	2.22375	0.02617	0.02617	0.02617	0.02617	0.02617

6.4. Discussion

As a first element of discussion, actually, these results only refer to the Ohio T1DM data set, so these numerical conclusions are limited, and further investigation should be carried out in our next papers to check whether or not this is a general feature of the proposed approach. Should this be the case, an inquiry of the reasons for this would be of high interest. In fact, this FL approach we propose only aims at reducing security problems and was not specifically designed to improve the performance of the classification achieved.

As a second point for discussion, and more importantly from our point of view, our framework has the great advantage that there is no need to transmit models through the communication network, whereas the approach we contrast with requires a final phase that can further pose security at stake. In fact, in order to understand which best local model can be seen as the global model, a series of steps must be taken, as listed in the following. Firstly, each client must send its own local model to the server, which, on its turn, must transmit all the models to all the clients; then, the clients must evaluate each of them (apart from the locally found one, of course) and must send back to the server the fitness value over the local data for each of these models. As a further step, the server must compute the average fitness value for each of these models in order to decide which has the best performance over all the patients considered. Finally, the server must inform each client of which the best model is. This set of steps requires that all the local best models must be transmitted, which could affect security. Hence,

our framework is preferable because it can yield higher security in transmission.

As a further point for a discussion, we have designed our approach aiming at reducing security problems. We are well aware that our mechanism, on its own, cannot completely eliminate these risks from FL. Consequently, further improvement in security can be obtained by also taking advantage of some of the most recent methods, good recent reviews for which are, e.g., [17,18]. As an example, the use of blockchain for cloud-based secure eHealth systems is discussed in [94–96].

Also, our framework currently proposes a final model that is the same for all the clients, thus, for all the participating subjects. In terms of PM, this is good whenever a new patient needs to be evaluated and, if necessary, treated: a general-purpose model, validated on as many subjects as possible, can be immediately used and can propose the actions to be taken for that new subject too. Yet, personalization is another important issue, meaning with this the possibility of providing each patient with a specific model tailored on them. The exploitation of the proposed methodology to achieve better-performing personalized models is an issue that has to be investigated in our future works.

7. Conclusions and future work

The problem of executing machine learning algorithms over data while, at the same time, complying with data privacy is of high relevance in many application areas, among which medicine in general and Precision Medicine in particular.

This paper proposes a new framework for FL that allows performing machine learning and tackling data privacy, while taking a step towards security during communication. The novelty of this new FL lies in its capability to perform collaborative learning without communicating the current model, thus mitigating the related risk of security loss due to potential cyberattacks during transmission. Of course, this new FL framework does not protect from other security threats, e.g., data poisoning or adversarial attacks, while greatly reducing the model poisoning ones. In this way, aggregation naturally emerges by integrating the local performance of the model under evolution.

In addition, the framework can be used with any machine learning algorithm provided that, during the learning phase, the model updating does not depend on the input data.

The experiments, performed by taking a GE algorithm into account, have shown, through the numerical simulations, the effectiveness of the proposed approach, and that it performs better than a non-FL approach. Moreover, this has allowed obtaining explicit knowledge that can be provided to the domain experts to justify the decisions made.

As a test case, glucose values prediction has been considered and has been tackled as a classification problem, the goal being to predict a possible range for any future value, rather than the exact value itself.

The results from the experiments have shown that the FL framework proposed can obtain performance not inferior to the one than is achievable through the development of local models due to a non-FL approach.

Of course, these results, although encouraging, are related to a single medical data set; hence, the conclusions shown here must be further tested by means of a wider experimental phase performed on a larger set of data sets. Therefore, in our next papers, we aim at further investigating our framework on other medical data sets. This will be done by using GE as well as other machine learning methodologies.

CRediT authorship contribution statement

I. De Falco: Methodology, Validation, Formal analysis, Writing – original draft. **A. Della Cioppa:** Methodology, Software, Validation, Investigation, Resources, Data curation, Writing – review & editing. **T. Koutny:** Writing – review & editing. **U. Scafuri:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – review & editing. **E. Tarantino:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgment

We acknowledge financial support from: PNRR MUR project PE0000013-FAIR.

References

[1] F.S. Collins, H. Varmus, A new initiative on precision medicine, New England J. Med. 372 (9) (2015) 793–795.
[2] Academy of Medical Sciences, 2015.
[3] B. Mesko, The role of artificial intelligence in precision medicine, in: Expert Review of Precision Medicine and Drug Development, Vol. 2, No. 5, Taylor & Francis, 2017, pp. 239–241.
[4] A. Ray, Artificial Intelligence and Blockchain for Precision Medicine, Inner Light Publishers, 2018, Retrieved.
[5] K. Ferryman, M. Pitcan, Fairness in Precision Medicine, Data & Society Research Institute, 2018.
[6] C. Krittanawong, H. Zhang, Z. Wang, M. Aydar, T. Kitai, Artificial intelligence in precision cardiovascular medicine, J. Am. Coll. Cardiol. 69 (21) (2017) 2657–2664.
[7] F.D. Beacher, L.R. Mujica-Parodi, S. Gupta, L.A. Ancora, Machine learning predicts outcomes of phase III clinical trials for prostate cancer, Algorithms 14 (5) (2021) 147.
[8] H.K. Bhargava, P. Leo, R. Elliott, A. Janowczyk, J. Whitney, S. Gupta, P. Fu, K. Yamoah, F. Khani, B.D. Robinson, et al., Computationally derived image signature of stromal morphology is prognostic of prostate cancer recurrence following prostatectomy in African American PatientsStroma predicts prostate cancer outcome in African Americans, Clin. Cancer Res. 26 (8) (2020) 1915–1923.
[9] J. Peng, E.C. Jury, P. Dönnes, C. Ciurtin, Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases: Applications and challenges, Front. Pharmacol. 12 (2021) 720694.
[10] A. Kline, H. Wang, Y. Li, S. Dennis, M. Hutch, Z. Xu, F. Wang, F. Cheng, Y. Luo, Multimodal machine learning in precision health: A scoping review, NPJ Digit. Med. 5 (1) (2022) 171.
[11] S.J. MacEachern, N.D. Forkert, Machine learning for precision medicine, Genome 64 (4) (2021) 416–425.
[12] S. Quazi, Artificial intelligence and machine learning in precision and genomic medicine, Med. Oncol. 39 (8) (2022) 120.
[13] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, Z. Lin, When machine learning meets privacy: A survey and outlook, ACM Comput. Surv. 54 (2) (2021) 1–36.
[14] J. Konečnỳ, B. McMahan, D. Ramage, Federated optimization: Distributed optimization beyond the datacenter, 2015, arXiv preprint arXiv:1511.03575.
[15] J. Konečnỳ, H.B. McMahan, D. Ramage, P. Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, 2016, arXiv preprint arXiv:1610.02527.
[16] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A survey on federated learning systems: Vision, hype and reality for data privacy and protection, IEEE Trans. Knowl. Data Eng. (2021).
[17] R. Gosselin, L. Vieu, F. Loukil, A. Benoit, Privacy and security in federated learning: A survey, Appl. Sci. 12 (19) (2022) 9901.
[18] H. Li, C. Li, J. Wang, A. Yang, Z. Ma, Z. Zhang, D. Hua, Review on security of federated learning and its application in healthcare, Future Gener. Comput. Syst. 144 (2023) 271–290.
[19] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, 1996.
[20] T. Back, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, IOP Publishing Ltd, 1997.
[21] D. Whitley, An overview of evolutionary algorithms: Practical issues and common pitfalls, Inf. Softw. Technol. 43 (14) (2001) 817–831.
[22] M. Tomassini, Parallel and distributed evolutionary algorithms: A review, in: Evolutionary Algorithms in Engineering and Computer Science, John Wiley & Sons, 1999, pp. 113–133.
[23] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, J.-J. Li, Distributed evolutionary algorithms and their models: A survey of the state-of-the-art, Appl. Soft Comput. 34 (2015) 286–300.

[24] C. Ryan, J.J. Collins, M.O. Neill, Grammatical evolution: Evolving programs for an arbitrary language, in: European Conference on Genetic Programming, Springer, 1998, pp. 83–96.
[25] M. O'Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. 5 (4) (2001) 349–358.
[26] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215.
[27] I. De Falco, A. Della Cioppa, T. Koutny, M. Ubl, M. Krcma, U. Scafuri, E. Tarantino, A federated learning-inspired evolutionary algorithm: Application to glucose prediction, Sensors 23 (6) (2023) 2957.
[28] C. Marling, R. Bunescu, The OhioT1DM dataset for blood glucose level prediction: Update 2020, in: CEUR Workshop Proceedings, Vol. 2675, NIH Public Access, 2020, p. 71.
[29] B. Pfitzner, N. Steckhan, B. Arnrich, Federated learning in a medical context: A systematic literature review, ACM Trans. Internet Technol. (TOIT) 21 (2) (2021) 1–31.
[30] C.-R. Shyu, K.T. Putra, H.-C. Chen, Y.-Y. Tsai, K.T. Hossain, W. Jiang, Z.-Y. Shae, A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications, Appl. Sci. 11 (23) (2021) 11191.
[31] Y. Kumar, R. Singla, Federated learning systems for healthcare: Perspective and recent progress, in: Federated Learning Systems: Towards Next-Generation AI, Springer, 2021, pp. 141–156.
[32] D.C. Nguyen, Q.-V. Pham, P.N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, W.-J. Hwang, Federated learning for smart healthcare: A survey, ACM Comput. Surv. 55 (3) (2022) 1–37.
[33] R.S. Antunes, C. André da Costa, A. Küderle, I.A. Yari, B. Eskofier, Federated learning for healthcare: Systematic review and architecture proposal, ACM Trans. Intell. Syst. Technol. 13 (4) (2022) 1–23.
[34] D. Kalra, The importance of real-world data to precision medicine, Pers. Med. 16 (2) (2019) 79–82.
[35] I. Mocanu, R. Smadu, M. Dragoi, A. Mocanu, O. Cramariuc, Testing federated learning on health and wellbeing data, in: 2021 International Conference on E-Health and Bioengineering, EHB, IEEE, 2021, pp. 1–4.
[36] H.T. El Kassabi, M.A. Serhani, A.N. Navaz, S. Ouhbi, Federated patient similarity network for data-driven diagnosis of COVID-19 patients, in: 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications, AICCSA, IEEE, 2021, pp. 1–6.
[37] S. Warnat-Herresthal, H. Schultze, K.L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N.A. Aziz, et al., Swarm learning for decentralized and confidential clinical machine learning, Nature 594 (7862) (2021) 265–270.
[38] C. Masciocchi, B. Gottardelli, M. Savino, L. Boldrini, A. Martino, C. Mazzarella, M. Massaccesi, V. Valentini, A. Damiani, Federated Cox Proportional Hazards Model with multicentric privacy-preserving LASSO feature selection for survival analysis from the perspective of personalized medicine, in: 2022 IEEE 35th International Symposium on Computer-Based Medical Systems, CBMS, IEEE, 2022, pp. 25–31.
[39] P. Petsophonsakul, A. Pirmani, E. De Brouwer, M. Akand, W. Botermans, F. Van Der Aa, J.R. Vermeesch, F. Offner, R. Wuyts, Y. Moreau, et al., Augmenting therapeutic effectiveness through novel analytics (ATHENA)–A public and private partnership project funded by the flemish government (VLAIO), 2022.
[40] A.-C. Hauschild, M. Lemanczyk, J. Matschinske, T. Frisch, O. Zolotareva, A. Holzinger, J. Baumbach, D. Heider, Federated Random Forests can improve local performance of predictive models for various healthcare applications, Bioinformatics 38 (8) (2022) 2278–2286.
[41] R. Qureshi, M. Irfan, H. Ali, A. Khan, A.S. Nittala, S. Ali, A. Shah, T.M. Gondal, F. Sadak, Z. Shah, et al., Artificial intelligence and biosensors in healthcare and its clinical relevance: A review, IEEE Access (2023).
[42] T. Hulsen, D. Friedeckỳ, H. Renz, E. Melis, P. Vermeersch, P. Fernandez-Calle, From big data to better patient outcomes, Clin. Chem. Laborat. Med. (CCLM) 61 (4) (2023) 580–586.
[43] W. Hoyos, J. Aguilar, M. Toro, Federated learning approaches for fuzzy cognitive maps to support clinical decision-making in dengue, Eng. Appl. Artif. Intell. 153 (2023) 106371.
[44] Z. Liu, F. Wu, Y. Wang, X. Pan, FedCL: Federated contrastive learning for multi-center medical image classification, Pattern Recognit. 143 (2023) 109739.
[45] C.B. Mawuli, J. Kumar, E. Nanor, S. Fu, L. Pan, Q. Yang, W. Zhang, J. Shao, Semi-supervised federated learning on evolving data streams, Inform. Sci. 643 (2023) 119235.
[46] S. Park, Y. Suh, J. Lee, FedPSO: Federated learning using particle swarm optimization to reduce communication costs, Sensors 21 (2) (2021) 600.
[47] H. Zhu, Y. Jin, Multi-objective evolutionary federated learning, IEEE Trans. Neural Netw. Learn. Syst. 31 (4) (2019) 1310–1322.
[48] Z.-y. Chai, C.-d. Yang, Y.-l. Li, Communication efficiency optimization in federated learning based on multi-objective evolutionary algorithm, Evol. Intell. (2022) 1–12.
[49] Y. Luo, J. Xu, W. Xu, K. Wang, Sliding differential evolution scheduling for federated learning in bandwidth-limited networks, IEEE Commun. Lett. 25 (2) (2020) 503–507.
[50] X. Liu, J. Zhao, J. Li, B. Cao, Z. Lv, Federated neural architecture search for medical data security, IEEE Trans. Ind. Inform. 18 (8) (2022) 5628–5636.
[51] J.Á. Morell, Z.A. Dahi, F. Chicano, G. Luque, E. Alba, Optimising communication overhead in federated learning using NSGA-II, in: Applications of Evolutionary Computation: 25th European Conference, EvoApplications 2022, Held As Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022, Proceedings, Springer, 2022, pp. 317–333.
[52] X. Cai, Y. Lan, Z. Zhang, J. Wen, Z. Cui, W. Zhang, A many-objective optimization based federal deep generation model for enhancing data processing capability in IOT, IEEE Trans. Ind. Inform. 19 (1) (2021) 561–569.
[53] J. Xu, Y. Jin, W. Du, S. Gu, A federated data-driven evolutionary algorithm, Knowl.-Based Syst. 233 (2021) 107532.
[54] A. Suresh, K. Harish, N. Radhika, Particle swarm optimization over back propagation neural network for length of stay prediction, Procedia Comput. Sci. 46 (2015) 268–275.
[55] L. Zaghloul, R. Zaghloul, M. Hamdan, Optimizing artificial neural network for functions approximation using particle swarm optimization, in: Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part I 12, Springer, 2021, pp. 223–231.
[56] G. Cappon, L. Meneghetti, F. Prendin, J. Pavan, G. Sparacino, S. Del Favero, A. Facchinetti, et al., A personalized and interpretable deep learning based approach to predict blood glucose concentration in type 1 diabetes, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 75–79.
[57] T. Zhu, X. Yao, K. Li, P. Herrero, P. Georgiou, Blood glucose prediction for type 1 diabetes using generative adversarial networks, in: CEUR Workshop Proceedings, Vol. 2675, 2020, pp. 90–94.
[58] J. Pavan, F. Prendin, L. Meneghetti, G. Cappon, G. Sparacino, A. Facchinetti, S. Del Favero, et al., Personalized machine learning algorithm based on shallow network and error imputation module for an improved blood glucose prediction, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 95–99.
[59] H. Rubin-Falcone, I. Fox, J. Wiens, Deep residual time-series forecasting: Application to blood glucose prediction., in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 105–109.
[60] X. Sun, M.M. Rashid, M. Sevil, N. Hobbs, R. Brandt, M.-R. Askari, A. Shahidehpour, A. Cinar, Prediction of blood glucose levels for people with type 1 diabetes using latent-variable-based model, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 115–119.
[61] H. Nemat, H. Khadem, J. Elliott, M. Benaissa, Data fusion of activity and CGM for predicting blood glucose levels, in: Knowledge Discovery in Healthcare Data 2020, 2675, CEUR Workshop Proceedings, 2020, pp. 120–124.
[62] T. Yang, R. Wu, R. Tao, S. Wen, N. Ma, Y. Zhao, X. Yu, H. Li, Multi-scale long short-term memory network with multi-lag structure for blood glucose prediction, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 136–140.
[63] D. Joedicke, G. Kronberger, J.M. Colmenar, S.M. Winkler, J.M. Velasco, S. Contador, J.I. Hidalgo, Analysis of the performance of genetic programming on the blood glucose level prediction challenge 2020, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 141–145.
[64] M. Mayo, T. Koutny, Neural multi-class classification approach to blood glucose level forecasting with prediction uncertainty visualisation, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, CEUR Workshop Proceedings, 2020, pp. 80–84.
[65] H. Hameed, S. Kleinberg, Investigating potentials and pitfalls of knowledge distillation across datasets for blood glucose forecasting, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 85–89.
[66] R. Bevan, F. Coenen, Experiments in non-personalized future blood glucose level prediction, in: CEUR Workshop Proceedings, Vol. 2675, 2020, pp. 100–104.
[67] A. Bhimireddy, P. Sinha, B. Oluwalade, J.W. Gichoya, S. Purkayastha, Blood glucose level prediction as time-series modeling using sequence-to-sequence neural networks, in: Knowledge Discovery in Healthcare Data 2020, CEUR Workshop Proceedings, 2020, pp. 131–136.
[68] J. Freiburghaus, A. Rizzotti, F. Albertetti, A deep learning approach for blood glucose prediction of type 1 diabetes, in: Proceedings of the 5th International Workshop on Knowledge Discovery in Healthcare Data Co-Located with 24th European Conference on Artificial Intelligence (ECAI), Santiago de Compostela, Spain, 29-30 August, 2020, pp. 137–141.
[69] H. Khadem, H. Nemat, J. Elliott, M. Benaissa, Multi-lag stacking for blood glucose level prediction, in: Knowledge Discovery in Healthcare Data 2020, Vol. 2675, CEUR-Workshop Proceedings, 2020, pp. 146–150.
[70] I. De Falco, A. Della Cioppa, T. Koutny, U. Scafuri, E. Tarantino, M. Ubl, Grammatical evolution-based approach for extracting interpretable glucose-dynamics models, in: 2021 IEEE Symposium on Computers and Communications, ISCC, IEEE, 2021, pp. 1–6.
[71] N. Ma, Y. Zhao, S. Wen, T. Yang, R. Wu, R. Tao, X. Yu, H. Li, Online blood glucose prediction using autoregressive moving average model with residual compensation network, in: Proceedings of the 5th Annual Workshop on Knowledge Discovery in Healthcare Data, 2020, pp. 151–155.

[72] V. Felizardo, N.M. Garcia, N. Pombo, I. Megdiche, Data-based algorithms and models using diabetics real data for blood glucose and hypoglycaemia prediction–A systematic literature review, Artif. Intell. Med. 118 (2021) 102120.
[73] C. Marling, R. Bunescu, The OhioT1DM dataset for blood glucose level prediction, in: K. Bach, R. Bunescu, O. Farri, A. Guo, S. Hasan, Z. Ibrahim, C. Marling, J. Raffa, J. Rubin, H. Wu (Eds.), International Workshop on Knowledge Discovery in Healthcare Data, third ed., KDH, 2018, pp. 60–63.
[74] I. De Falco, A. Della Cioppa, T. Koutny, M. Krcma, U. Scafuri, E. Tarantino, Genetic programming-based induction of a glucose-dynamics model for telemedicine, J. Netw. Comput. Appl. 119 (2018) 1–13.
[75] I. De Falco, A. Della Cioppa, A. Giugliano, A. Marcelli, T. Koutny, M. Krcma, U. Scafuri, E. Tarantino, A genetic programming-based regression for extrapolating a blood glucose-dynamics model from interstitial glucose measurements and their first derivatives, Appl. Soft Comput. 77 (2019) 316–328.
[76] I. De Falco, A. Della Cioppa, T. Koutny, M. Krcma, U. Scafuri, E. Tarantino, A grammatical evolution approach for estimating blood glucose levels, in: Proc. 11Th IEEE Global Communications Conf. - Int. Workshop on AI-Driven Smart Healthcare (AIdSH), Taipei, Taiwan, 8-10 December, 2020, pp. 1–6.
[77] I. De Falco, A. Della Cioppa, A. Marcelli, L. Stellaccio, U. Scafuri, E. Tarantino, Prediction of personalized blood glucose levels in type 1 diabetic patients using a neuroevolution approach, in: Proc. Genetic and Evolutionary Computation Conference Companion, Lille, France, 10-14 July, 2021, pp. 1708–1716.
[78] S. Contador, J.M. Colmenar, O. Garnica, J.M. Velasco, J.I. Hidalgo, Blood glucose prediction using multi-objective grammatical evolution: analysis of the "agnostic" and "what-if" scenarios, Genet. Program. Evolvable Mach. 23 (2022) 161–192.
[79] A. Della Cioppa, I. De Falco, T. Koutny, M. Ubl, U. Scafuri, E. Tarantino, Reducing high-risk glucose forecasting errors by evolving interpretable models for type 1 diabetes, Appl. Soft Comput. 134 (2023) 110012.
[80] T. Zhu, K. Li, P. Herrero, J. Chen, P. Georgiou, A deep learning algorithm for personalized blood glucose prediction, in: KHD@ IJCAI, 2018, pp. 64–78.
[81] A. Varela Lorenzo, A. Delgado Gutierrez, Glucose Classification and Prediction System with Neural Networks (Ph.D. thesis), University of Madrid, 2020.
[82] T. Danne, R. Nimri, T. Battelino, R.M. Bergenstal, K.L. Close, J.H. DeVries, S. Garg, L. Heinemann, I. Hirsch, S.A. Amiel, et al., International consensus on use of continuous glucose monitoring, Diabetes Care 40 (12) (2017) 1631–1640.
[83] Y. Sun, A.K. Wong, M. Kamel, Classification of imbalanced data: A review, Int. J. Pattern Recognit. Artif. Intell. 23 (4) (2009) 687–719.
[84] N. Chinchor, MUC-4 evaluation metrics, in: Proc. of the Fourth Message Understanding Conference, Morgan Kaufmann, 1992, pp. 22–29.
[85] P. Branco, L. Torgo, R.P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM Comput. Surv. 49 (2) (2016) 1–50.
[86] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[87] G.M. Weiss, Mining with rarity: A unifying framework, ACM Sigkdd Explorat. Newsletter 6 (1) (2004) 7–19.
[88] H. Cramér, Mathematical Methods of Statistics, Vol. 43, Princeton University Press, 1999.
[89] J.R. Landis, G.G. Koch, The measurement of observer agreement for categorical data, biometrics (1977) 159–174.
[90] M. O'Neill, C. Ryan, Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language, Kluwer Academic Publishers, USA, ISBN: 1402074441, 2003.
[91] J.I. Hidalgo, J.M. Colmenar, G. Kronberger, S.M. Winkler, O. Garnica, J. Lanchares, Data based prediction of blood glucose concentrations using evolutionary methods, J. Med. Syst. 2017 41 (142) (2017) 1–20.
[92] I. Rodríguez-Fdez, A. Canosa, M. Mucientes, A. Bugarín, STAC: A web platform for the comparison of algorithms using statistical tests, in: 2015 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, IEEE, 2015, pp. 1–8.
[93] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[94] S. Cao, G. Zhang, P. Liu, X. Zhang, F. Neri, Cloud-assisted secure ehealth systems for tamper-proofing EHR via blockchain, Inform. Sci. 485 (2019) 427–440.
[95] D. Li, D. Han, T.-H. Weng, Z. Zheng, H. Li, H. Liu, A. Castiglione, K.-C. Li, Blockchain for federated learning toward secure distributed machine learning systems: A systemic survey, Soft Comput. 26 (9) (2022) 4423–4440.
[96] A. Qammar, A. Karim, H. Ning, J. Ding, Securing federated learning with blockchain: A systematic literature review, Artif. Intell. Rev. 56 (5) (2023) 3951–3985.