FRAMH: a Federated Learning Risk-based Authorization Middleware for Healthcare

Carlo Mazzocca, *Student Member, IEEE,* Nicolò Romandini, *Student Member, IEEE,* Michele Colajanni, Rebecca Montanari, *Member, IEEE*

***Abstract*—Modern healthcare systems operate in highly dynamic environments requiring adaptable access control mechanisms. Access to sensitive data and medical equipment should be granted or denied according to the current health situation of the patient. To handle the need for adaptable access control of healthcare scenarios, we propose a novel model that allows dynamic access control decisions based on the context characterizing the source, type of access request, patient, and estimated risk corresponding to the conditions of the patient. Estimating patient status risk requires analyzing vital physiological data whose availability is growing thanks to the widespread diffusion of the Internet of Medical Things (IoMT) devices. Inferring the patient health status risk through Machine Learning (ML) techniques is possible, but to achieve better accuracy, the training phase requires the aggregation of vast amounts of data from different sources. This aggregation could be difficult or even impossible due to organization regulations and privacy laws. To address these issues, this paper proposes a novel Federated Learning Risk-based Authorization Middleware for Healthcare (FRAMH) that supports risk-based access control to deal with changing and unforeseen medical situations. Our solution infers the risk of health status through a federated learning (FL) approach enriched with blockchain to avoid the weaknesses of centralized servers. The implemented prototype and a large set of experimental results demonstrate the advantages of FL in estimating the risk in healthcare scenarios. Through this approach, even a medical institution with a limited dataset can achieve a satisfying risk estimation and efficient access control enforcement.**

***Index Terms*—Risk-based Access Control, Access Control, Authorization, Federated Learning, Healthcare**

I. Introduction

THE diffusion of the Internet of Medical Things (IoMT) is playing a key role in the delivery of reliable and effective healthcare services enhancing the wellness of patients and elderly people anywhere and anytime [1]. Many tiny-powered and lightweight wireless sensors can be used to remotely monitor the health condition of a patient and collect vital physiological data [2], which are helpful for emergency medical decisions and chronic disease detection. In such a scenario, caregivers can provide assistance anywhere and anytime. They typically operate in a continuously changing environment with the need to access different data and device resources in various situations. As a consequence, access control models for healthcare, which prevent unauthorized access to data and medical equipment, should be dynamically adaptable to address different circumstances. *Adaptability* to dynamic conditions is a fundamental feature that modern access control frameworks should offer to deal with changing and unforeseen situations. For example, in emergency cases, such as car accidents or when the condition of a patient suddenly worsens and the patient is not already hospitalized, electronic medical records can contribute to saving patient lives. Access to emergency personal information should be granted to ambulance personnel or even to walking-by caregivers accidentally next to the patient, although they would not have permission in other normal situations [3]. Adopting statically pre-configured security policies and mechanisms to prevent data disclosure or access to physical resources may hamper attempts to save patient life. Therefore, the trade-off between privacy and safety is a fundamental aspect to consider while designing and developing authorization solutions.

We consider risk-based access control as a valuable approach to achieving adequate adaptation even in unpredictable situations and conditions. Risk-based access control is a dynamic access control technique that overcomes the limits of traditional authorization approaches by adapting access control decisions according to the level of risk associated to access requests. In healthcare scenarios, the level of risk is typically calculated by considering various elements ranging from the context conditions of the requester to the health status of the patient. One main challenge is the estimation of the risk related to the patient condition because it requires the correlation of a large amount of vital physiological data. The widespread diffusion of IoMT favors this data availability, especially from a future perspective.

Machine Learning (ML) is a promising solution that can be adopted to analyze large datasets and predict the health status of a patient thus supporting medical professional operations [4]. However, the heterogeneity, incompleteness, timeliness, and longevity of healthcare data, in addition to privacy and ownership constraints, open a series of research challenges [5] that tend to hinder the adoption of traditional centralized ML approaches. Current regulations, such as the European GDPR and state privacy laws in healthcare scenarios, prevent that information related to the health of an individual be processed or shared without the explicit consent of the interested party (e.g., Article 9 of the GDPR [6]). These constraints limit the possibility for hospitals and clinics to share information related to patients. As a consequence, the adoption of centralized ML techniques centrally collecting health information from different sources for the training phase are discouraged. In

Manuscript received

The authors are with the Department of Computer Science and Engineering, University of Bologna, 40136 Bologna, Italy (e-mail: carlo.mazzocca@unibo.it; nicolo.romandini@unibo.it; michele.colajanni@unibo.it; rebecca.montanari@unibo.it).

addition, relying only on the amount of data collected by one medical organization may be insufficient to train adequately an ML model. For these reasons, we claim that Federated Learning (FL) can be envisioned as a promising and effective paradigm to take advantage of patient data while preserving their privacy [7]. FL allows the training of a global model in a central server without the need to aggregate data in the server itself. Data are kept locally to the institutions where they originated, thus preserving privacy and ownership. Moreover, we propose to enrich FL with a blockchain system to avoid weaknesses related to a centralized server, such as single point of failure, low scalability, and to limit possible biases inducing to prefer some partial models over others. [8]. A blockchain can be adopted to store and aggregate partial models and to obtain the needed global vision. The characteristics of these technologies can guarantee to every client joining the FL process the trustworthiness of the model employed to predict the patient condition.

In this paper, we present a Federated Learning Risk-Based Authorization Middleware for Healthcare (FRAMH) that allows users to access patient data and equipment according to the context information of the requester and patient including the health status. We use FL to train an ML model that outputs the level of risk related to the current patient condition. All partial models are stored in the blockchain ensuring their integrity and trustworthiness. To the best of our knowledge, we are the first to adopt FL to infer risk related to the patient health status and in general to estimate health status risks in risk-based access control models for healthcare scenarios [9]. We implement a prototype of the proposed middleware and demonstrate with extensive experimental tests the effectiveness of FL to estimate the health status of patients when single medical institutions have limited available data sets.

The remainder of this paper is structured as follows. Section II provides some useful background on FL and blockchain for this paper. Section III discusses related approaches to infer risks in access control models and risk-based access control models for healthcare scenarios. In Section IV, we introduce the adopted risk classification as well as the context modeling on which our proposal relies. Section V presents the architecture of FRAMH, whose implementation is detailed in Section VI. Section VII shows the effectiveness of employing FL in risk-based access control models for healthcare. Finally, Section VIII draws our conclusion with some indications for future work.

II. Background

This section presents some background about the use of federated learning and blockchain which are the two key technologies of our research proposal.

A. Federated Learning

FL is a decentralized machine learning technique that decouples model training from the need to directly access all raw data. This paradigm is suitable to avoid data sharing issues related to law and privacy regulations in healthcare scenarios [10]. Unlike traditional ML which typically relies on centralized data and computational resources, FL does not require that data are sent to a central location. Training is directly performed over remote clients (e.g., server nodes and even powerful devices) using data residing on the premises of the owner. Each client that is involved in FL learning trains a local ML model with its data; then, it sends a partial model to a server that merges them through a suitable aggregation strategy. A similar approach does not require that original raw data are sent outside the perimeter of granted access. FL strategies can be classified according to the algorithm and type of synchronization employed to aggregate partial models.

McMahan et al. [11] presented Federated SGD (FedSGD), an algorithm that applies stochastic gradient descent (SGD) to optimize federated problems. In each round, a client uses local data to perform one step of the gradient descent on its current model, and then sends updates (various gradients) to the server. The authors also proposed FederatedAveraging (FedAVG), a slightly modified version of FedSGD, which sends weights instead of gradients. Such an approach also allows clients to perform multiple updates on local weights before sending them. Most strategies are based on a synchronous aggregation of partial models. Hence, the global model generation is performed only when all the clients have given their contributions. In recent asynchronous approaches [12], the global model is sent back to clients for additional learning round. In summary, we leverage FL because it allows ML algorithms to collect experience from different datasets in different locations, thus enabling multiple organizations to collaborate on model development without having to share private data.

B. Blockchain

Blockchain is a distributed ledger structured as a chain of blocks linked through hashes: each block has a reference of the hash to the previous block. Tempering a block will result in a different hash, thus breaking the hash chain. In our application, a blockchain represents a digital distributed storage where no party can tamper it without being detected as evidenced in [13]. Moreover, blockchain introduces fault tolerance and resilience by design since it is maintained by peer-to-peer networks, without the need for a centralized third party. Each node keeps its copy of the ledger and uses consensus mechanisms, such as Proof of Work (PoW) or Proof of Stake (PoS), to ensure secure synchronization of data across different nodes. These algorithms also discourage malicious users from performing cyber-attacks on healthcare datasets. Blockchain networks can be classified, according to access models, as permissionless or permissioned. The former is characterized by a public network where anyone can interact with it and join the consensus process. On the other hand, in a permissionless blockchain, only authorized parties can interact with the blockchain, depending on granted permissions. In our work, we employed a permissioned blockchain.

Furthermore, a growing number of blockchains support the execution of smart contracts [14] that is, executable code stored and executed on a blockchain-based platform. These programs are fault-tolerant, tamper-proof, and traceable by de-

sign. All these characteristics are important for the applications considered in this paper.

III. Related Work

Most access control models do not satisfy the necessary adaptability required by highly dynamic medical contexts. For this reason, we consider related work proposing risk-based access control solutions and ML techniques for risk estimation with a focus on healthcare scenarios.

Various researchers address issues related to changing situations in healthcare environments. Recently, Atlam et al. [15] presented a novel Neuro-Fuzzy System, which is a combination of an artificial neural network and a fuzzy logic system, to estimate risk in a risk-based access control model and applied it to a children's hospital. Risk is estimated according to the action and the sensitivity of the data involved, while the patient health status condition is not considered in the risk factor evaluations. In [16], the authors proposed a context-sensitive risk-based access control framework that grants or denies access according to the severity of the context. Doctors are bound to a permission set based on the severity of the situation, symptoms, and treatments. Risk is estimated through the correlations between requested data and the corresponding permission profile. Access is granted when the level of risk falls below a threshold (depending on the situation), but there is no ML model to support it. Another risk-based adaptive security solution for healthcare is presented in [17] where the authors refer to risk, estimated through game theory, in terms of the possibility to compromise a medical device when adapting security methods and mechanisms. In these last two proposals, the authors do not explain how the risk is quantitatively estimated and do not present any implementation to demonstrate the applicability and results of their theories. Existing solutions tend to consider access requests to patient data without mentioning other relevant scenarios for healthcare, such as doctors that have to update treatments or a nurse that has to open a medical fridge to administer insulin to a diabetic patient. We propose and implement a risk-based access control framework that enables fine-grained access to data and resources and avoids the limits of previous models. Unlike other research proposals which only discuss qualitatively the risk estimation, we detail how the risk can be estimated through FL models.

Recent research proposed different approaches [9] to estimate the risk, but few of them employ ML. Moreover, to the best of our knowledge, we are the first to propose the adoption of FL in risk-based access control models and to apply it to predict the health status of patients. The authors in [18] use a learned classifier of access control decisions to infer a decision. In their work, the degree of risk is given by the uncertainty that affects the predicted access control decision when there is no exact match between the access request and the corresponding decision. Another paper [19] adopts different ML-based approaches to dynamically decide whether or not a request should be granted or denied in a hospital management system. The best results were obtained by combining auto-encoder for feature selection and random forest for classification. Although the proposed risk-adaptive access control model presents some common features to our solution, it mainly focuses on the authenticity and trust of the requester. It does not take into account the current health status of the patient, representing a key element of our proposal. In [20], the authors presented a fuzzy modeling technique to estimate the risk of each access request. However, the paper does not provide information on how to estimate risk and the construction of fuzzy rules requires prior knowledge of various environmental scenarios.

To sum up, in most existing risk-based access control frameworks for healthcare, independently of whether they use ML or not, the health status of a patient is not included among the risk factors for access control decisions. Related works only discuss qualitatively how risk is practically estimated. Unlike prior works, we base access control decisions even on the current patient condition. Granting or denying access requests according to the health status and using FL to infer such information are two major novelties. Considering the health status of the patient also opens up the possibility to provide outdoor care support to patients anywhere and anytime even when they are on the move or at their homes. As a final consideration, we consider access requests that could target different kinds of resources, and we add implementation details that are often omitted in other research.

IV. FRAMH Access control Model

The FRAMH access control middleware is designed to address the adaptability required by modern healthcare scenarios. In particular, it exploits two kinds of visibility to govern access control decisions depending on the desired trade-off between the need for security and safety: context information related to the requester and the patient to control access to patient data and the health status risk. In the following sections, we describe the FRAMH underlying risk and context models and the needed support services to enforce context-sensitive risk-based access control decisions.

A. Risk Classification

FRAMH classifies the severity of the health status of patients based on the same risk classification proposed in [16] that assumes that a patient condition could be *critical*, *serious*, or *stable*. Adopting a more detailed risk classification could be possible but at the cost of adding complexity to policy management that should govern overlapping policies applying to different health risk levels. In addition, we have not found realistic cases that call for finer-grained risk classification. It is important to observe that FRAMH can work with different risk classifications if needed with minimal modifications. Figure 1 shows the risk levels adopted and their relationship with safety and privacy. It is worth outlining that as the situation becomes riskier, safety outweighs privacy. On the other hand, under a stable situation, privacy is preferred over safety.

- *Critical*: patient life is significantly in danger. In this case, privacy becomes secondary, and requests to access data or medical equipment can be granted. Medical personnel

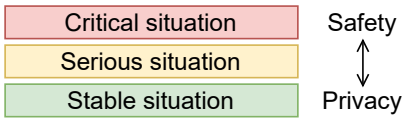

Fig. 1. Considered risk levels.

has to obtain information as soon as possible to do their best and try to save patient life.
- *Serious*: patient conditions are considered urgent. In such a situation, doctors can access patient data from other departments. We grant access to riskier data to expedite patient treatments.
- *Stable*: users can only access authorized data. There is no reason to grant access to additional information since the patient is not facing a life-threatening situation.

The proposed risk classification is adopted to label the outputs predicted by our ML model. Further details about this phase are discussed in Section VII-B.

B. Context Modeling

The context model is used for representing, storing, and retrieving context information that is employed to build access control policies and consequently take access control decisions. To allow a simple policy specification, while still achieving a high degree of flexibility, we adopt the context model reported in Figure 2. The employed context model comprises elements providing all context information along with their attributes that are used to govern access control policies. The *Context Provider* includes context information that will be evaluated to grant or deny access related to both the *Requester* and the *Patient*. The requester is typically an expert caregiver but, in some cases, could also be a relative or friend. For example, in a smart home environment, medicine for treating diabetes or other critical drugs can be kept in a locked smart medical refrigerator [21] that regulates the access to insulin or other drugs and monitor their use by patients. In case of an emergency impeding the patient from personally accessing the medicine in a set time frame, access to the refrigerator could be granted to everyone in the house.

The context provider has context elements that are characterized by a set of attributes. Both the requester and patients have a `physical context` including `time` and `location` context information. The former can be an `instant` or an `interval`, while the latter may refer to an `absolute` or a `relative` geo-position (that is, longitude and latitude or room within a hospital). Information related to time enables regulating access during a specific period, while geographical data can be used to grant or deny access according to the current location occupied by the requester. The requester has its context that comprises: `role`, `department`, and `usual`. The `role` indicates the healthcare professional category or the relationship linking the patient and the requester, such as a familiar or friend. The `department` attribute contains the department information related to the requester specialization. The `usual` attribute outlines if the patient has been already visited by that requester. Context information specific to the patient involves the `treatment` that has been undergoing and the `health status` (critical, serious, or stable).

Our model captures the characteristics of context information based on an ontology such as the Web Ontology Language (OWL) [22]. Domain-independent information of the context ontology (in white boxes) refers to common information that is re-usable for other application domains. In the domain-dependent part (colored boxes), we adopt the major concepts derived from the Health Level Seven (HL7) Reference Information Model [23].

C. Access Control Model Formalization

Concerning the context model of Figure 2 that classifies information belonging to the requester context as requester attributes and those belonging to the patient context as patient attributes. Although the health status refers to the patient context, due to its continuously evolving nature, we classify it as general context information, such as the time and location. Our formal model relies on the following entities:

- *Requester*: we denote with $\mathcal{U}$ the set of requester u;
- *Requester Attribute*: we denote with $\mathcal{RA}$ the set of possible values that a requester attribute ra can assume;
- *Patient Attribute*: we denote with $\mathcal{PA}$ the set of possible values that a patient attribute pa can assume;
- *Resource*: we denote with $\mathcal{R}$ the set of resources r that can be accessed by a requester u;
- *Action*: we denote with $\mathcal{A}$ the set of actions that a requester u can perform on a resource r associated to patient p;
- *Context Parameter*: we denote with $\mathcal{CP}$ the set of possible values that a context parameter cp^i, with $i \in \mathcal{U} \cup p$, can assume. A context parameter is characterized by a name cpn and value cpv, thus, cp is tuple $\langle cpn, cpv \rangle$.

The set of context parameter names $\mathcal{CPN}$ is determined by pre-specified parameter names. In this paper, we consider as context information $\mathcal{CPN} = \{$`Time, Location, HealthStatus`$\}$. Context is defined as follows:

Definition 1 (Context). *A Context C is a set of n context parameters $cp \in \mathcal{CP}$. For each cp_i^k, cp_j^z, with $i \neq j$ and $k = z$, we have that $cp_i^k.name \neq cp_j^z.name$. While, with $i \neq j$ and $k \neq z$, it may result that $cp_i^k.name = cp_j^z.name$. In a context C, there cannot be two cp belonging to the same requester u with the same name, while this may happen for a requester and a patient p.*

Context information has a primary role in dynamically granting or denying access to resources. In risk-based access control policies, such information enables adapting to different circumstances overcoming the limits of rigid static access control approaches designed to always apply the same access control policies that do not consider environmental changes and unpredictable situations. We formally define an access control policy as follows:

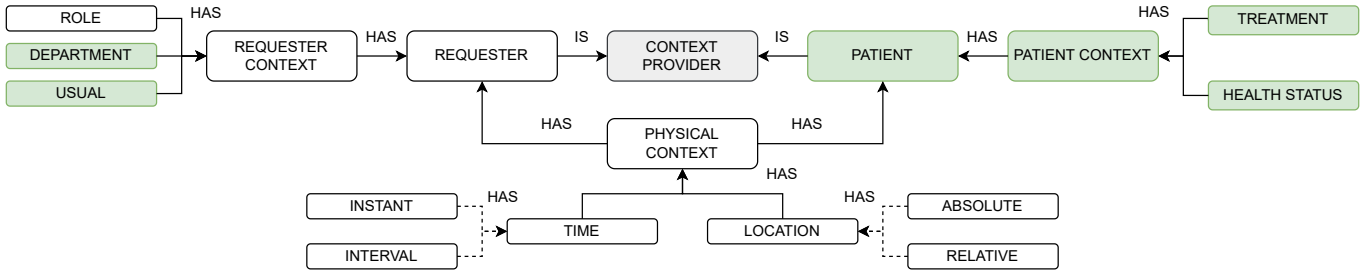

Fig. 2. The context model employed to build risk-based access control policies.

TABLE I
EXAMPLES OF RISK-BASED CONTEXT-SENSITIVE POLICIES.

Requester Attribute	Patient Attribute	Resource	Action	Context Information
Role: "Doctor", Hospital: "X" Department: "Orthopedic"	Treatment: "Orthopedic" Hospital: "X"	Data: Neurological	Read:False, Delete: False	HealthStatus: "Stable"
Role: "Doctor", Hospital: "X" Department: "Orthopedic"	Treatment: "Orthopedic" Hospital: "X"	Data: Neurological	Read:True, Delete: False	RequesterLocation *is equal to* PatientLocation, HealthStatus: "Serious"
Role: "Nurse", Hospital: "X" Department: "Diabetes"	Treatment: "Diabetes" Hospital: "X"	Equipment: Refrigerator	Open:True, Close: True, IncreaseTemperature: True, DecreaseTemperature: True	RequesterLocation *is equal to* PatientLocation, HealthStatus: "Stable"

Definition 2 (Access Control Policy)**.** *An access control policy ap is a set of clauses given by* $RA \cup PA \cup A \cup R \cup C$ *with* $RA \subset \mathcal{RA}$*,* $PA \subset \mathcal{PA}$*,* $A \subset \mathcal{A}$*, and* $R \subset \mathcal{R}$*.*

A requester u with attributes RA can perform the actions A, under the context C, on the patient resources PR whose owner has attributes PA.

For the sake of clarity, in Table I, we report some examples of risk-based context-sensitive policies, such as the first access control policy claims an orthopedic of hospital X cannot read neurological data of a patient whose conditions are stable. The adopted mechanism allows many-to-many access control since policies are not bound to a specific requester or patient.

V. FRAMH ARCHITECTURE

FRAMH architecture consists of three layers: *Authorization*, *Patient*, and *Learning* which collaborate to manage effective risk-based access control. The Authorization Layer provides the components employed to verify access requests and enforce access control decisions. The Patient Layer offers services for the collection of patient context data. Finally, the Learning Layer, based on FL integrated with the blockchain, is in charge of supporting the prediction of the level of risk associated with the current patient condition. Figure 3 shows the FRAMH architecture.

A. Authorization Layer

The Authorization Layer includes different components belonging to the XACML architecture [24] as shown in Figure 3.

1) Policy Enforcement Point: The Policy Enforcement Point (PEP) handles all incoming access requests. It extracts information from the requests and builds a query that can be understood by the Policy Decision Point (PDP). Furthermore, if the incoming request does not provide all the needed data, then PEP collects additional information from external sources that are included in the query to evaluate.

2) Policy Decision Point: The PDP is the entity responsible for evaluating incoming requests and determining whether they should be granted or denied. It has to be fed up with policies and data needed to make access control decisions.

3) Policy Information Point: The Policy Information Point (PIP) is a component deployed on each node that provides the additional information required by the PDP to make an access control decision. In particular, it provides patient information including those related to the current health status predicted by the global model.

4) Policy Administration Point: The Policy Administration Point (PAP) is the service that allows administrators to manage policies and upload data that are used during policy evaluation.

B. Patient Layer

The Patient Layer includes the Patient device and a FRAMH Agent as described below.

1) Patient Device: A personable wearable IoMT device that continuously monitors the health of a patient. It periodically sends patient data to its associated FRAMH Agent.

2) FRAMH Agent: Each Patient Device has a FRAMH Agent associated with it. FRAMH Agent is a local component deployed within smart homes, hospitals, and/or clinics that comprises all services for calculating the risk level of the patient health status and for providing patient data along with

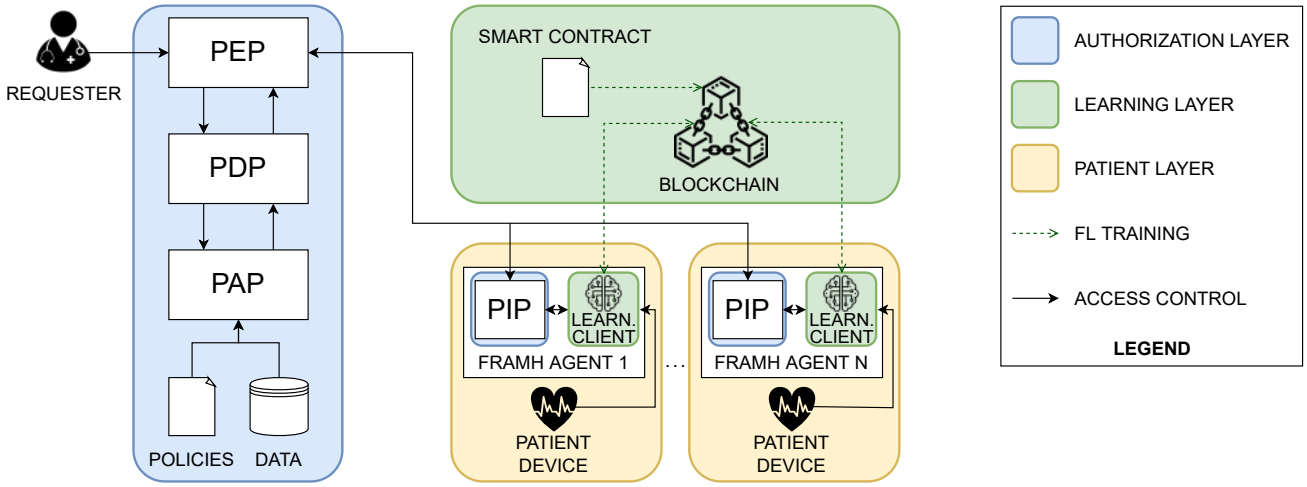

Fig. 3. The FRAMH architecture with the Authorization, Patient, and Learning layers highlighted with different colors.

the current health status to other FRAMH architecture modules. The FRAMH Agent also comprises the Learning Client module (see Section V-C). Every FRAMH Agent receives vital physiological data from the Patient Device. Such information is periodically provided as an input to the global model, embedded in the Learning Client, that outputs the health status of the patient. The FRAMH Agent grants access to these data if the access request satisfies the corresponding access control policy. When deemed by the PDP, it sends context information of the patient needed to make access control decisions.

C. Learning Layer

Let us analyze the main components that are involved in the learning process.

1) Blockchain: We use a blockchain because it guarantees transparency, fairness, and impartiality even among untrusted parties that collaborate with the FL process (e.g., [25]). In addition, by relying on smart contract technology, we eliminate the need for a centralized entity that is typically responsible for aggregating all partial models in the traditional FL approach. The adoption of smart contracts allows clients to publish partial models on the blockchain. It is the smart contract that aggregates them in an automated way. The blockchain also stores all the versions of the global model. One global model version can be retrieved by looking at the transaction history. One advantage is, for example, that if there is a drop in performance during the training phase due to overfitting, it is possible to restore the best version of the model. Thanks to blockchain features of immutability and non-repudiation, all changes to the partial and global models can be easily tracked. Furthermore, when keys used to sign transactions are linked to physical entities, it is possible to provide accountability during the FL process. These properties are relevant in healthcare contexts, where an error in the model generation may have even legal consequences. The use of a blockchain also prevents a malicious user may contribute to generating the global model without being detected. An attacker may, for instance, provide a partial model with some backdoor to corrupt the performance of the global model on specific sub-tasks [26] (e.g., classifying critical patient condition as stable).

Although the legal nature of blockchain is still debatable, some countries have begun to admit data saved on blockchain as a piece of legal evidence. For example, on June 27, 2018, ruling on *Hangzhou Huatai Yimei Culture Media Co., Ltd. ("Huatai") v. Shenzhen Daotong Technology Development Co., Ltd. ("Daotong")* [27], the Hangzhou Internet Court was the first to accept electronic data stored on a blockchain as legal evidence in a process.

2) Learning Client: Learning Clients (also referred to as clients) are the entities involved in the training process. They use vital physiological data of patients to train locally an ML model. The training takes advantage of both local decisions (optimization algorithm) and global decisions (algorithm for aggregating the various partial models). The latter discriminates which data should be published on the blockchain as well as the frequency of publication. Once the local training has ended, clients rely on smart contract support to publish their results. The blockchain server exploits information provided by the clients to update the global model which is then sent back to clients and used as a starting point for the next round of local training. Once the training has ended, Learning Clients are provisioned with a global model that enables them to infer the health status of patients.

D. FRAMH at Work

The FRAMH middleware can manage access control policies and dynamically verify access requests according to the current circumstance. We provide a simple but effective example that highlights the adaptability of the proposed system to deal with different situations while performing access control verification, which is the core function of our proposal. Let us consider the case of an orthopedic who wants to access the neurological data of a patient who is suffering from some mobility problems. The doctor works at hospital X where the patient is hosted. The following steps have to take place a priori before any resource access attempt:

1) Each FRAMH Agent trains a local model using the data collected by monitored patients. Once the final global model has been generated and sent to FRAMH Agents, FRAMH is ready to receive and evaluate access requests;
2) The wearable IoMT device, which monitors the patient, periodically sends vital physiological data to the FRAMH Agent that provides them with the global model that in turn outputs the current health status of the patient. Therefore, when a new access request arrives the FRAMH agent is aware of the patient condition.

At run time, when the orthopedic requires access to the patient data the access control process comprises the following phases:

1) The doctor sends an access request, including her/his information, to the PEP;
2) The PEP collects the data provided by the requester as well as further context information (i.e., location and timestamp of the request). Moreover, it retrieves, through the PIP, the health status as well as other context information of the patient associated with that access request. Let us assume that, when the orthopedic tries to access the patient neurological data, the context information provided by the PIP states that the patient condition is stable;
3) The PEP uses the retrieved information to build a query that can be interpreted by the PDP;
4) The PDP evaluates the access request sent by the PEP according to the corresponding policy. Policy and data that are used by the PDP during the access control verification are provided by the PAP. In this case, according to the first policy reported in Table I, the request is denied since the doctor is attempting to access data that goes beyond her/his specialization without a satisfying reason.

Let us assume that the following day, the orthopedic tries to access again the same piece of information. The access request goes through the same steps described above. However, if the patient condition has turned serious, for example, due to a sudden worsening of her/his vital parameters, and the doctor is sending the request from the same location as the patient. The orthopedic will be justified to access neurological data as claimed by the second policy of Table I.

VI. Implementation

In this section, we present the implementation details of FRAMH.

A. Authorization Layer

1) PEP: The PEP was implemented through Flask[1] a web micro-framework written in Python that offers REST API. It obtains the context information of the requester thanks to Python libraries, while those related to the patient are retrieved through the PIP. Such information is used to build a query (JSON payload) that will be evaluated by the PDP. In Listing 1, we report the structure of an access request that matches the second row of Table I.

[1]https://flask.palletsprojects.com/en/2.1.x

Listing 1 Example of access request

```
{
    "requester_attributes":{
        "role":"doctor",
        "hospital":"X",
        "department":"orthopedic"
    },
    "patient_attributes":{
        "treatment":"orthopedic"
    },
    "resource":{
        "type":"data",
        "value":"neurological",
        "action":"read"
    },
    "context":{
        "health_status":"serious",
        "user_location":"X",
        "patient_location":"X"
    }
}
```

Listing 2 Example of access control policy

```
allow {
  doctor_is_orthopedic
  is_serious
  access_neurological_data
}

doctor_is_orthopedic {
  requester := input.requester_attributes
  requester.role == "doctor"
  requester.hospital == "X"
  requester.department == "orthopedic"
}

is_serious {
  cont := input.context
  cont.health_status == "serious"
  cont.user_location == cont.patient_location
}

access_neurological_data {
  patient  := input.patient_attributes
  resource := input.resource

  patient.treatment[_] == "orthopedic"
  resource.type == "data"
  resource.value == "neurological"
  resource.action == "read"
}
```

2) PDP: For our PDP, we employed Open Policy Agent (OPA)[2], a lightweight general-purpose policy engine service that decouples policy decision-making from policy enforcement. Policies are written in Rego[3], the native policy language of OPA that support structured document models such as JSON. For the sake of clarity, we provide, in the snippet below, the implementation of the second access control policy of Table I. As shown in Listing 2, Rego enables implementing fine-grained access control policies. In order to show its flexibility, we implemented the aforementioned policy by combining *doctor_is_orthopedic*, *is_serious*, and *access_neurological_data*. The former verifies that the requester is an orthopedic who works for the X hospital. The second checks that the patient condition is serious. Finally, the latter defines the patient

[2]https://www.openpolicyagent.org
[3]https://www.openpolicyagent.org/docs/latest/policy-language

attribute that allows accessing neurological data. The `input` keyword enables accessing parameters of the JSON payload, while the underscore _ is a special Rego iterator, in this case, it looks for "orthopedic" within the patient treatments. In Listing 2, only to favor its comprehensibility, we compared the request's fields to predefined values. However, these comparisons are typically performed between the request's fields and those memorized in JSON files provided, as well as policies, by the PAP.

3) PIP: As the PEP, also the PIP was implemented through Flask. It periodically receives patient data that are fed up to the global shared model to predict the health status of the patient. The PIP exposes REST API to collect data from the patient data and to provide patient and context information to the PEP.

4) PAP: Our PAP comprises CouchDB[4], an open-source document oriented NoSQL database, and *bundle server*. The former stores access control policies and data that are deemed by the PDP during policy evaluation, while the latter subscribes to changes of access control policies memorized in CouchDB and consequently updates the cache. The bundle server is the component that provides policies and data to the PDP.

B. Learning Layer

1) Blockchain: The blockchain was implemented through Hyperledger Fabric[5], an open-source, modular, and extensible framework for deploying permissioned blockchains. Using a permissioned blockchain allows for more fine-grained control over the operations performed in the network. Every node that wants to submit transactions must have an identity issued by a known Certificate Authority (CA). Furthermore, the adoption of a permissioned blockchain allows checking partial models submitted by clients and eventually excludes them from the model aggregation phase. Fabric enables executing smart contracts written in general-purpose languages, such as Java, Go, and NodeJS. Moreover, Fabric-based applications are enterprise-grade and offer a high level of security, scalability, confidentiality, and performance. Indeed, as shown in recent work [28], Fabric can support about 200 transactions per second, with an average latency of about 0.16 seconds, and up to 100,000 participants.

We achieve the FL process through a smart contract, written in NodeJS, that implements the FedAVG algorithm. Each client sends an array containing the weights and biases of the neural network trained locally. For the sake of simplicity, the aggregation was implemented synchronously: the smart contract waits for all the models from the clients and, once it receives the last one, it automatically aggregates them, generating the global model. Finally, to notify the successful aggregation, the smart contract emits an event by exploiting Fabric's event service and the clients download the global model using the corresponding method of the smart contract. The use of a blockchain does not negatively impact the performance of our proposal. The global model has to be trained before putting FRAMH into execution and possible model updates have to take place only during the specified time window.

2) Learning Client: Entities participating in the FL process and developed to be Flower [29] compliant. Flower is a framework that offers an FL infrastructure to ensure low engineering effort that enables developers to focus only on ML aspects. It is compatible with the most widely used ML frameworks, such as Tensorflow and Pytorch. Moreover, Flower is designed to simulate realistic situations with a large number of heterogeneous devices having different computational capabilities and ecosystems. To enable the communication between Hyperledger Fabric and the Flower Client, we developed an adapter, transparent to the client, that enables the exchange of data with the blockchain. The Flower Client creates a gRPC[6] connection with the adapter, which acts as a Flower Server and forward data to the blockchain.

3) Optimization Techniques: One of the most significant limitations that hinder the use of blockchain for ML is the size of transactions. Indeed, although Fabric is one of the few blockchains that allows large transaction sizes (around 100MB), ML models can easily go beyond this threshold. For example, some models available on Keras[7] reach up to 500MB in size. To address such concerns, we adopted the following model size optimization techniques:

- Model Compression: this technique can be used with any type of data. In our case, it consists of compressing the global and partial models before saving them on the blockchain. We used *pako*[8], a porting of *zlib* for NodeJS. *zlib* is a free, open-source software library for lossless data compression and decompression. It is based on the DEFLATE algorithm [30], which uses a combination of the LZ77 lossless data compression algorithm and the Huffman coding;
- Model Quantization: this method is specific to neural network weights. Although compression is generally useful in decreasing the size of data, in the case of ML models it turns out to be underperforming. The *float32* weights of a neural network are usually not suitable for such compression due to the noise-like variation in the parameter values, which contains few repeating patterns [31]. For this reason, we decided to quantize the weights of the neural network after performing local training. Quantization refers to the operation of mapping the 32-bit float values of the original network weights to a more compact representation, such as 8-bit integers. In our case, we mapped the weights ($w \in [\alpha, \beta]$) to 8-bit signed integer values ($w_q \in [\alpha_q, \beta_q]$), with $\alpha_q = -128$ and $\beta_q = 127$. The quantization process is defined, for each weight, as follows :

$$w_q = round(\frac{1}{s}w + z) \quad (1)$$

and the de-quantization process is defined as:

$$w = s(w_q - z) \quad (2)$$

[4]https://couchdb.apache.org
[5]https://www.hyperledger.org/use/fabric
[6]https://grpc.io/
[7]https://keras.io/api/applications/
[8]https://github.com/nodeca/pako

where s is the scale and z is the zero point [32]. The scale is an arbitrary positive real number, while the zero point is the quantized value corresponding to the real value 0. It is possible to derive s and z as follows:

$$\begin{cases} s = \dfrac{\beta - \alpha}{\beta_q - \alpha_q} \\ z = round(\dfrac{\beta\alpha_q - \alpha\beta_q}{\beta - \alpha}) \end{cases} \tag{3}$$

Since some neural networks have weights that differ by orders of magnitude between layers, s and z are calculated by considering one layer at a time. This approach improves the accuracy of the quantization process and makes it less subject to variance in values.

VII. Evaluation

We evaluate the feasibility of applying the proposed FL to estimate the level of risk corresponding to the current patient conditions. We first describe the employed dataset, then we introduce the adopted ML model. Finally, we discuss experimental results and draw some considerations.

A. Dataset

We used the public dataset from the PhysioNet/Computing in Cardiology Challenge 2012 *Predicting Mortality of ICU Patients* [33]. This dataset includes the medical records of 12,000 intensive care unit (ICU) patients who have survived or passed away. All patients were adults and were hospitalized for several reasons. For each patient, data were collected for 48 hours after their admission to the ICU. Observations covered 42 different variables, which included information from laboratory tests or non-invasive examinations. However, not all variables were collected every hour of hospitalization.

Before performing the training, we pre-processed the dataset. First, we followed the data cleaning and feature extraction process described in [34]. Then, we checked each patient measurement to correct any errors through domain knowledge. Physiologically implausible values were replaced with valid measures or *NaN*. Finally, we transformed the time series of the individual variables into scalar features. We extracted the following features for each temporal variable: minimum, maximum, median, first and last values. Furthermore, to simulate a scenario where patients are monitored through IoMT devices, we used features that can be monitored through non or minimally invasive readings. Non-invasive means that there is no need to cut the skin or enter any of the body spaces to measure a vital parameter. For this reason, we only take into account the following features: age (Age), glucose in blood (Glucose), heart rate (HR), non-invasive diastolic arterial blood pressure (NIDiasABP), non-invasive mean arterial blood pressure (NIMAP), non-invasive systolic arterial blood pressure (NISysABP), respiration rate (RespRate), O_2 saturation in hemoglobin (SaO2), and temperature (Temp). Each feature X was normalized so that each value x is mapped to the range $[0, 1]$:

$$x_{norm} = \frac{x - min(X)}{max(X) - min(X)} \tag{4}$$

Not all measurements were available for all the patients and this produced missing data in the dataset. For this reason, we first eliminated those patients with at least two-time variables that have never been collected. In such a way, the dataset was reduced from 12,000 to nearly 8,000 patients. In addition, we replaced the *NaN* values with the median for each feature.

As a remarkable issue, we note that the dataset is highly unbalanced since it contains many more survived than dead patients. After various transformations, the ratio between deceased and survivors during the period is approximately 1:6. There are many techniques to handle unbalanced data sets (e.g., [35]). Since we decided to only use real data and avoid those synthetic, we performed a random under-sampling of the dominant class to obtain a balanced dataset. This step further reduced the size of the dataset to 2254 patients.

B. Model

As our ML model, we use the Multilayer Perceptron (MLP), which is a fully connected class of feedforward Artificial Neural Networks (ANNs). The choice of hyper-parameters, such as the number of layers, neurons, and initial learning rate, is crucial to obtaining a performing model. There is no unique strategy for discovering the optimal configuration. However, some recommendations can be followed [36]. In this work, we adopt a search technique based on a manual trial and error of different hyper-parameter configurations. The best configuration found consists of four dense layers, with 256, 512, 128, and 1 neuron, respectively. The first three layers use Rectified Linear Unit (ReLU) as the activation function [37], while the last layer (the output layer) adopts the Sigmoid to obtain a value between 0 and 1, representing the probability of the patient death. In addition, layers using the ReLU function are initialized using the He Normal initializer, while the last layer is initialized using the Glorot Normal initializer. All biases are initialized to zero. As the optimization algorithm, we employed Adam, which is a stochastic gradient descent method based on adaptive estimation of the first-order and second-order moments [38], with a learning rate of 0.01 with a decay of 0.005. As demonstrated in [39], decreasing the learning rate during training contributes to achieving better performance. Finally, since we aim to address a binary classification problem, we used binary cross-entropy as the loss function. The output value o of the model is mapped into the risk levels shown in Figure 1 as follows:

$$\begin{cases} o \in [0{,}0.33[\rightarrow Stable \\ o \in [0.33{,}0.66[\rightarrow Serious \\ o \in [0.66{,}1] \rightarrow Critical \end{cases} \tag{5}$$

As anticipated in Section V, vital physiological data collected through IoMT devices are provided to the global model that outputs one of such risk levels that is then used by the PDP to make access control decisions.

C. Experiments

To show the effectiveness of FL in healthcare contexts, we applied our model in both centralized and federated configurations. The dataset was split into 90% training data and

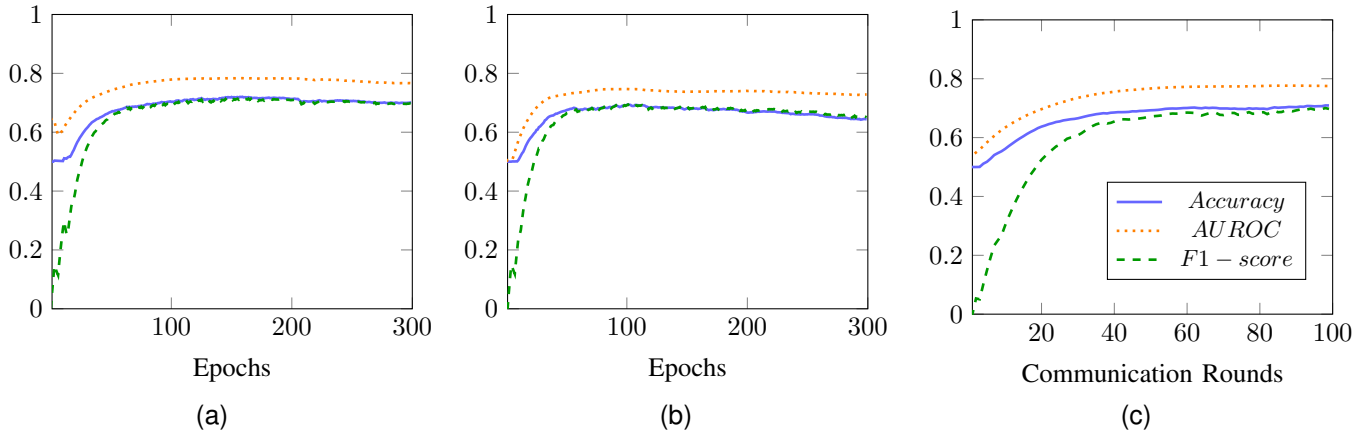

Fig. 4. Performance comparison of the centralized approach with 100% of the training set (a), centralized approach with 75% of the training set (b), and federated approach with 3 clients each using 33% of the training set (c).

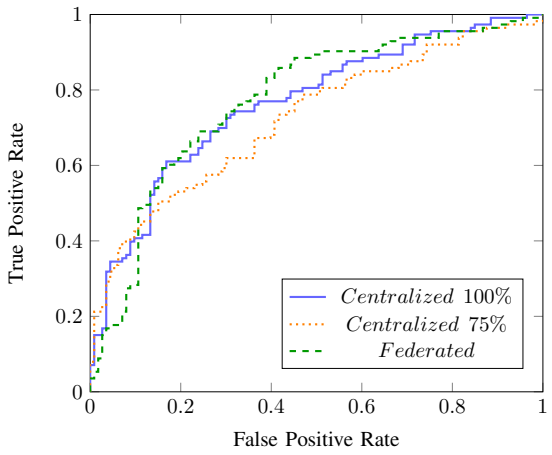

Fig. 5. ROC comparisons of the centralized approach with 100% of the training set, centralized approach with 75% of the training set, and federated approach with 3 clients each using 33% of the training set.

10% testing data. We first defined a performance benchmark by performing centralized training using the entire training set. In the centralized approach, a single client, which could be a hospital, performs the training process for 300 epochs. However, since the hospital may have a smaller training set, we ran a second experiment using 75% of the training data.

To evaluate the federated approach, we considered three clients, that can represent three hospitals of the same region, that join the federated learning process. Each hospital uses 33% of the training data, with no overlap, to perform the training of a partial model. On each client, the model is trained for 3 epochs before being sent for the aggregation, while the global training involves 100 rounds of communication. Thus, we are in the same condition as the centralized approach, since overall each model will be trained for 300 epochs.

D. Results

In Figure 4 we report the main results of the experiments. From the three graphs, we can state that the model obtained through FL achieves comparable performance with that of the centralized model that has been trained with the entire training set. Interestingly, the centralized model trained with 75% of the training set performs worse than the federated one. Having less data will result in a worse ML model, and a single clinic cannot have an amount of data comparable to a federation of medical institutions that collaborate to achieve a common goal. This realistic observation justifies our choice to adopt FL to infer the health status of patients in healthcare scenarios where sharing sensitive data is one of the major concerns. On the basis of the collected results, we can state that medical institutions having a small-size dataset can remarkably benefit from joining the FL process to infer the risk related to the current patient condition. Although each client trains a model using a restricted dataset, this knowledge is shared through the aggregated model with other peers. Thanks to the proposed process, medical institutions can achieve comparable results to a centralized ML approach as they employed the whole amount of data produced by all involved organizations.

TABLE II
COMPARISON OF ML APPROACHES THAT USE THE MIMIC-III DATASET.

Project	Dataset	Model	Performance
Centralized 100%	Subset of MIMIC-III	MLP	Accuracy = 0.70 AUROC = 0.76 F1-score = 0.70
Centralized 75%	Subset of MIMIC-III	MLP	Accuracy = 0.64 AUROC = 0.72 F1-score = 0.65
Federated	Subset of MIMIC-III	MLP	Accuracy = 0.70 AUROC = 0.77 F1-score = 0.70
Baker at al. [40]	Complete MIMIC-III	CNN-BiLSTM	Accuracy = 0.76 AUROC = 0.85
Sadeghi et al. [41]	Complete MIMIC-III	Decision Tree	AUROC = 0.93 F1-score = 0.91
Brand et al. [42]	Complete MIMIC-III	CNN	AUROC = 0.87 F1-score = 0.77

Furthermore, in Table II, we compare the obtained results with those of the state-of-the-art research proposals that use the MIMIC-III dataset or derivatives and only consider vital signs. Our comparisons are based on three metrics widely

adopted in literature: Accuracy, Area Under the Receiver Operating Characteristics (AUROC), and F1-score. AUROC is a performance measurement for classification problems at various threshold settings. It tells how much the model is capable of distinguishing between classes. Figure 5 reports the ROC of our experiments, the graphs highlight that the AUROC of the centralized approach using 100% of the training set and that of the federated one are larger than that of the centralized configuration using 75% of the training set. Thus, this implies that the centralized approach that uses 75% of the training set makes more wrong predictions than the other two configurations. The F1-score can be interpreted as a harmonic mean of precision and recall. The precision is the number of true positives divided by the number of all positives; while the recall is the number of true positives divided by the number of all samples that should have been identified as positive. In healthcare scenarios predicting false negatives and effectively distinguishing between patients with the disease and no disease are two serious concerns. For this reason, the F1-score and the AUROC are the two key metrics to consider while evaluating the ML model. Some of the papers considered, however, do not show all the metrics. AUROC is the only one present in each of them.

The model presented in [40] was trained to predict three different cases of risks of mortality: within 3 days, 7 days, and 14 days respectively. Table II shows the performance of risk mortality within 14 days. The model consists of a hybrid neural network, composed of a Convolutional Neural Network (CNN) and a Long Short Term Memory (LSTM). CNNs are widely used to identify patterns, while LSTM networks are known for their ability to remember which information in a sequence is the most important. The data used for training are the readings of some vital signs recorded over a 24-hour window. In [41], the authors trained eight different classifiers: decision tree, linear discriminant, logistic regression, Support Vector Machine (SVM), random forest, boosted trees, gaussian SVM, and K-nearest Neighborhood (K-NN). In Table II, we reported the performance of the decision tree since it achieves the best performance. To predict risk, quantitative features were extracted from the heart rate signals of ICU patients with cardiovascular disease. Each signal is described in terms of 12 statistical and signal-based features. Finally, the authors of [42] used a CNN architecture to predict the patient risk of death. They employed the time series of readings of some vital signs (Heart Rate, Respiratory Rate, Systolic Blood Pressure, and Diastolic Blood Pressure) as training data. They compared their model with other architectures (i.e., recurrent neural network and logistic regression), showing how it succeeds in outperforming them. The papers show how models not based on deep neural networks manage, in some cases, to match or even outperform them. In general, the most recent works tend to prefer models based on 1-dimensional CNNs, whose usage is recently emerging in processing time series.

VIII. Conclusions

Access control mechanisms that decide whether to grant or deny access according to static and predefined permissions are inadequate in dynamic contexts such as healthcare systems. Here, adaptability is a key feature but it requires access control frameworks capable of governing decisions according to the risk level of patient conditions by taking into account context information characterizing the role of the access requester and the patient.

FRAMH is a risk-based authorization framework that exploits FL integrated with blockchain, as the main novelty, to calculate the level of health status risk which is crucial to properly tune access control decisions based on the trade-off between security and patient safety. The widespread diffusion of IoMT devices, which can continuously monitor individual conditions, has significantly contributed to increasing the availability of healthcare data with the possibility of improving the quality of medical services based on data evidence. However, several legal and regulatory principles prevent the possibility of sharing sensitive information among different medical institutions with consequent limits to leveraging traditional ML techniques. For this reason, we propose the adoption of FL in medical environments. This approach lays the basis of an original risk-based authorization middleware, namely FRAMH, that exploits FL to infer the health status of patients. We implement a prototype and demonstrate the effectiveness of FL in healthcare scenarios where hospitals may be driven by common goals that could be hindered by concerns related to sensitive information sharing. Experimental results show that the FL approach achieves performance comparable to the centralized approach by using the same training set. However, by reducing the training set to 75%, the federated configuration shows even better performance even though each client has been trained with a smaller data partition (33%) of the entire training set. Although the involved parties may have a restricted dataset, the collaborative model of FL allows them to achieve better performance than the adoption of a centralized ML approach with a larger dataset.

The results of this paper that are focused on risk-based dynamic access control can be extended to other smart healthcare applications, such as home healthcare, and to any online services requiring differential accesses based on dynamic risk-based conditions.

References

[1] Y. Yuehong, Y. Zeng, X. Chen, and Y. Fan, "The internet of things in healthcare: An overview," *Journal of Industrial Information Integration*, vol. 1, pp. 3–13, 2016.

[2] P. Gope and T. Hwang, "BSN-Care: A secure IoT-based modern healthcare system using body sensor network," *IEEE sensors journal*, vol. 16, no. 5, pp. 1368–1376, 2015.

[3] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.

[4] U. Ahmed, J. C.-W. Lin, and G. Srivastava, "Deep Hierarchical Attention Active Learning for Mental Disorder Unlabeled Data in AIoMT," *ACM Trans. Sen. Netw.*, feb 2022, just Accepted.

[5] L. Hong, M. Luo, R. Wang, P. Lu, W. Lu, and L. Lu, "Big data in health care: Applications and challenges," *Data and information management*, vol. 2, no. 3, pp. 175–197, 2018.

[6] EU GDPR, "Article 9 - Processing of special categories of personal data," 2016. [Online]. Available: https://www.privacy-regulation.eu/en/article-9-processing-of-special-categories-of-personal-data-GDPR.htm

[7] U. Ahmed, J. C.-W. Lin, and G. Srivastava, "Hyper-Graph Attention Based Federated Learning Method For Mental Health Detection," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2022.

[8] D. C. Nguyen et al., "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, 2021.
[9] H. F. Atlam, M. A. Azad, M. O. Alassafi, A. A. Alshdadi, and A. Alenezi, "Risk-based access control model: A systematic literature review," *Future Internet*, vol. 12, no. 6, p. 103, 2020.
[10] P. Bellavista, L. Foschini, and A. Mora, "Decentralised Learning in Federated Deployment Environments: A System-Level Survey," *ACM Comput. Surv.*, vol. 54, no. 1, feb 2021.
[11] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *AISTATS*, 2017.
[12] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous Federated Learning on Heterogeneous Devices: A Survey," *ArXiv*, vol. abs/2109.04269, 2021.
[13] M. Di Pierro, "What is the blockchain?" *Computing in Science & Engineering*, vol. 19, no. 5, pp. 92–95, 2017.
[14] B. K. Mohanta, S. S. Panda, and D. Jena, "An Overview of Smart Contract and Use Cases in Blockchain Technology," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2018, pp. 1–4.
[15] H. F. Atlam, M. A. Azad, and N. F. Fadhel, "Efficient NFS Model for Risk Estimation in a Risk-Based Access Control Model," *Sensors*, vol. 22, no. 5, p. 2005, 2022.
[16] D. Choi, D. Kim, and S. Park, "A framework for context sensitive risk-based access control in medical information systems," *Computational and mathematical methods in medicine*, vol. 2015, 2015.
[17] H. Abie and I. Balasingham, "Risk-based adaptive security for smart IoT in eHealth," in *Proceedings of the 7th International Conference on Body Area Networks*, 2012, pp. 269–275.
[18] I. Molloy, L. Dickens, C. Morisset, P.-C. Cheng, J. Lobo, and A. Russo, "Risk-based security decisions under uncertainty," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, 2012, pp. 157–168.
[19] K. Srivastava and N. Shekokar, "Machine learning based risk-adaptive access control system to identify genuineness of the requester," in *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*. Springer, 2020, pp. 129–143.
[20] J. Li, Y. Bai, and N. Zaman, "A fuzzy modeling approach for risk-based access control in eHealth cloud," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2013, pp. 17–23.
[21] P. Kuwik, T. Largi, M. York, D. Crump, D. Livingston, and J. C. Squire, "The smart medical refrigerator," *IEEE Potentials*, vol. 24, no. 1, pp. 42–45, 2005.
[22] D. L. McGuinness, F. Van Harmelen *et al.*, "OWL web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.
[23] T. J. Eggebraaten, J. W. Tenner, and J. C. Dubbels, "A health-care data model based on the HL7 Reference Information Model," *IBM Systems Journal*, vol. 46, no. 1, pp. 5–18, 2007.
[24] O. Standard, "Extensible Access Control Markup Language (XACML) version 3.0," *A:(22 January 2013). URl: http://docs. oasis-open. org/xacml/3.0/xacml-3.0-core-spec-os-en. html*, 2013.
[25] Z. Wang and Q. Hu, "Blockchain-based Federated Learning: A Comprehensive Survey," *ArXiv*, vol. abs/2110.02182, 2021.
[26] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the Tails: Yes, You Really Can Backdoor Federated Learning," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 16 070–16 084.
[27] Hangzhou Internet Court of the People's Republic of China, "Hangzhou Huatai Yimei Culture Media Co., Ltd. v. Shenzhen Daotong Technology Development Co." *Digital Evidence and Electronic Signature Law Review*, pp. 61–70, 12 2019.
[28] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 536–540.
[29] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A Friendly Federated Learning Research Framework," *ArXiv*, vol. abs/2007.14390, 2020.
[30] P. Deutsch, "DEFLATE Compressed Data Format Specification version 1.3," *RFC*, vol. 1951, pp. 1–17, 1996.
[31] S. Cai, S. Bileschi, and E. Nielsen, *Deep Learning with JavaScript: Neural networks in TensorFlow.js*. Manning, 2020.
[32] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2704–2713, 2018.
[33] I. Silva, G. B. Moody, D. J. Scott, L. A. Celi, and R. G. Mark, "Predicting in-hospital mortality of ICU patients: The PhysioNet/Computing in cardiology challenge 2012," *2012 Computing in Cardiology*, pp. 245–248, 2012.
[34] A. E. Johnson et al., "Patient specific predictions in the intensive care unit using a Bayesian ensemble," in *Computing in Cardiology (CinC), 2012*. IEEE, 2012, pp. 249–252.
[35] G. L. Aguiar, B. Krawczyk, and A. Cano, "A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework," *ArXiv*, vol. abs/2204.03719, 2022.
[36] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," in *Neural Networks: Tricks of the Trade*, 2012.
[37] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *ArXiv*, vol. abs/1803.08375, 2018.
[38] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6980, 2015.
[39] K. You, M. Long, J. Wang, and M. I. Jordan, "How Does Learning Rate Decay Help Modern Neural Networks," *arXiv: Learning*, 2019.
[40] S. B. Baker, W. Xiang, and I. M. Atkinson, "Continuous and automatic mortality risk prediction using vital signs in the intensive care unit: a hybrid neural network approach," *Scientific Reports*, vol. 10, 2020.
[41] R. Sadeghi, T. Banerjee, and W. L. Romine, "Early Hospital Mortality Prediction using Vital Signals," *Smart health*, vol. 9-10, pp. 265–274, 2018.
[42] L. Brand, A. Patel, I. Singh, and C. Brand, "Real Time Mortality Risk Prediction: A Convolutional Neural Network Approach," in *HEALTHINF*, 2018.

Carlo Mazzocca received his M.Sc. and B.Sc. degrees in Computer Engineering in 2018 and 2020, respectively, both from the University of Naples Federico II, Italy. He is currently a Ph.D. student in Computer Science and Engineering at the University of Bologna, Bologna, Italy. His research interests mainly include authentication and authorization solutions for the cloud-to-thing continuum.

Nicolò Romandini graduated from the University of Bologna, Italy, where he received MSc degree in computer science engineering. He is currently a Ph.D. student at the Department of Computer Science and Engineering at the University of Bologna.His research focuses mainly on blockchain, cybersecurity and machine learning, and how to integrate them into IoT domains.

Michele Colajanni received the master's degree in computer science from the University of Pisa and the PhD degree in computer science and engineering from the University of Roma. He is currently full professor of computer science and engineering at the University of Bologna. He was assistant professor at the University of Rome and full professor at the University of Modena since 2000. His research interests include cybersecurity, performance and prediction models, and big data on cloud systems.

Rebecca Montanari full professor at the University of Bologna since 2020 carries out her research in the area of information security and of the design/development of middleware solutions for the provision of services in mobile and IoT systems. Her research is currently focused on blockchain technologies to support various supply chains, including agrifood, manufacturing and fashion and on security systems for Industry 4.0.