IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 June 2022, accepted 27 June 2022, date of publication 6 July 2022, date of current version 21 July 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3188808*

METHODS

# F-CBR: An Architecture for Federated Case-Based Reasoning

**AMAR JAISWAL[1], KASSAYE YITBAREK YIGZAW[2], AND PINAR ÖZTÜRK[1]**
[1]Department of Computer Science, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway
[2]Norwegian Centre for E-health Research, 9019 Tromsø, Norway

Corresponding author: Amar Jaiswal (amar.jaiswal@ntnu.no)

**ABSTRACT** Case-based reasoning (CBR) is a problem-solving methodology in artificial intelligence that attempts to solve new problems using past experiences known as cases. Experiences collected in a single case base from an institution or geographical region are seldom sufficient to solve diverse problems, especially in rare situations. Additionally, many institutions do not promote peer-to-peer (p2p) communication or encourage data sharing through such networks to retain autonomy. The paper proposes a federated CBR (F-CBR) architecture to address these challenges. F-CBR enables solving new problems based on similar cases from multiple autonomous CBR systems without p2p communication. We also designed an algorithm to minimize (irrelevant or unsolicited) data sharing in an F-CBR system. We extend the F-CBR design to support institutions with organizational or geographical hierarchies. The F-CBR architecture was implemented and evaluated on two public datasets and a private real-world (non-specific musculoskeletal disorder patient) dataset. The findings demonstrate that the retrieval quality of F-CBR systems is comparable to or better than a single CBR system that persists all the cases on a centralized case base. F-CBR systems address data privacy by incorporating the data minimization principle. We foresee F-CBR as a viable real-world design that can aid in federating legacy CBR systems with minimal or no changes. The CBR systems used in this study are shared on GitHub to support reproducibility.

**INDEX TERMS** Case-based reasoning, data minimization, data privacy, data silos, decision support systems, federated architecture, federated case-based reasoning.

## I. INTRODUCTION

In recent years artificial intelligence (AI) has been recognized to be used in decision making in real life, even when the decisions may have a vital impact on people's life [1], [2]. This led to enormous interest in AI-based decision or decision support systems. With the surge of commonplace use of AI, the ethical concerns related to AI are triggered by the enforcement of regulations, such as the General Data Protection Regulation (GDPR) [3] in Europe. Explainability, transparency, and privacy are a few primary concerns in adapting AI to everyday use [4], [5].

A branch of AI, the so-called ''black-box'' approach in machine learning (ML), attempts to solve the privacy problem using federated learning. Each autonomous member black-box system in the federation learns a model using own data locally and shares the learned model to the federator. As such no raw data in the process is sent or shared at all. However, explainability and transparency problems pertinent to such black-box AI systems highlight the need for AI systems that align with GDPR's privacy and explainability regulations [3].

Case-based reasoning (CBR), a problem-solving methodology in AI that emerged from work in cognitive science [6], is considered to mimic the human reasoning process and possess explainability and transparency traits [7]. Because it explicitly assesses similarity between the new problem and the previously solved problems. CBR being transparent, easily understood by users, and explainable up to a large extent make it an easily adaptable AI methodology in various domains such as healthcare, law, planning, designing, process control, etc. [8]–[11].

The associate editor coordinating the review of this manuscript and approving it for publication was Kostas Kolomvatsos.

The premise of CBR is that *similar problems have similar solutions and are likely to recur in the future* [7]. Therefore, in CBR, a new problem is solved by retrieving and reusing similar previously-solved problem(s) referred to as experiences or cases [12]. A case typically consists of structured description of a problem situation in terms of relevant features and its solution. In CBR, cases are persisted in a case base, and a reasoning engine performs the reasoning task. The learning in a CBR system happens by adding new cases in its case base. CBR does not involve training a model as in main stream ML approaches; rather, it incorporates continuous learning by dynamically adding new cases to the case base as new problems are solved [13].

The competency of a CBR system relies primarily upon the solved cases it has collected or solved, and the ability to retrieve the relevant case(s) similar to a new problem. However, experiences collected in one institution, or geographical region are seldom sufficient to solve diverse problems, especially for rare conditions. Yet, vast amount of data collected across institutions remain in silos where they are collected [14]–[16].

In addition to the desire to have the autonomy over controlling their systems and data, institutions are often skeptical or unwilling to share their data for centralized persistence or participate in peer-to-peer[1] (p2p) networks [19] due to privacy concerns, especially in the healthcare domain [20]. Apart from the challenges due to fear of losing autonomy and jeopardizing data privacy, strict regulations such as the GDPR also started to play a key role in creating data silos.

We distinguish autonomous CBR systems that can benefit from experiences collected in other institutions into two types: one that support p2p communication and the second that does not. Comprehensive studies have been performed on the first type in the CBR research community, discussed in detail under section II. On the contrary, to the best of our knowledge the second type is neglected mostly due to some of the challenges mentioned above.

This paper proposes a simple yet effective system architecture, **federated case-based reasoning (F-CBR)**, that aims to improve decision support systems' overall competence through multiple CBR systems deployed in different institutions. The architecture is generic in the sense that it can be used for different application domains and is developed following the principles of federated system design [21]. The architecture would allow CBR systems to leverage the benefits of heterogeneous cases acquired locally by various CBR systems in numerous institutions and geographical locations while minimizing the amount of data to be shared. F-CBR mitigates the disadvantages of data silos without involving p2p communication between the member CBR systems, hence preserving the data owners' privacy, and minimizes the amount of data to be shared through a tailored algorithm while ensuring the autonomy of the member CBR systems. The key contributions of this paper are as follows:

1) A generic F-CBR methodology and architecture to federate autonomous CBR systems without p2p communication.
2) Algorithms that minimize the sharing of cases in the proposed F-CBR system using a two-stage federated retrieval.
3) The F-CBR methodology is further enhanced to federate institutions with organizational or geographical hierarchies.

Besides F-CBR being a generic design and independent of any domain, it can help homogeneous legacy CBR systems to participate in a federation with minimal to no adaptation of the existing system. We hypothesize that the proposed F-CBR approach would encourage institutions and data owners to participate in federated CBR research and applications.

The paper is structured as follows: section II provides the background of this research, followed by section III, which presents the proposed F-CBR architecture on two-stage retrieval algorithm, and hierarchical F-CBR design. Section IV is devoted to experiments, results, and discussions. The recommendations for deploying a F-CBR system is discussed in section V. Finally, the conclusion of this paper is presented in section VI.

## II. BACKGROUND

This section describes a number of concepts that relate to the notion of "federated" either as defining characteristic or a contrast because these key notions are at times mixed up or interchangeably used without highlighting possible differences.

### A. DE/CENTRALIZED, DISTRIBUTED, AND FEDERATED

The concepts of centralized, decentralized, and distributed systems were first defined in Baran's legacy paper in 1964 [22]. Centralized and decentralized system designs are mostly referred to in the context of network control. In a centralized approach, a single system controls the entire network, contrary to a decentralized approach.

Distributed and federated architectures are sometimes mistaken for one another and used interchangeably in certain situations. Nevertheless, they are two distinct architectural styles [23], [24]. A distributed design is adopted primarily to address the issues of parallelism, scalability, and availability. This term is often used at an enterprise level when the resources are distributed over multiple geographical locations [25]. In contrast, A federated design helps perform computation on data locally without needing p2p communication while retaining autonomy to a large extent [26].

The term federation is mainly used in the context of autonomy and privacy preservation [27]. Authors in [21] have

[1]We refer to p2p communication in the context of networks and decentralized systems with three defining properties: self-organization, symmetric communication, and distributed control [17], [18].

presented in-depth concepts, terminologies, and architectures for federated information systems.

### B. FEDERATED DATABASE, SEARCH, AND LEARNING

To understand and scope out how the various approaches associated with "federated" design differ from our proposed federated CBR, we discuss here the three most common approaches among them:

*Federated database* [28] is a collection of autonomous heterogeneous yet cooperating database systems to achieve interoperability. It focused primarily on the basic data operations in data management systems like insert, delete, search, and merge.

*Federated Search* [29] is linked to the field of web search engines, where multiple searches are executed on numerous local autonomous systems. The primary task of a federator in these systems is to coordinate and re-rank local search results. From an end-user point of view, there might not be a significant difference in whether a search is performed as a federated search or not.

*Federated Learning* [26], Contrary to federated database and search, members in federated learning share only the locally trained model parameters to a federator and not the raw data. Thus, user data permanently resides locally with its owners. Google's research coined the term federated learning. The primary obstacle federated learning addresses is real-world data privacy [30].

### C. RELATED CBR LITERATURE

Various terminologies linked to distributed CBR systems might seem similar to the proposed federated CBR design, which we argue are not. We attempt to clarify this in the remaining parts of this section.

*Collaborative CBR* [31]–[33] architectures and designs have been extensively explored and were confined to a specific domain such as route planning, recommender system, and elderly health assessment. The CBR systems used for collaboration leverage p2p communication capabilities to solve problems.

*Distributed CBR* [34] by Plaza and McGinty presented how distributed CBR strategies can improve the performance and maintainability of CBR systems with several additional benefits. However, these systems do not incorporate principle of federated design.

*Ensembled CBR* was proposed by Plaza and Ontanón [35]; they studied the "ensemble effect" where a collection of agents with uncorrelated case bases improves the accuracy of any individual system. Nevertheless, their design does not aim to avoid p2p communications.

*Multi-Agent System* (MAS) based CBR designs are well discussed in detail in the survey paper [36]. It is one of the most common techniques that has achieved promising results. MAS is a computerized system of multiple intelligent agents interacting with each other to solve a specific problem based on p2p communication.

**TABLE 1.** Literature comparison table.

| Related Literature | Research Sub-field | Federated Design | Application Domain Independent |
|---|---|---|---|
| Federated Database | database | Yes | Yes |
| Federated Search | information retrieval | Yes | Yes |
| Federated Learning | machine learning | Yes | Can be |
| Collaborative CBR | case-based reasoning | No | Can be |
| Distributed CBR | case-based reasoning | No | Can be |
| Ensembled CBR | case-based reasoning | No | No |
| MAS CBR | case-based reasoning | No | No |
| **F-CBR** | **case-based reasoning** | **Yes** | **Yes** |

Table 1 shows in a structured way the differences between our proposed F-CBR (in bold) against the relevant reviewed literature to the best of our knowledge. Apart from the federated collaboration, other main pillar in the approach is the degree of application domain independence.

### D. CONVENTIONAL AND FEDERATED CBR CYCLES

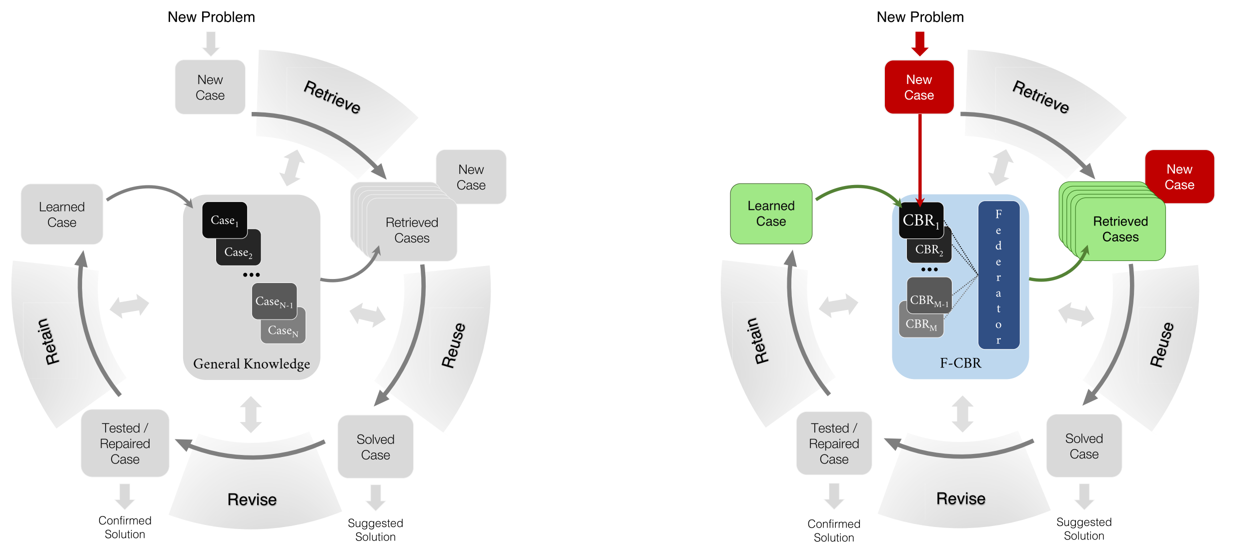
We refer to a conventional CBR as a CBR system which has a single case base, thus all the experiences of that system is centrally persisted. The critical difference between the conventional CBR (Fig. 1a) and the proposed federated CBR (Fig. 1b) design at an abstract level is at the center block of the CBR cycle, where the "F-CBR" replaces the "General Knowledge" block. The "General Knowledge" block is depicted as a case base persisting experiences/cases. Whereas in Fig. 1b, the F-CBR block is a federation of multiple autonomous CBR systems, thus the cases are distributed. The primary difference in the process is how the two CBR variants (conventional and federated) serve a retrieval request. Similar cases are retrieved from a case base in a conventional CBR, whereas similar cases are retrieved from a federation of multiple CBR systems in federated CBR. However, from an end-user's perspective, whether the cases retrieved from a conventional, or a federated system are not different.

The learning in both designs happens in the retain phase. A learned case, in a conventional CBR, is persisted in the case bases of a CBR system accessible by the user. In F-CBR, a learned case is persisted in the case base of a local CBR system, and the federator would not play any role in this, as shown in Fig. 1b. Thus a member has complete autonomy over the reuse, revise, and retain phases in an F-CBR process cycle.

Our approach may possess some similarities with the federated search approach. However, in the proposed federated CBR, not all the locally retrieved results are shared with a federator or an end-user, contrary to the federated search. F-CBR minimizes the exposure of more than required cases to a federation, with the effect to improve data privacy by the "data minimization" (Article 5(1)(c) of the GDPR). Through this paper, we propose a generic architecture where legacy systems also could easily be ported into a federation.

## III. FEDERATED CASE-BASED REASONING (F-CBR)

This section describes requirements and assumptions for F-CBR, the proposed federated CBR architecture, the proposed two-stage retrieval algorithm supported with an exemplified retrieval, and the proposed hierarchical F-CBR design.




(a) Conventional CBR cycle adapted from [13]. All the learned cases ($Case_1$ to $Case_N$) are stored in a single case base, as shown in the *General Knowledge* block.

(b) F-CBR cycle with $M$ federated members ($CBR_1$ to $CBR_M$). The learned cases are stored locally in a member's case base.

**FIGURE 1. Comparison of CBR process cycle between the conventional and federated CBR approach.**

## A. REQUIREMENTS AND ASSUMPTIONS FOR F-CBR

Although institutions may make better decisions when they benefit from each other's experiences, institutions owning autonomous CBR systems are often averse to sharing their cases for centralized consolidation and persistence, and participating in p2p communication networks. The aimed F-CBR architecture should address these challenges, *requirement 1*.

The autonomous member CBR systems are assumed not to rely on p2p communication among themselves, *requirement 2*. Since the reasoning process in CBR relies on the retrieval of cases, the architecture should ensure that not more than necessary cases are shared with an end-user or the federation, *requirement 3*. The identity of subjects and data owners are potentially sensitive information and thus should not be disclosed to a federated member or an end-user, *requirement 4*. The member CBR systems have local case bases, and they can use different similarity measures, *requirement 5*. However, they must adhere to a standard range of similarity scores, recommended to be between 0 and 1, *requirement 6*. The federator (controller) is assumed not to possess learning capabilities from the cases or its owners, *requirement 7*. We assume that end-to-end encryption is in place for request and response exchanges over a network for real-world applications. We also rely on the respective institutions for data masking, de-identification, anonymization, and adequate security and privacy measures.

## B. PROPOSED F-CBR ARCHITECTURE

We describe *federated case-based reasoning* (F-CBR) as a method that enables solving new problems based on similar past cases retrieved from multiple autonomous CBR systems without peer-to-peer communication and centralized case

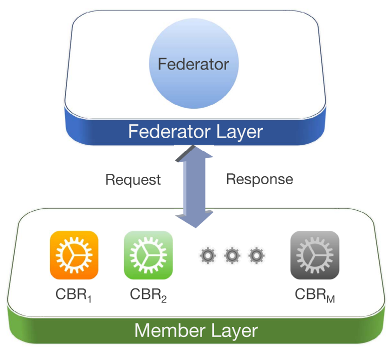



**FIGURE 2. Two-layer F-CBR architecture.**

persistence. The reasoning in F-CBR is performed by a loose federation of participating local CBR systems, known as **members** and is coordinated by a central server, the **federator**. The F-CBR architecture is inspired by the paper on design patterns for federated architectures [23].

Fig. 2 presents a two-layered generic architecture for an F-CBR system. The *federator layer* is not available for direct access by an end-user, while only the members from the *member layer* can access it. The member layer holds several autonomous CBR systems with their local case bases. It is critical to note that the members do not have p2p communication capability; thus, they would be ignorant of each other's identity and capabilities. Additionally, they expect the federation to limit data sharing to the greatest degree possible.

Therefor, we propose two variants of F-CBR for data minimization, one that performs data minimization at the federator layer, and the other performs additional data minimization at the member layer. We start with retrieval process description in an F-CBR system, using the first variant due to its simple yet effective approach.

An end-user interacts with the F-CBR system by submitting a query (new problem) to their local CBR system. The member who receives a query from an end-user will be referred to as *initiator* in the process until the end-user receives a response to the query.

The retrieval process in an F-CBR system starts upon a query being submitted to a local CBR system to retrieve the desired $k$ number of most similar cases, where $k > 0$. The initiator then delegates the end-user's request to the federator. The federator further delegates the retrieval query to all its members, including the initiator. The federation has $M$ members ($CBR_1$ to $CBR_M$) at the member layer, as shown in the Fig. 2. Upon receival of request from the federator, each member performs a retrieval in their local case bases and returns (at most) the $k$ most similar cases to the federator. Consequently, the federator receives a maximum of $M \times k$ cases if every member has at least $k$ number of cases in its case base. Else, suppose all members have less than $k$ cases in their case base, then the maximum number of cases a federator can receive is calculated using the formula $\sum_{i=1}^{M} |m_i|$, where $|m_i|$ is the total number of cases in the case base of a member $m_i$. Based on similarity scores of all received cases, the federator filters out cases other than the top $k$ similar cases from the received $M \times k$ cases. The federator then re-orders the $k$ cases in descending order of their similarity scores and returns them to the initiator as a response, concluding the F-CBR retrieval process.

The filtering of more than $k$ cases (redundant) by the federator directly relates to the *data minimization* principle of GDPR for privacy preservation. However, the minimization did not happen in the member but in the federator layer. Thus, $(M - 1) \times k$ unnecessary cases were shared (exposed) to the federator which should be avoided to the extent possible for further enhancing the data privacy. This case redundancy can be mitigated at the member layer itself, by incorporating the proposed two-stage federated retrieval, as discussed in detail below.

### C. TWO-STAGE FEDERATED RETRIEVAL

Fig. 3 depicts the two-stage retrieval process visually using sub-figures in chronological order. Further, the two-stage retrieval process (Algorithm 1), runs at the federator and depends on two other algorithms to be executed in each member locally, Algorithm 2 and Algorithm 3.

Stage 1 is about similarity threshold computation. After receiving a request from an initiator, the federator sends the query to all the members which in parallel find the $k$ most similar cases, (Fig. 3c). Each member runs the received query locally to retrieve the desired $k$ most similar cases from its case base and stores them ephemerally (short-lived) for a query *id*, (see Algorithm 2 for more details). After that, each member returns only the similarity scores of the $k$ most similar cases to the federator in descending order, (Fig. 3d). Note that the cases themselves are not shared. The federator might receive similarity scores from the members asynchronously. After receiving the scores from all the members, the federator finds the $k^{th}$ highest similarity

**Algorithm 1** Federated Retrieval

(*Executed by the federator*) - Performs Two-Stage Federated Retrieval.
*id*: A Unique Identifier for a Query.
*q*: The Query, That Is, the Problem to Be Solved.
*k*: The Desired Number of Similar Cases.
*M*: Number of Federated Members.
*retrievedCases*: List of $k$ Most Similar Cases, Retrieved From All the $M$ Members

```
FederatedRetrieval(id, q, k):
    // size of list_of_lists = |M|, total members
    list_of_lists[ ]
    for each member m ∈ M in parallel do
        sim_list ← SimilarityScores(id, q, k)  list_of_lists ← append(sim_list)
    end
    // threshold = k^th largest similarity score
    threshold ← getThreshold(list_of_lists, k)
    for each member m ∈ M in parallel do
        cases ← EphemeralCases(id, threshold)  retrievedCases ← append(cases)
    end
    sortedCases ← reverseSort(retrievedCases)  sortedCases ← (select top k cases) retrievedCases ← anonymize(sortedCases)
return retrievedCases
```

**Algorithm 2** Member Similarity Scores

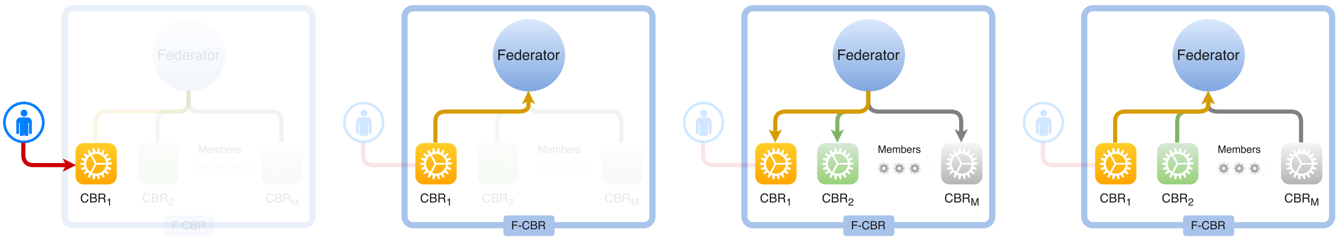
(*Executed by all the member*)

```
SimilarityScores(id, query, k):
    cases ← cbrRetrieval(query, k)
    saveEphemeralCases(id, cases)    // save for id
    scores ← getSimScores(cases, k)
return scores    // descending order
```

**Algorithm 3** Member Ephemeral Cases

(*Executed by all the members*)

```
EphemeralCases(id, threshold):
    cases ← fetchEphemeralCases(id)
    adequateCases ← discard(cases, threshold)
return adequateCases    // similarity ≥ threshold
```




(a) User Query (b) Request to Federator (c) Similarity Score Request (d) Similarity Score Response

(e) Ephemeral Case Request (f) Ephemeral Case Response (g) Federator Response (h) Query Response

**FIGURE 3. Two-stage retrieval in F-CBR. Non-activated components are faded out in each step. The sub-figures depict the retrieval process in chronological order from (a) to (h).**

score among the scores delivered by all the members, which will later be used as the similarity *threshold*. This concludes stage 1.

The reason for choosing the $k^{th}$ similarity score as the threshold is based on the user's request for $k$ most similar cases to solve a query (problem). A similarity score is not merely a number but a representative of an ephemeral case held locally by a member. Thus, when these are arranged in descending order, the $k^{th}$ similarity score would serves as a filter for discarding the ephemeral cases whose similarity scores are below it by each member. The threshold will be used in stage 2.

Stage 2 is for retrieving only adequate ephemeral cases from a member. Once the threshold is computed, the federator sends the query id and the threshold value to all the members once again for selection of the adequate similar cases (Fig. 3e). The members serve this request as described in Algorithm 3. Consequently, the federator receives only cases with similarity scores above or equal to the threshold. The members discard the rest of the ephemeral cases for the given query *id* (Fig. 3f). The federator further reorders the retrieved cases in descending order of their similarity scores, selects (at most) the top $k$ cases to share with the initiator, and discards the rest (Fig. 3g). This concludes stage 2. Finally, the user receives the desired $k$ most similar cases from the F-CBR system (Fig. 3h), which concludes the overall retrieval process.

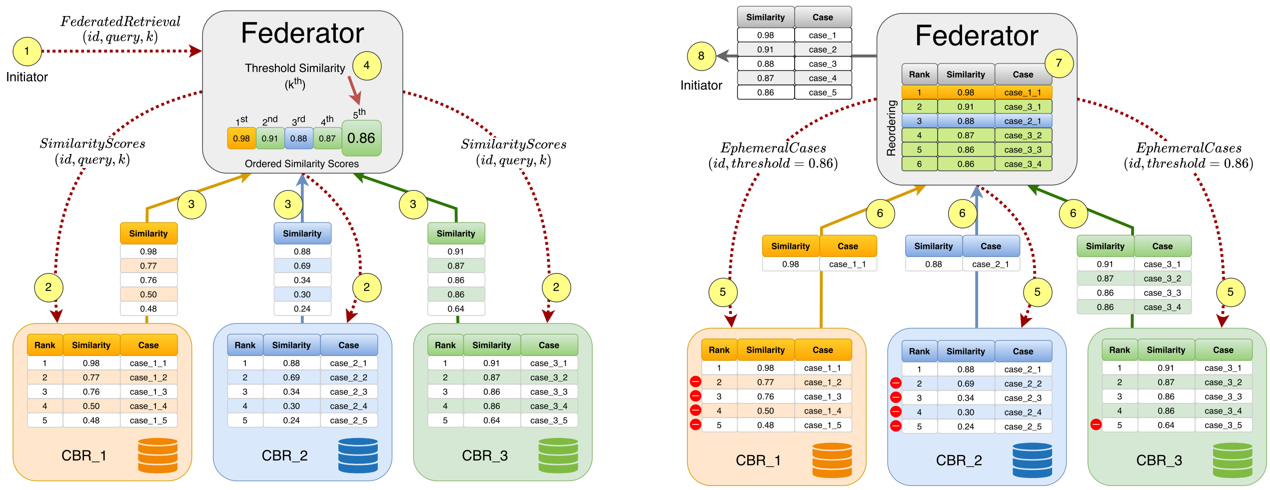
The two-stage federator retrieval process minimizes the number of cases that need to be shared with the federator for a query at the member layer itself. However, the extent of data minimization is essentially dependent on the similarity measures used by a member for a retrieval. For instance, if all locally retrieved cases from all members have identical similarity scores, no data minimization will occur at any member. To illustrate this, in a worst-case scenario, assume that all the top $k$ similar cases in all the members have identical similarity score of 0.9 for a query. Thus, each member will send a list of size $k$ of which all elements are equal to 0.9 in stage 1. The threshold similarity score will then be 0.9. Since all ephemeral case similarity scores are 0.9, each member in stage 2 delivers $k$ cases to the federator without discarding any case. Hence, we emphasize that similarity measures play a vital role in data minimization in an F-CBR system and must be diligently designed. Our paper [37] presents a method for creating multiple similarity measures, including feature selection, based on a data-driven approach. The paper also showcases the effects of numerous similarity measures on a few public datasets. It emphasizes building a baseline similarity measure for the incremental assessment of the quality of retrievals.

### D. EXEMPLIFIED TWO-STAGE F-CBR RETRIEVAL

In Fig. 4, we illustrate the two-stage federated retrieval processes using an exemplified F-CBR system with 3 federated members: *CBR_1*, *CBR_2*, and *CBR_3*.

The retrieval process in an F-CBR commences when a user sends a request to its local CBR system, say *CBR_1*. For this illustration, we assume that $k$ is equal to 5. *CBR_1* delegates the request to the Federator with a unique request identifier, as shown with a yellow circle numbered 1 in Fig. 4a. The Federator forwards the request to all the members, in parallel, as shown with yellow circles numbered 2 in Fig. 4a. Each member retrieves 5 most similar cases from its respective case base and ephemerally stores them with the unique request identifier ($id$) as a reference. The vital point to note here is that the members do not share the retrieved similar cases with the Federator at this stage, merely sends the similarity scores of only 5 most similar cases, sorted in descending order, as shown with yellow circles 3 in Fig. 4a.

Thus, the Federator receives a total of 15 similarity scores. Sharing the similarity scores does not threaten data privacy as no case data is shared with the Federator at this stage. After that, the Federator finds the $5^{th}$ highest similarity score, which




(a) Federator retrieval process stage 1.

(b) Federator retrieval process stage 2.

**FIGURE 4.** **Exemplified federator's two-stage retrieval process. Yellow circles with numbers are cues for the chronological order of the retrieval process. The red circles in (b) indicate ephemeral cases which are not shared with the federator in stage 2.**

acts as a threshold for filtering the locally stored ephemeral cases by all the members. In our example the $5^{th}$ highest similarity score is 0.86 (yellow circle 4).

In **stage 2**, the Federator sends a second request, in parallel, to all the members with the query *id* and the threshold similarity score of 0.86, (yellow circles 5 in Fig. 4b). Each member retrieves ephemerally stored cases for the respective *id* and discards the ephemeral cases whose similarity scores are below the threshold value. The discarded cases are shown in red color in Fig. 4b.

As a result, *CBR*_1 and *CBR*_2 share only one case each (*case*_1_1, and *case*_2_1 respectively) but *CBR*_3 shares four cases (*case*_3_1, *case*_3_2, *case*_3_3, *case*_3_4) with the Federator, shown with yellow circles 6 in Fig. 4b. In this way, the F-CBR addresses data privacy by reducing the exposure of more than adequate cases to the federation. The case labeling is just for an illustration and is based on the pattern: "case_<member>_<rank>," that is, if a case is from *CBR*_2 and is at rank 3 in the retrieval list, then its label is *case*_2_3. After receiving the similar cases from all members, the Federator rearranges all the cases in descending order of their similarity scores, shown by the yellow circle 7 in Fig. 4b. The Federator then picks only the top $k = 5$ cases and discards the rest, which in our example is *case*_3_4. Selecting only $k$ cases preserves the data privacy by not exposing redundant cases to the end-user. Also, the Federator erases any further information related to identifying a member in the federation from the final $k = 5$ cases, such as member identifiers: *case*_1_1, *case*_3_1, *case*_2_1, *case*_3_2, and *case*_3_3. This de-identification ensures that the end-user would not identify a member from a retrieved case directly. As a result, de-identification would preserve the identity of members in the federation.

Finally, the initiator will receive $k = 5$ most similar cases, as marked by the yellow circle 8 in Fig. 4b; this ends stage 2 of the Federator. Subsequently, the initiator member system could share the federated retrieval result with the end-user or perform post-processing such as adaptation.

### E. PROPOSED HIERARCHICAL F-CBR

In this paper, we have discussed single-layer federating CBR systems, where there is a single federator with multiple local CBR systems. This can be seen as a limitation for implementing a federated CBR design for hierarchically structured organizations or data owners. Therefore, we make a further attempt to propose a multilayered federation through this paper, where multiple federators can be arranged in a hierarchical structure to perform as an F-CBR system, as shown in Fig. 5. The design of such a system can be understood as follows: the root federator will be responsible for federating sub-federators under it. This can be viewed as a network of federators with intermediate nodes as sub-federators and leaf nodes as local CBR systems. All the federators and local CBR systems immediately below a parent federator are referred to as its members. The remaining design is similar to a single federator F-CBR system.

The retrieval process in a hierarchical F-CBR is as follows: a query is posed at a local CBR system which delegates it upwards to its parent node, a federator. If this federator is not the root federator, the query is further delegated to the parent of the current federator till it reaches the root federator. Once the root federator receives the query, it delegates it to all its sub-federators. This delegation continues further in a downward direction until a leaf node, a local CBR system, has reached where every local CBR system performs a local retrieval and persists the retrieval result ephemerally. After that, the members send only the similarity scores to their immediate federator. These immediate federators find their respective thresholds and perform the second stage retrieval as shown in Algorithm 1. Once a federator has retrieved the ephemeral cases, it sends the similarity scores of these cases

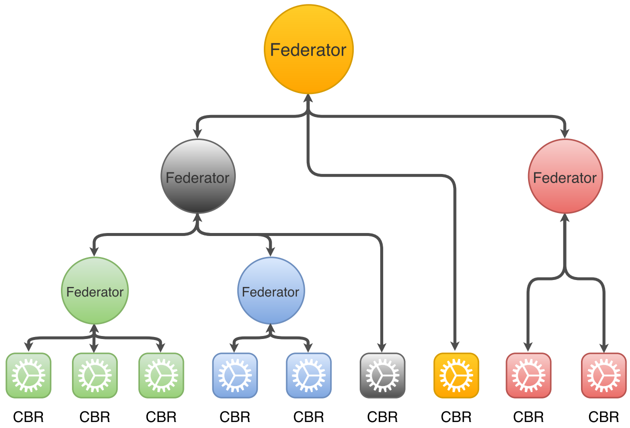



**FIGURE 5. Hierarchical F-CBR design. A member CBR is assigned the color of its federator to improve clarity.**

to its parent instead of the initiator, as in the case of single-layer federator.

All the child federators must accomplish the stage 2 retrieval process before a parent initiates its second stage retrieval. This process continues until the root federator finishes its two-stage retrieval process. Once the root federator has completed stage 2, it passes the result of hierarchical federated retrieval to the member who sent the request to the root federator. This process continued until a leaf member, the local CBR system, gets its retrieved results. The retrieval request might also come from a local CBR system (the third CBR from the right in Fig. 5) directly to a root federator.

The retrieval process in hierarchical F-CBR is similar to the single-layer federation but cascaded. The only difference is that sub-federators, once they compute their final result, only send the similarity scores and ephemerally persist their two-stage retrieval results for their parent's second stage retrieval. Sub-federators are unaware whether the retrieved results came from a sub-sub-federator or a member CBR system. From a federator's perspective, it abstracts the underneath hierarchical structure of the entire federation. Once the whole retrieval cycle is complete, no federator holds any cases, not even ephemerally. The cases always reside in the local case bases of their respective CBR systems or leaf nodes. It is critical to observe that a federator may become a leaf node if all its members have left; therefore, a federator should be capable of handling such scenarios.

## IV. EXPERIMENTS, RESULTS, AND DISCUSSION

The proposed F-CBR method has been implemented and validated for a prototypical F-CBR system, which was applied to two public Lymphography [38] and Zoo [39] datasets. These datasets are meant for the classification task. We have further applied the F-CBR approach on a real-world patient dataset, musculoskeletal disorder (MSD) patient dataset, described in details in paper [40], and the task performed on this dataset is patient similarity search [41], [42] using CBR system. More details about these datasets are provided in their respective experiment sections.

To classify a test sample we used the top three ($k = 3$) most similar cases' target labels retrieved by an F-CBR system. We chose $k$ based on the dataset size and the target label distribution in the datasets. Further, we consider a correct (true) classification if the top $k$ retrieved cases have the same class as the test sample. Else, the classification is considered incorrect (false), instead of selecting a majority class label as the predicted class for a test sample. This was so done to have a strict criterion for evaluating the F-CBR systems in this paper.

The primary goal of the experiments is to demonstrate that the retrieval quality of an F-CBR system is same compared to a conventional CBR system where all the cases from all the members are persisted in a centralized case base.

### A. COMMON EXPERIMENTAL SETUP

For each dataset, multiple CBR systems with local case bases were built to implement an F-CBR system where the case bases of these CBR systems were disjoint. This design choice helps to capture the "worst case" scenario where each member CBR system has unique experiences (cases) in their case base that other members have never gained (learned).

We also built a conventional CBR system for each dataset with a centralized case base (a case base containing all the cases built in the respective datasets).[2]

The similarity measures used in this paper are based on the local-global principle [44]. A global similarity function is the weighted sum of all the local feature level similarity scores. The similarity function used in this paper is shown by equation 1. The $sim(q, c)$ produces a similarity score for a query $q$ against a case $c$. For an $i^{th}$ feature, a local similarity function is defined as $sim_i(q_i, c_i)$, where $q_i$ and $c_i$ are the feature values of a query ($q$) and a case ($c$), respectively. The similarity score range is $[0, 1]$; the reader can refer to our previous paper [37] for more details about the similarity measures. To build a "baseline" similarity measure for each dataset, all the feature weights $w_i$ are assigned a value equal to 1.

$$sim(q, c) = \frac{\sum_{i=1}^{n} w_i \times sim_i(q_i, c_i)}{\sum_{i=1}^{n} w_i} \tag{1}$$

where,

$$\begin{aligned} i &: i^{th}\ feature \\ n &: total\ number\ of\ features \\ w_i &: weight\ for\ i^{th}\ feature \end{aligned}$$

[2]Tools used for creating the CBR system are myCBR-sdk, myCBR-rest, and myCBR-workbench, whose details can be found in our workshop paper [43]. The built CBR systems are shared, except the MSD dataset, in a public repository for reproducibility and benchmarking purposes under the GitHub link: https://github.com/amardj/cbr-benchmark-projects/tree/master/f-cbr

The CBR systems were built on microservices architecture [45], and run as an independent process to emulate member CBR systems running in silos with their local case bases. None of the CBR systems was aware of any other CBR system, and hence no peer-to-peer communication between the members.

### B. EXPERIMENT WITH LYMPHOGRAPHY DATASET

The Lymphography dataset is meant for multi-class classification tasks with four classes, fibrosis, malign lymph, metastases, and normal. A total of 5 CBR systems were built where one is a conventional CBR (C-CBR) system containing all 148 cases built form the dataset in a centralized case base. The other four CBR systems were designed to be used for federation with similarity measures identical to the one used in the C-CBR system. The 4 member CBR systems (CBR-1, CBR-2, CBR-3, and CBR-4) contain exclusively 4, 61, 81, and 2 patient cases, respectively. Each CBR system contains cases of one class only. This was done to represent the heterogeneity of different member CBR systems in terms of the experiences/cases. The retrieval results presented in Table 2 are based on "baseline" similarity measures. The table results from a test query with the target class as fibrosis, and the query was executed in F-CBR and C-CBR systems. The retrieval results from F-CBR, and C-CBR appeared to be the same regarding similarity scores and ranking of the retrieved cases. For the two-stage F-CBR system, the threshold similarity score for $k = 5$ was 0.667, rounded to three decimal places. At first glance, the target class, at rank 4 in the table, are different, that is, fibrosis versus malign_lymph, in F-CBR and C-CBR, respectively. However, with careful observation, we found out that it happened because the federator delegates the retrieval requests to its members in parallel, and reorders the cases received from them. Hence, the order of the cases with identical similarity scores may differ depending on when a member delivers its results.

The retrieved cases from the members for the F-CBR system were as follows: $1^{st}$ and $2^{nd}$ rank cases were from *CBR-1*, $3^{rd}$ rank case was from *CBR-3*, $4^{th}$ rank case was from *CBR-1*, and the $5^{th}$ rank case was from *CBR-2*.

We have performed the query for all the 148 cases in F-CBR and C-CBR systems and have found that the retrieval quality is identical for both the system for a given query. The experiments demonstrated that the F-CBR system is invariant to the distribution of cases among the member CBR systems.

A detailed retrieval comparison based on similarity scores is presented in the form of *self-similarity matrix* (SSM) plots as shown in Fig. 6. The SSM has been well explored in music for better comparison and visualization [46], [47]. We describe a SSM, in CBR, as a square matrix of similarity scores, where similarity scores are computed for all the cases against each other in a CBR system, that is, every case is used as a query to retrieve similarity scores of all the cases in CBR system. The row represents the query cases and the column represents the retrieved cases.

SSM for C-CBR and F-CBR is computed for all the 148 cases of the Lymphography dataset. In C-CBR, all the cases are retrieved from a single case base, while in F-CBR, the cases are retrieved from the 4 autonomous CBR systems' case bases. The labels are shown for every 3rd case on the rows and columns without dropping any query and case similarity scores for visualization. The color bar corresponds to the similarity scores ranging between [0, 1].

The SSM plot depicts that the F-CBR system identically match the C-CBR system's retrieval quality. However, we observe that a few case labels in the F-CBR plot (Fig. 6b) are not in the same order as in the C-CBR plot (Fig. 6a). This is due to the fact that many cases have resulted in identical similarity scores based on the similarity measure being used, the baseline. Additionally, retrievals in F-CBR from its members happen in parallel which also affects the retrieval order of the cases with identical similarity scores, as discussed earlier in this section.

Hence, an F-CBR design mitigates the need for centralized data persistence. However, the key aspect we demonstrate is that the member CBR systems being heterogeneous in their populated case bases preserve the patient data and member identity. Thus we have managed to achieve the performance level of a centralized CBR system in terms of retrieving cases using an F-CBR design without peer-to-peer communication between members.

### C. EXPERIMENT WITH ZOO DATASET

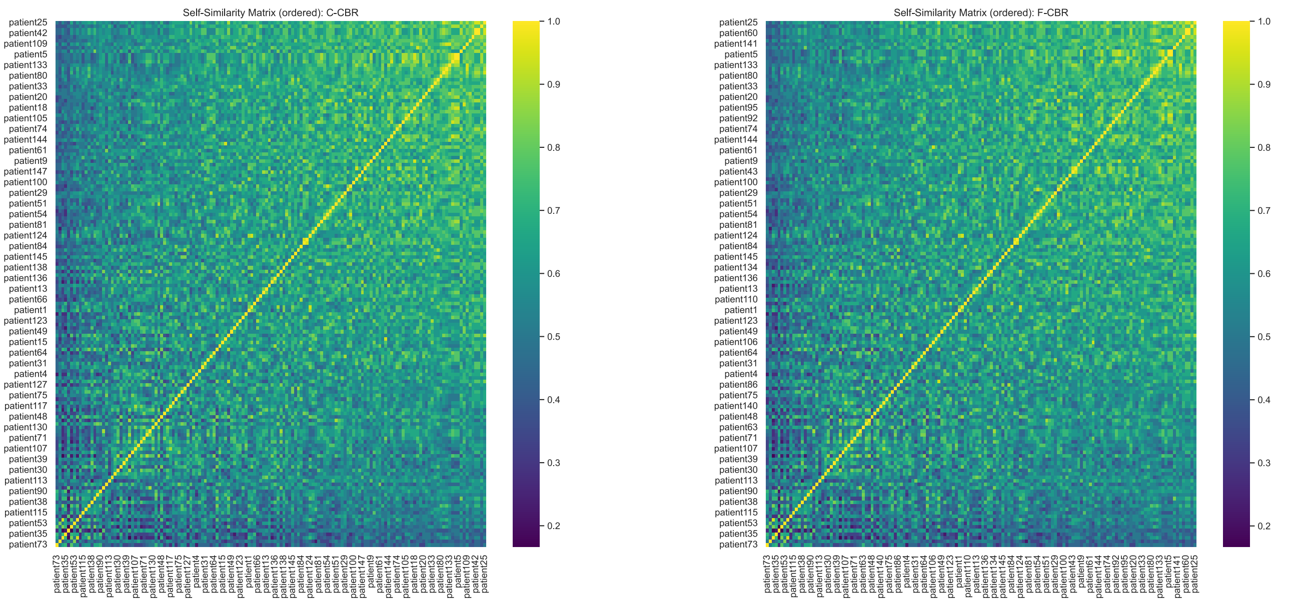
The zoo dataset is also meant for multi-class classification tasks with 7 classes. A total of 8 CBR systems were built where one is a conventional CBR (C-CBR) system containing all 101 cases built form the dataset in a centralized case base. The other 7 CBR systems were designed to be used for federation with similarity measures identical to the one used in the C-CBR system. The 7 member CBR systems (CBR-1, CBR-2, CBR-3, CBR-4, CBR-5, CBR-6, and CBR-7) contain exclusively 4, 20, 13, 8, 10, 41, and 5 animal cases as amphibians, birds, fish, insect, invertebrate, mammal, and reptile species, respectively. This was done to represent the heterogeneity of different member CBR systems in terms of the experiences/cases similar to the Lymphography experiment previously discussed in this paper.

The retrieval results based on "baseline" similarity measures from C-CBR and F-CBR were identical with few cases having identical similarity scores were being ordered differently. The results were in accordance with our expectation of an F-CBR system's retrieval results are indifferent compared to a C-CBR system, as observed in the Lymphography experiment.

The experiment described below for the zoo data set is to demonstrate that for some domains F-CBR with member specific similarity measures can outperform a conventional CBR system in terms of the retrieval quality. To demonstrate this, we developed a new similarity measure, which we term as a **regional similarity**. The proposed similarity measure is

**TABLE 2. Lymph dataset, retrieval results from F-CBR, C-CBR, member CBR-1, member CBR-2, member CBR-3, and member CBR-4. F-CBR is a federated CBR with CBR-[1 to 4] as members. C-CBR is a conventional CBR with all the cases in a single case base.**

| | F-CBR | | C-CBR | | CBR-1 | | CBR-2 | | CBR-3 | | CBR-4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rank | Similarity | Target | Similarity | Target | Similarity | Target | Similarity | Target | Similarity | Target | Similarity | Target |
| 1 | 1.000 | fibrosis | 1.000 | fibrosis | 1.000 | fibrosis | 0.667 | malign_lymph | 0.722 | metastases | 0.500 | normal |
| 2 | 0.833 | fibrosis | 0.833 | fibrosis | 0.833 | fibrosis | 0.667 | malign_lymph | 0.667 | metastases | 0.444 | normal |
| 3 | 0.722 | metastases | 0.722 | metastases | 0.667 | fibrosis | 0.667 | malign_lymph | 0.611 | metastases | n.a. | n.a. |
| **4** | **0.667** | fibrosis | **0.667** | malign_lymph | 0.556 | fibrosis | 0.667 | malign_lymph | 0.611 | metastases | n.a. | n.a. |
| 5 | 0.667 | malign_lymph | 0.667 | malign_lymph | n.a. | n.a. | 0.667 | malign_lymph | 0.611 | metastases | n.a. | n.a. |




(a) C-CBR (b) F-CBR

**FIGURE 6. Lymphography self-similarity matrix plots for C-CBR and F-CBR.**

also based on the local-global similarity function as shown in equation 1. However, the feature selection was based on *candidate-elimination* algorithm [48] for every CBR system, where feature was selected to participate in the similarity measure if its value did not change for a given target class label. For instance, in the CBR-2, which contains only bird cases, the feature "feathers" and "fins" are among the selected features because all the cases have identical values as "Yes" and "No", respectively. The selected features contributing to the similarity measure were assigned with identical weights which is equal to 1. It is to be noted that the regional similarity measure in one CBR system could be different from the other CBR systems.[3]

Fig. 7 shows the obtained test results in a confusion matrix for both the baseline (Fig. 7a) and regional (Fig. 7b) similarity measures. The numbers in the cells correspond to the classification of a test sample. For instance, in the F-CBR system based on baseline similarity measure (Fig. 7a), one test query was miss-classified as a "bird" by the system, whereas, its true class was "reptile". Therefore, the intersection of "reptile" row and "bird" column of the matrix is populated by value 1. The major diagonal of the confusion matrix represents the count of correct classifications for the experiment. The numbers appearing other than the major diagonal are the miss-classified samples. The F-CBR systems based on the baseline and regional similarity measures demonstrated a total of 4 and 0 miss-classifications, respectively. Using the F-CBR design, regional similarity, and members representative for an individual species class, we have achieved 100% classification accuracy for a test set of size 101. The test set contained all the 101 animal samples from the zoo dataset. The classification and evaluation criteria are precisely the same as described in the experimental setup section IV-A.

The proposed regional similarity measures showcase promising results, but we do not claim that this to be applicable for all data sets. The proposed similarity measure might not be universally applicable since a CBR system is developed for a specific domain. However, if the domain and data support the proposed regional similarity measure, promising results might be achieved. We also clarify that the regional similarity approach might not be feasible for a conventional CBR design with a single case base.

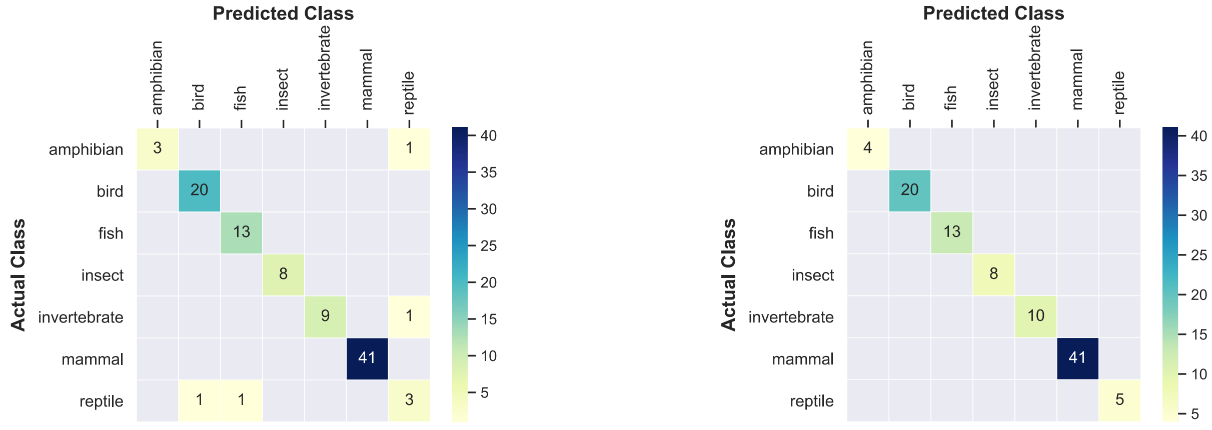
### *D. EXPERIMENT WITH MSD DATASET*

The MSD patients' dataset was collected under a research project SupportPrim,[4] which is a spin-off project from FYSIOPRIM[5] research.

[3]The exact similarity measures used can be obtained from myCBR project files with the help of myCBR-workbench tool, mentioned in the experiment setup section of this paper.

[4]https://www.ntnu.edu/supportprim

[5]https://www.med.uio.no/helsam/english/research/groups/fysioprim/index.html




(a) Baseline Similarity Measure (b) Regional Similarity Measure

**FIGURE 7. Confusion matrix from classification results obtained using two F-CBR systems (a) and (b) with baseline and regional similarity measures, respectively, for the zoo dataset. The color bars provides a reference to the number of cases shown in each cell of the matrix.**

Prototypical-experiments done on the MSD dataset for the paper includes a subset of 50 patient records from five distinct clinics called *support2*, *support5*, *support9*, *support15*, and *support70* including 2, 25, 12, 1, and 10 patient cases, respectively. A total of 6 CBR systems were developed, one of which is a conventional CBR (C-CBR) system with a centralized case base that contains all 50 cases. Other 5 CBR systems were built for federation using the same baseline similarity measures as the C-CBR system. The 5 CBR systems (CBR-02, CBR-05, CBR-09, CBR-15, and CBR-70) each include solely 2, 25, 12, 1, and 10 cases. Each member's CBR system corresponds to a clinic, simulating a real-world scenario in which members have varying degrees of MSD competence. Using two of our previous papers, [49] for developing case representation and [37], for designing the baseline similarity measures. Domain experts from FYSIOPRIM and SupportPrim assisted us in the development of baseline similarity measures.

As seen in Fig. 8, the retrieval outcomes based on "baseline" similarity measures from C-CBR and F-CBR were equal. The graph displays an average of the top 3 retrieved cases' similarity scores against all the 50 patient cases queried in both systems. These average similarity scores are represented by the y-axis for each patient. The patient IDs along the x-axis are arranged based on the highest to the lowest average similarity scores.

To determine how much data minimization can be accomplished with the present experimental configuration, we queried the F-CBR system for all 50 cases. Each query was executed to retrieve the top $k$ most similar cases, with the value of $k$ varying from 1 to 10 to simulate the sharing of unsolicited/redundant cases with the federator. The results of these experiments were encouraging since just two out of 500 ($50 \times 10$) queries with varied $k$ had exposed unsolicited cases (only 2 cases) to the federator.

The first incident of unsolicited case sharing was noticed using the *patient*45 case as a query with $k = 7$, which resulted in the sharing of unsolicited case *patient*27 out of a maximum of 17 potential cases for the given experimental setup. The number 17 is the result of summation of maximum number of cases from each member for $k = 7$ and subtracting 7 from the summation, that is, $17 = (2 + 7 + 7 + 1 + 7) - 7$ which is based on the total number of cases (2, 25, 12, 1, 10) each member has in its case bases, respectively. Similarly, the second incidence was noticed with *patient*5 for $k = 8$; again, just one patient case out of a maximum of 19 potential cases was shared as unsolicited. Multiple factors, such as poor similarity measures, multiple cases with identical coverage, inadequate case representation, insufficient system testing, poor or erroneous similarity computation algorithm, rounding of similarity score values, missing values in a query or a case, etc., can contribute to such incidents.

The queried case *patient*45, with $k = 7$ and the baseline similarity measure, yielded equal similarity score of 0.768966 for cases *patient*21 and *patient*13. Similarly, *patient*18 and *patient*28 cases have the same similarity score of 0.689655 for the queried case *patient*5 for $k = 8$ with the baseline similarity measure. Consequently, out of 6950, that is, ($50 \times 139$), possible redundant/unsolicited cases for values of $k$ ranging from 1 to 10, merely 2 cases were shared to the federator in the entire experiment. The value 50 corresponds to the number of queries (cases) executed for each $k$ value. And the number 139 results from the summation of all potential redundant cases in a worst-case scenario that might have been shared with the federator, that is, $4 + 7 + 9 + 11 + 13 + 15 + 17 + 19 + 21 + 23 = 139$, where each term in the summation corresponds to $k$ values from 1 to 10.

An essential aspect of the two-stage F-CBR is that the case redundancy is not directly dependent on the number of members participating in the federation. Therefore, the increased number of federated members may not impose a significant risk of increasing case redundancy to the federator.

### *E. HOW MUCH DATA CAN BE MINIMIZED?*

Consider a medical F-CBR system with 50 federated members, where each member system is accessible by only one clinician. Also, assume that each clinician, on average,

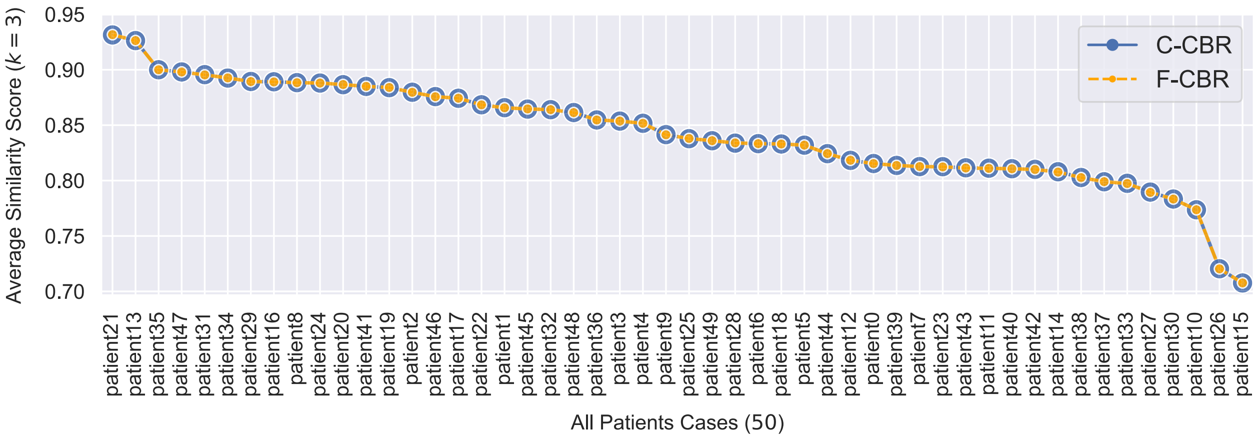



**FIGURE 8.** C-CBR and F-CBR retrieval result comparison based on the average of the top ($k = 3$) retrieved cases' similarity scores for each query, contributing to the y-axis. All the 50 MSD patient cases were used as a query for this experiment and their IDs contribute to the x-axis.

encounters 10 patients in a day and would execute only one query per patient. Let us consider that a clinician needs only 3 most similar patient cases from the F-CBR system to assess or diagnose a new patient. In such a scenario, the federator, in the worst case, would receive 150 patient cases, that is, 3 most similar patients from all the 50 members. However, compared to the best-case scenario with two-stage retrieval as proposed in this paper, the F-CBR system would preserve $((50-1) \times 10 \times 3 \times 50) = 73,500$ cases being unintentionally exposed each day to the federator. Whereas, adequately, only $(1 \times 10 \times 3 \times 50) = 1,500$ patient cases are required in a day for the given scenario.

The above effect can be visualized in Fig. 9, which shows the number of cases that can be preserved from exposure to a federator of an F-CBR system with 50 members in a day. It also showcases the variability of case preservation for increasing the number of desired similar cases required for problem-solving. In principle, the proposed two-stage F-CBR design helps minimize sharing unsolicited cases at the member layer (Fig. 2).

## V. DEPLOYMENT OF A F-CBR SYSTEM

First, one needs to find existing CBR system(s) owners who are willing to participate in a federation without facilitating p2p communication and without losing their anonymity with the aim to improve their decision-making. Then, all the members (the CBR systems available for the federation) need to facilitate the two-stage retrieval as described in in section III-C. When the owner of a legacy system wants to federate their CBR system, they would need to create a wrapper for the legacy CBR system, which implements the Algorithms 2 and 3. The organization or interested vendor, responsible for creating a federation, needs to support a web service that implements the Algorithm 1 and performs the role of a federator. It is desirable that the federator facilitates parallel processing.

Once all the members and the federator are realized, each member needs to register themselves with the federator's member registry. The registry can be dynamic, where members may be free to participate or leave the federation at their will. However, a federator facilitator may impose obligatory rules or guidelines for participating or leaving the federation, which is currently out of the scope of the current work. All the members need to implement an additional service that can send and receive retrieval requests and responses to and from a federator. In the worst-case scenario, there might be only one member, the initiator itself.

The architecture itself does not impose any specific similarity metric to be followed except that all the members produce the same range of similarity values. A CBR engineer on the member CBR side can build similarity measures to meet domain needs by either imposing standard similarity measures across all federated members or by allowing them to develop their own. However, it is expected that all the members put their due diligence while designing their exclusive similarity measures.

A federator may impose that only members can communicate with a federator service. A federator service may not be exposed to a non-member system or to public access for avoiding unsolicited data exposure and access. The federator might also be prohibited from disclosing a member's identity to the others. Additional constraints could be imposed on a federator to verify if a request originated from a legitimate member.

For comparative analysis of the retrieval quality of an F-CBR, a CBR engineer can configure a member to retrieve two sets of retrieval results; one from the federator and the other from its local system using the local case base.

Possible limitations for an F-CBR are that it relies on the internet availability, assumes the similarity measures are relevant and optimal, multiple CBR systems with relevant exclusive experiences without p2p communication exist and are willing to federate. The user of a federator system's retrieval results would be accountable for the decision making, and should use the system as an assistance unless proven to operate independently. The reason is that there can be a scenario that a federation might not have or is with limited prior experience to solve a new problem. Every member might be responsible for their similarity measures

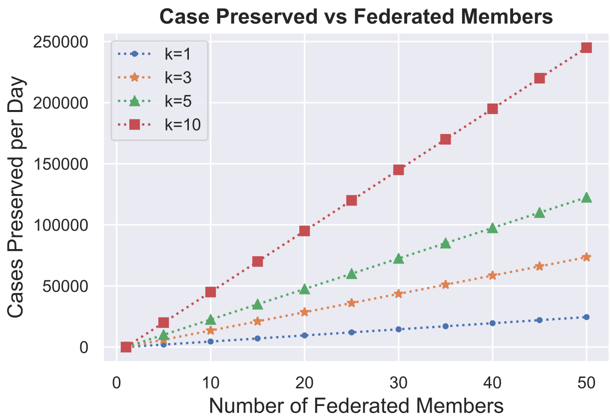



**FIGURE 9. Data preservation per day for a best-case scenario with respect to the number of member CBR systems and a varying number of cases (*k*) adequately needed to solve a problem.**

until mandated by the federation. Also, members may have different or conflicting case representations that must be resolved before registering to a federator. Fine grained requirement details are beyond the scope of this study.

## VI. CONCLUSION

Case-based reasoning, a problem-solving methodology in AI, drives its competence mostly from the cases it has amassed or learned. Therefore, cases gathered from a single institution or geographical region are seldom adequate to solve diverse problems, especially in rare situations. Moreover, to preserve their autonomy, these institutions often do not promote peer-to-peer communication between their CBR systems nor encourage data consolidation and persistence in a centralized location. The proposed two-stage F-CBR architecture design, extensible to hierarchical institutional structures, mitigates these challenges while collaborating such CBR systems into a federation. In this approach, the members are not required to possess peer-to-peer communication capabilities. The suggested design also contributes to compliance with the data minimization principle of the GDPR, the European privacy and security regulation. We have demonstrated in this research that prototype F-CBR systems implemented for two public datasets and a private real-world dataset can attain the same retrieval quality as a conventional CBR system built on the same datasets. Consequently, we anticipate that the two-stage F-CBR design will be a viable approach for federating CBR systems for real-world applications. The developed CBR systems are made available on GitHub public repository for reproducibility and further study.

## REFERENCES


[1] J. Shaw, F. Rudzicz, T. Jamieson, and A. Goldfarb, "Artificial intelligence and the implementation challenge," *J. Med. Internet Res.*, vol. 21, no. 7, Jul. 2019, Art. no. e13659.

[2] N. Noorbakhsh-Sabet, R. Zand, Y. Zhang, and V. Abedi, "Artificial intelligence transforms the future of health care," *Amer. J. Med.*, vol. 132, no. 7, pp. 795–801, Jul. 2019.

[3] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, vol. 10, 1st ed. Cham, Switzerland: Springer, 2017.

[4] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[5] L. Longo, R. Goebel, F. Lecue, P. Kieseberg, and A. Holzinger, "Explainable artificial intelligence: Concepts, applications, research challenges and visions," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction*. Cham, Switzerland: Springer, 2020, pp. 1–16.

[6] R. C. Schank, *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. New York, NY, USA: Cambridge Univ. Press, 1983.

[7] J. L. Kolodner, "An introduction to case-based reasoning," *Artif. Intell. Rev.*, vol. 6, no. 1, pp. 3–34, 1992.

[8] W. Cheetham and I. Watson, "Fielded applications of case-based reasoning," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 321–323, Sep. 2005.

[9] S. Montani and L. C. Jain, *Successful Case-Based Reasoning Applications*, 1st ed. Berlin, Germany: Springer, 2010.

[10] A. Holt, I. Bichindaritz, R. Schmidt, and P. Perner, "Medical applications in case-based reasoning," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 289–292, 2005.

[11] N. Choudhury and S. Ara, "A survey on case-based reasoning in medicine," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 8, pp. 136–144, 2016.

[12] R. L. De Mantaras, D. Mcsherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T. Cox, K. Forbus, M. Keane, A. Aamodt, and I. Watson, "Retrieval, reuse, revision and retention in case-based reasoning," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 215–240, Sep. 2005.

[13] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 1994.

[14] J. Kim, H. Ha, B.-G. Chun, S. Yoon, and S. K. Cha, "Collaborative analytics for data silos," in *Proc. IEEE 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 743–754.

[15] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *npj Digit. Med.*, vol. 3, no. 1, pp. 1–7, Dec. 2020.

[16] A. Sadilek, L. Liu, D. Nguyen, M. Kamruzzaman, S. Serghiou, B. Rader, A. Ingerman, S. Mellem, P. Kairouz, E. O. Nsoesie, J. MacFarlane, A. Vullikanti, M. Marathe, P. Eastham, J. S. Brownstein, B. A. Y. Arcas, M. D. Howell, and J. Hernandez, "Privacy-first health research with federated learning," *npj Digit. Med.*, vol. 4, no. 1, pp. 1–8, 2021.

[17] M. Roussopoulos, M. Baker, D. Rosenthal, T. Guili, P. Maniatis, and J. Mogul, "2 P2P of not 2 P2P?" in *Proc. Int. Workshop Peer-Peer Syst.*, 2004, pp. 26–27.

[18] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-peer computing," HP Labs, Palo Alto, CA, USA, Tech. Rep., HPL-2002-57, 2002.

[19] B. Pourebrahimi, K. Bertels, and S. Vassiliadis, "A survey of peer-to-peer networks," in *Proc. 16th Annu. Workshop Circuits, Syst. Signal Process.*, 2005, pp. 570–577.

[20] C.-R. Shyu, K. T. Putra, H.-C. Chen, Y.-Y. Tsai, K. S. M. T. Hossain, W. Jiang, and Z.-Y. Shae, "A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications," *Appl. Sci.*, vol. 11, no. 23, p. 11191, Nov. 2021.

[21] S. Busse, R.-D. Kutsche, U. Leser, and H. Weber, "Federated information systems: Concepts, terminology and architectures," *Forschungsberichte Fachbereichs Informatik*, vol. 99, no. 9, pp. 1–38, 1999.

[22] P. Baran, "On distributed communications networks," *IEEE Trans. Commun. Syst.*, vol. CS-12, no. 1, pp. 1–9, Mar. 1964.

[23] G. Fernandez, L. Zhao, and I. Wijegunaratne, "Patterns for federated architecture," *J. Object Technol.*, vol. 2, no. 1, pp. 135–149, 2003.

[24] J. Collier, "What is autonomy?" in *Proc. 5th Int. Conf. Comput. Anticipatory Syst. (CASY)*, vol. 12, 2002, pp. 212–221.

[25] M. van Steen and A. S. Tanenbaum, "A brief introduction to distributed systems," *Computing*, vol. 98, no. 10, pp. 967–1009, 2016.

[26] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, in Proceedings of Machine Learning Research, vol. 54, Apr. 2017, pp. 1273–1282.

[27] S. Shen, T. Zhu, D. Wu, W. Wang, and W. Zhou, "From distributed machine learning to federated learning: In the view of data privacy and security," 2020, *arXiv:2010.09258*.

[28] A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM Comput. Surv.*, vol. 22, no. 3, pp. 183–236, Sep. 1990.
[29] M. Shokouhi and L. Si, "Federated search," *Found. Trends® Information Retrieval*, vol. 5, no. 1, pp. 1–102, 2011.
[30] K. Zhang, X. Song, C. Zhang, and S. Yu, "Challenges and future directions of secure federated learning: A survey," *Frontiers Comput. Sci.*, vol. 16, no. 5, pp. 1–8, Oct. 2022.
[31] L. M. Ginty and B. Smyth, "Collaborative case-based reasoning: Applications in personalised route planning," in *Proc. Int. Conf. Case-Based Reasoning*. Berlin, Germany: Springer, 2001, pp. 362–376.
[32] S. Aguzzoli, P. Avesani, and P. Massa, "Collaborative case-based recommender systems," in *Proc. Eur. Conf. Case-Based Reasoning*. Berlin, Germany: Springer, 2002, pp. 460–474.
[33] P. Hu, D.-X. Gu, and Y. Zhu, "Collaborative case-based reasoning for knowledge discovery of elders health assessment system," *Open Biomed. Eng. J.*, vol. 8, p. 68, Jan. 2014.
[34] E. Plaza and L. McGinty, "Distributed case-based reasoning," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 261–265, 2005.
[35] E. Plaza and S. Ontanón, "Ensemble case-based reasoning: Collaboration policies for multiagent cooperative CBR," in *Proc. Int. Conf. Case-Based Reasoning*. Berlin, Germany: Springer, 2001, pp. 437–451.
[36] M. A. Jubair, S. A. Mostafa, A. Mustapha, and H. Hafit, "A survey of multi-agent systems and case-based reasoning integration," in *Proc. Int. Symp. Agent, Multi-Agent Syst. Robot. (ISAMSR)*, Aug. 2018, pp. 1–6.
[37] A. Jaiswal and K. Bach, "A data-driven approach for determining weights in global similarity functions," in *Proc. Int. Conf. Case-Based Reasoning*. Cham, Switzerland: Springer, 2019, pp. 125–139.
[38] M. Zwitter and M. Soklic. (1988). *Lymphography*. UCI Machine Learning Repository. [Online]. Available: https://archive-beta.ics.uci.edu/ml/datasets/lymphography
[39] R. Forsyth. (1990). *Zoo*. UCI Machine Learning Repository. [Online]. Available: https://archive-beta.ics.uci.edu/ml/datasets/zoo
[40] K. A. I. Evensen, H. S. Robinson, I. Meisingset, A. Woodhouse, M. Thielemann, W. S. Bjorbækmo, G. Myhre, A. E. Hansen, O. Vasseljen, and N. K. Vøllestad, "Characteristics, course and outcome of patients receiving physiotherapy in primary health care in Norway: Design of a longitudinal observational project," *BMC Health Services Res.*, vol. 18, no. 1, pp. 1–10, Dec. 2018.
[41] S.-A. Brown, "Patient similarity: Emerging concepts in systems and precision medicine," *Frontiers Physiol.*, vol. 7, p. 567. Nov. 2016.
[42] E. Parimbelli, S. Marini, L. Sacchi, and R. Bellazzi, "Patient similarity for precision medicine: A systematic review," *J. Biomed. Informat.*, vol. 83, pp. 87–96, Jul. 2018.
[43] K. Bach, B. M. Mathisen, and A. Jaiswal, "Demonstrating the myCBR rest API," in *Proc. ICCBR Workshops*, vol. 2567, 2019, pp. 144–155.
[44] M. M. Richter and R. O. Weber, *Case-Based Reasoning*, 1st ed. Berlin, Germany: Springer, 2016.
[45] N. Dragoni, I. Lanese, S. T. Larsen, M. Mazzara, R. Mustafin, and L. Safina, "Microservices: How to make your application scale," in *Proc. Int. Andrei Ershov Memorial Conf. Perspect. Syst. Inform.* Cham, Switzerland: Springer, 2017, pp. 95–104.
[46] J. Foote, "Visualizing music and audio using self-similarity," in *Proc. 7th ACM Int. Conf. Multimedia*, 1999, pp. 77–80.
[47] M. Müller and M. Clausen, "Transposition-invariant self-similarity matrices," in *Proc. 8th Int. Conf. Music Inf. Retr.*, 2007, pp. 47–50.
[48] T. M. Mitchell, "Version spaces—A candidate elimination approach to rule learning," in *Proc. 5th Int. Joint Conf. Artif. Intell.* San Francisco, CA, USA: Morgan Kaufmann, 1977, pp. 305–310.
[49] A. Jaiswal, K. Bach, I. Meisingset, and O. Vasseljen, "Case representation and similarity modeling for non-specific musculoskeletal disorders—A case-based reasoning approach," in *Proc. 32nd Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, 2019, pp. 359–363.

**AMAR JAISWAL** was born in Ayodhya, Uttar Pradesh, India, in 1984. He received the B.E. degree in medical electronics from Visvesvaraya Technological University (VTU), India, in 2007, and the M.S. degree in medical software from Manipal University, India, in 2009. He is currently pursuing the Ph.D. degree in computer science with the Data and Artificial Intelligence (DART) Group, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

From 2008 to 2011, he worked as a Software Engineer with the Positron Emission Tomography (PET) and Computerized Tomography (CT) Division, General Electric (GE) Healthcare, Bengaluru, India. From 2011 to 2014, he worked with the X-ray and Mammography Division, Philips Healthcare, Bengaluru. From 2014 to 2017, he was involved in developing telemedicine and remote consultation solutions with Ricoh Healthcare, Bengaluru. Lately, from 2021 to 2022, he worked at the Norwegian Centre for E-health Research, Trømso, Norway. His research interests include case-based reasoning and artificial intelligence for medical applications.

**KASSAYE YITBAREK YIGZAW** was born in Addis Ababa, Ethiopia, in 1986. He received the B.Sc. degree in electrical engineering and the M.Sc. degree in telemedicine and e-health from Hawassa University, Ethiopia, and the Ph.D. degree in computer science from the University of Tromsø–The Arctic University of Norway, in 2017. From 2017 to 2020, he was a Researcher at the Norwegian Centre for E-health Research, where he has been a Senior Researcher, since 2020. He has long experience in various e-health research projects with a research focus on the security and privacy aspects.

**PINAR ÖZTÜRK** received the Ph.D. degree in computer science from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2000.

From 2001 to 2021, she has been an Associate Professor. Since 2021, she has been a Professor with the Artificial Intelligence Group, Department of Computer Science, NTNU. Her main research interest includes artificial intelligence but she does multidisciplinary research linking AI with other parts of cognitive science. Her research activities lie in knowledge representation and modeling, case-based reasoning, multiagent systems, and recently bias and (un)fairness in AI systems.