Health and Technology (2022) 12:285–304
https://doi.org/10.1007/s12553-022-00640-3

REVIEW PAPER

Privacy-preserving deep learning for pervasive health monitoring: a study of environment requirements and existing solutions adequacy

Amine Boulemtafes[1] · **Abdelouahid Derhab**[2] · **Yacine Challal**[3]

Received: 29 July 2021 / Accepted: 21 January 2022 / Published online: 4 February 2022

Abstract
In recent years, deep learning in healthcare applications has attracted considerable attention from research community. They are deployed on powerful cloud infrastructures to process big health data. However, privacy issue arises when sensitive data are offloaded to the remote cloud. In this paper, we focus on pervasive health monitoring applications that allow anywhere and anytime monitoring of patients, such as heart diseases diagnosis, sleep apnea detection, and more recently, early detection of Covid-19. As pervasive health monitoring applications generally operate on constrained client-side environment, it is important to take into consideration these constraints when designing privacy-preserving solutions. This paper aims therefore to review the adequacy of existing privacy-preserving solutions for deep learning in pervasive health monitoring environment. To this end, we identify the privacy-preserving learning scenarios and their corresponding tasks and requirements. Furthermore, we define the evaluation criteria of the reviewed solutions, we discuss them, and highlight open issues for future research.

Keywords Deep learning · Deep neural network · Privacy · Pervasive health monitoring · e-Health · m-Health

1 Introduction

Deep learning (DL) for healthcare is nowadays one of the most attractive research topics, which covers different applications related to electronic health records, wearable computing, and genomics analysis [1]. Pervasive health monitoring (PHM) is one of the most interesting healthcare applications, which allow anywhere and anytime monitoring of patients. With the increasing technological advancements in sensing platforms and rapid development of machine and deep learning, more interesting PHM applications are deployed. In fact, by combining wearables and sensing platforms with the power of deep learning, PHM applications are able to target various health concerns and diseases like pneumonia, sleep apnea, heart health assessment, or even the nowadays worldwide pandemic Covid-19 [2–9]. The Defense Threat Reduction Agency and Defense Innovation Unit of US Department of Defense, for instance, is working since a few years on RATE (Rapid Analysis of Threat Exposure) technology [2]. It consists of non-invasive wearable devices that measure key biomarkers, and process them on the cloud with the help of artificial intelligence and machine learning for early detection of infections. RATE technology was tested on different infections such as pneumonia, SARS, and more recently Covid-19.

As in many domains, deep learning capabilities in the healthcare domain are often improved by leveraging powerful cloud infrastructures [10], especially in case of PHM applications. In fact, PHM operates between the client the remote cloud server. It generally relies on constrained client devices like sensors and mobile devices, as well as on different communication networks between the client and the cloud, some of which may be unreliable or costly. Such

✉ Amine Boulemtafes
aboulemtafes@cerist.dz

Abdelouahid Derhab
abderhab@ksu.edu.sa

Yacine Challal
y_challal@esi.dz

[1] Division Sécurité Informatique, Centre de Recherche sur l'Information Scientifique et Technique, Algiers, Algeria, and also Département Informatique, Faculté des Sciences exactes, Université de Bejaia, Bejaia, Algeria

[2] Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

[3] Laboratoire de Méthodes de Conception des Systèmes, Ecole Nationale Supérieure d'Informatique, Algiers, Algeria

a technological configuration potentially represents a constrained client-side environment. Wearables and mobile devices are resource-constrained, from the hardware point of view, and because of the daily usage of multiple apps that could quickly deplete the mobile device's battery. Such potential high load on the mobile device should not disrupt their daily usage by the user, nor the functionality of the PHM application.

However, leveraging the clouds comes at the expense of privacy when sensitive data is offloaded to train deep learning models or requesting inferences [10]. In the context of PHM, designing privacy-preserving solutions is impacted by the constrained client-side environment, including devices requirements, communication network impediments, as well as effectiveness requirements of the PHM application.

- **Related work**

Much efforts were devoted to design efficient solutions for privacy-preserving deep learning. Zhang et al. [11] reviewed some solutions, particularly those related to collaborative learning, and considered the two key phases of deep learning, i.e., training and inference. Chang and Li [12] focused on privacy issues during training and inference phases, including attacks on trained models, along with their corresponding threats and countermeasures. More recently, Tanuwidjaja et al. [13] discussed a number of privacy-preserving solutions based on three concepts, namely, homomorphic encryption, multiparty computation and differential privacy. The survey also presented a comparison of the reviewed solutions under each concept. Similarly, Riazi et al. [14] reviewed privacy-preserving solutions for deep learning, but focused on cryptographic methodologies. The review also presented solutions description and performance comparisons, along with main attacks on deep neural networks (DNNs). Boulemtafes et al. [10] also presented a recent review of existing privacy-preserving solutions for deep learning along with their evaluation results, and highlighted open research along with suggested recommendations. However, the above-mentioned surveys only addressed the privacy issue in a general context, which do not consider specific target environment constraints.

Zheng et al. [15] focused on the IoT context, and presented a taxonomy of different privacy-preserving machine learning approaches for training and inference phases, then discussed the limitations of applying them on IoT end-devices. In the same work, the authors introduced a privacy-preserving inference solution based on obfuscation. The authors further detailed their solution in [16]. However, the review does not give a detailed description of existing solutions, but only presents a brief summary of limitations and drawbacks of classes of privacy-preserving solutions. Moreover, the limitations are not evaluated based on a set of criteria. The review also does not differentiate between training local and remote models.

Differently from related work, and particularly from [15], this study:

- Focuses on:
  - Privacy-preserving deep learning, including inference and training of both local and remote models,
  - PHM applications, i.e., it considers PHM architecture and constraints,
  - Particularly constrained client-side environment at IoT and edge computing level.
- Identifies the privacy-requiring scenarios and constraints of the target context, and defines solutions requirements.
- Reviews the adequacy of each solution with the target context, using the set of defined evaluation criteria. Reviewed solutions include the approach proposed in [15, 16].
- Discusses privacy-preserving approaches for deep learning with respect to key technological concepts.
- Outlines open research challenges.

To this end, privacy-requiring scenarios are defined, and a number of recent solutions for privacy-preserving deep learning are evaluated against criteria derived from environment constraints and requirements of target solution. More specifically, we make the following contributions:

(1) We present a generic architecture for deep learning-based PHM, i.e., the main components and their roles, as well as local and remote analysis scenarios.
(2) For each scenario, we identify the required corresponding tasks.

For example, in order to perform local analysis, the local model needs first to be trained either individually by a single client or collaboratively among different clients. Once trained, the model can be used for inference at the client level.

(3) For each task, we identify the privacy properties that need to be ensured.
(4) We present the target environment constraints, and identify its corresponding requirements.
(5) From the identified environment requirements, we define a set of criteria in order to evaluate the adequacy of reviewed solutions to the target environment.
(6) We classify the reviewed solutions according to key concepts, and evaluate them against defined criteria.
(7) We discuss the evaluation study, the drawbacks of solutions, and the impact of introducing privacy on deep learning-based PHM applications.

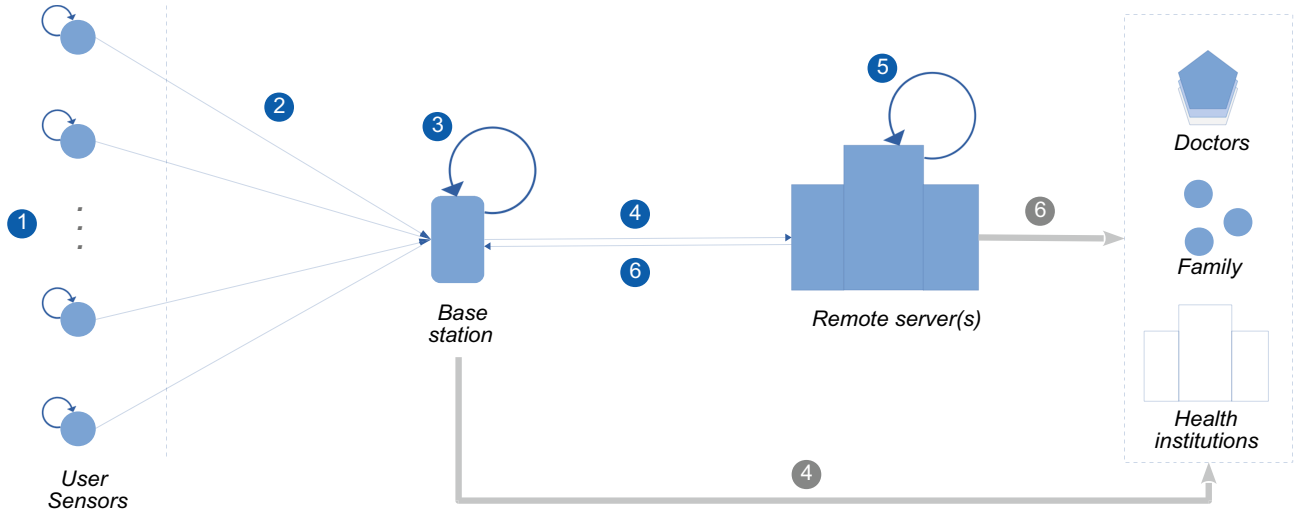

Fig. 1 DL-based PHM components and process flow

(8) We outline for each key concept, a set of recommendations for future research directions to address the identified limitations.

The remainder of this paper is organized as follows: Sect. 3.2 defines generic PHM architecture including components and process flow. Section 4 identifies PHM scenarios requiring privacy preservation. Section 4.1 studies the environment constraints and requirements of solution, and defines a set of related evaluation criteria. Section 5 evaluates and discusses the solutions. Section 6 presents open challenges and outlines some potential future research directions. Finally, Sect. 7 concludes the paper.

2 A generic architecture for DL-based PHM

PHM is one of the main applications of pervasive healthcare, which provides preventive healthcare and deals with emergencies using ubiquitous computing technology. PHM allows anywhere and anytime monitoring, and generally relies on a three-tier architecture comprising (i) sensors or medical devices, (ii) a base station, and (iii) servers [17–20].

We propose and describe below a generic architecture, and a process flow of a deep-learning-based PHM.

2.1 Components

As shown in Fig. 1, the PHM architecture is composed of the three following main components:

- **Sensors** They capture data such as pulse rate and body temperature, and perform preprocessing and basic processing. They include wearable and ambient sensors.
- **Base station** It gathers data from sensors, performs preprocessing and real-time analysis, displays results, issues alerts, and transmits data/results to the remote servers, and any relevant system's actor such as doctors or family. It could be a tablet or a raspberry, or a mobility support device like the user mobile phone.
- **Remote servers** They perform deeper and more powerful analysis, and share the results with the base station and/or other system's actors.

2.2 Process flow

Figure 1 also shows the following generic process flow of information.

- At sensors level:

1. Data is captured, and preprocessed.
   Basic processing can also be performed (such as test measurements against thresholds).
2. Data is transmitted to the base station.

- At base station level:

3. Data is preprocessed,
   then locally analyzed for real-time/emergency inference results (such as fall detection, or dangerous sleep postures).
4. Data/results are transmitted to the remote servers, and any relevant system's actor such as doctors.

- At remote servers level:

5. A *more deeper analysis* is performed using deep learning (such as symptoms detection, potential early disease prediction, or health status prediction).
6. Results are shared with the user (base station) and/or other actors of the system.

3 PHM scenarios and privacy requirements

As described in the process flow, two main deep-learning-based scenarios can be distinguished in PHM applications:

(i) *Local analysis*, performed by the base station, in order to provide real-time and emergency results.

(ii) *Remote analysis*, performed by remote servers, in order to conduct more complex analysis.

3.1 Local DL-based analysis scenario

Before performing analysis, a local deep learning model needs first to be trained, which can be done:

(i) *Individually*, i.e., by a single client and with the help of the remote servers, especially if the client is resource-constrained.

(ii) C*ollaboratively,* where multiple participants jointly take part in the deep learning process. Collaborative learning can be achieved directly between the participants, or with the help of the remote servers for coordination and aggregation of updates.

Once trained, the model can be used locally by each client in order to get inferences. Therefore, only the training phase requires privacy-preservation.

- **Privacy requirements**

During individual training, private training data of the client, and maybe the model need to be protected from the remote servers.

During collaborative training, private data of each participant need to be protected from the remote servers, as well as from the other participants. Also, the model might also need to be protected from the remote server. Once the trained model is distributed to the participants, original private training data of each participant should not be leaked by others..

3.2 Remote DL-based analysis scenario

Before deploying a model into production, i.e., on the remote server, it needs to be trained by the different participants in order to take advantage of the whole shared data.

Once trained, the model can be used remotely by clients in order to get inferences. Therefore, both training and inference phases require to preserve privacy.

- **Privacy requirements**

During the training phase, private training data of the participants need to be protected from the remote server, while the model might also need to be protected from the participants.

When the trained model is used by clients to get inferences, the model might need to be protected from the clients, while their private data and maybe inferences should also be protected from the remote server hosting the model.

4 PHM constraints, solution requirements and evaluation criteria

4.1 PHM environment

In the PHM environment, we can find three constrained elements, namely, the client-side devices, the input data, and the communication network.

- **Client-side devices** Sensors and mobile devices have limited resources, in terms of both power and computation. Sensors are generally considered as low-resourceful from a hardware perspective. On the other hand, mobile devices, particularly smartphones, are nowadays generally powerful, however, they run different daily applications and are often continuously powered on, which might constraint them in terms of energy. Also, any background task should be performed in transparency and not affect the performances of other applications.
- **Input data** In a healthcare context, raw data can have different modalities like voice, images, texts, signals, and so on. It is therefore important to consider this heterogeneity and multi-modality of data when designing a privacy-preserving solution, especially if multiple sensitive information need to be protected. As shown in [21], the obfuscation technique, which is used to protect input data, needs more investigation in case the sensitive data to be protected are collected from multiple sensors.
- **Communication network** Different communication networks are used in the PHM environment such as cellular, wifi, bluetooth, …etc. At the client-side, cellular networks are generally used in order to communicate with the cloud. Depending on the client's mobile internet plan, a large amount of data continuously exchanged, can be costly to the client. It can also lead to a quick consumption of the available data, which disconnects the client from the cloud. Moreover, as

disconnections can occur in current networks [22], unreliability of the client-side connectivity needs to be considered.

4.2 Privacy-preserving solutions requirements

By considering the needs of the healthcare domain, and PHM environment constraints, a number of requirements need to be considered for the design of privacy-preserving solutions for DL-based PHM.

- **Effectiveness** Because people's health is crucial, the accuracy of results in health applications (such as the ability to detect the symptoms of a disease) is a very important criteria of effectiveness. For this reason, it is important to ensure that integrating privacy preserving into deep learning analysis of health data (including multi-modal data), still keeps high accuracy, i.e., similar or close to non-privacy-preserving models.
- **Efficiency** Due to the previously described PHM environment constraints, it is important to ensure low communication and computation overheads at the client-side, as well as the support of dropouts and disconnections of clients. Server-side overhead is not considered since servers are supposed to have all the required resources.
- **Privacy** As described in the privacy-preserving deep learning scenarios, different information need to be protected in PHM applications. It is important to protect the users sensitive input data of training and inference from both the server (cloud) and other participants. Intermediate results produced during the execution of the model also need to be protected as they can leak some sensitive information. Depending on the target application and users privacy concerns, the resulting inferences might also need to be hidden from the cloud. Besides, the providers of deep learning models may also require that their models are kept private.

In addition to the main above requirements, the following features are needed:

- Support of any deep learning architecture, including:
  - Any model, or at least popular ones used in healthcare like MLP, CNN, and LSTM.
  - Any activation function.
  - Any number of layers (depth of the model), particularly knowing that some existing privacy-preserving solutions were shown to be weak under very deep models [23].
- Support of continuously trained models, i.e., models that are retrained periodically in order to enhance their performance [24, 25].

4.3 Evaluation criteria

In Table 1, we define a set of criteria to be considered to evaluate the adequacy of solutions to PHM requirements, including the three main tasks: privacy-preserving training of local models, privacy-preserving training of remote models, and privacy-preserving remote inference.

We carry out the evaluation by assuming an honest-but-curious (HbC) adversary model, where the parties, including the cloud and participants, are honest, i.e., they follow the protocol, but at the same time, they are curious, i.e., they can try to deduce private information within the limits of what the protocol allows [26]. A more honest adversary model can be considered as a limitation, while supporting more curious or less-honest parties can be considered as an advantage.

5 Existing privacy-preserving solutions vs PHM environment

In this section, we present a set of recent solutions covering the above three privacy-preserving tasks, including training a local model, training a remote model, and remote inference i.e., requesting inference through a remote model.

We evaluate the solutions against the defined criteria for adequacy to PHM requirements (see Table 1). Unless mentioned, the reviewed solutions adopt the HbC adversary model.

Noting that the effectiveness results of the solutions are mainly taken from their respective published papers, and they are based on different datasets and experimentation settings. Therefore, these results cannot serve as a base for a fair comparison between the reviewed solutions or key concepts, and thus they are only reported in our survey in order to identify the most likely causes of accuracy loss. However, an observed loss does not necessarily imply a low accuracy. As mentioned above, a high accuracy depends on many parameters, including the evaluation settings.

5.1 Privacy-preserving training of a local model

Table 2 summarizes the main characteristics of the reviewed privacy-preserving solutions for training a local model, which are classified according to four main technologies, representing the key base concepts that are used by these solutions, namely, homomorphic encryption (HE), partial sharing (i.e., share only a fraction of locally learned parameters for global aggregation on the cloud), transformation of sensitive information, and shared model (i.e., a model is pre-trained by the cloud then is fine-tuned by the clients).

Table 3 evaluates the solutions against the criteria defined in Sect. 4.3. Although these solutions are designed for training remote models, some of them can be used to train a local

Table 1 Evaluation criteria for PHM environment adequacy

	Training of a local model	Training of a remote model	Remote inference
Effectiveness	[**Preserve**]* Accuracy		
Efficiency	[**Low**]* Overhead in terms of Communication and Computation at the client-side (**OVC**) [**No**]* Impact of client Dropout on a Training round (**DIT**), and on an Inference process (**DII**)		
Privacy	[**Protect**]* Training Data (**TD**) from the participants and the cloud, as well as the Local Model (**LM**) from the cloud	[**Protect**]* Training Data (**TD**) from the participants and the cloud, as well as the Remote Model (**RM**) from the participants	[**Protect**]* Input Data (**IN**) and Inferences (**IF**) from the cloud, as well as the Remote Model (**RM**) from the client

[]*: desired properties

model, but with some privacy gaps, as shown in the Table 3. The table cells, which are highlighted in gray, show the main criteria that are not fully satisfied, along with their limitations that are underlined.

- **Discussion**

Homomorphic encryption (HE) HE-based solutions are characterized by high overhead at the client-side, especially those based on fully HE. Moreover, the solutions relying on local training at each round increase the overhead in terms of both computation and communication. Besides, HE-based solutions for collaborative learning can preserve high accuracy, as well as individual learning, if polynomial approximation is not involved [42]. Client dropout impact on the training round process is generally low, but may increase in case of solutions that require coordination or a threshold of participants to update the global model at each round. As for privacy, deep learning network structure needs to be shared with the cloud in individual learning solutions, however, training data are protected through encryption, and model parameters are ensured to be protected through encrypting shared weights. In collaborative learning solutions, training data and network structure do not need to be shared with the cloud. The model parameters are protected by encryption, and if more security is needed perturbation is added. However, perturbation leads to a trade-off between accuracy and privacy. In general, HE-based solutions do not make restrictions on deep model or activation function, except for solutions that require polynomial approximation. However, an adaptation of the model to the HE domain is needed as explained in [43].

Partial sharing (PS) PS-based solutions are characterized by high client-side overhead due to local training. They also make a trade-off between accuracy and privacy, which is controlled by the fraction of parameters shared and the level of perturbation. Besides, the impact of a participant's dropout on the training round process is low, and no restrictions are made on the deep model and activation functions. As for privacy, training data and network structure do not need to be shared, while only a fraction of local parameters is revealed to the cloud. However, the computed global model gradients remain in clear, and thus are not protected.

Transformation Transformation-based solutions can reach high accuracy but they are characterized by high overhead due to local training, and lead to a trade-off between accuracy and privacy due to transformations or perturbations that are applied on data or objective functions [44]. In fact, applying more transformations allows for better data protection, but leads to less accuracy. Moreover, in such solutions,

Table 2 Privacy-preserving solutions for training a local model

Key concept	Ref	Main characteristics
Homomorphic Encryption	Q. Zhang et al. [27]	Fully BGV HE | Taylor theorem polynomial approximation
	Bu et al. [28]	Fully BGV HE | Maclaulin formula polynomial approximation
	Phong et al. [29]	Partially additive LWE-based and Paillier HE | TLS/SSL secure channels
	X. Zhang et al. [30]	Partially lightweight El Gamal HE | Shamir's threshold secret sharing | Local differential privacy
	Hao et al. [31]	Partially additive HE | Differential privacy
Partial sharing	Shokri and Shmatikov [32]	Partial sharing of parameters | Laplace differential privacy | Sparse vector technique
	Liu et al. [33]	Partial sharing of parameters
Transformation	Zhao et al. [34]	Functional exponential mechanism | Polynomial approximation for objective function | Cryptography and hashing against eavesdrop attacks
	Hartmann and West [35]	Cancelable noise (differential privacy) | Anonymization network (such as Tor)
	Fu et al. [36]	Mixup data augmentation
Shared model	Servia-Rodriguez et al. [37]	Start from the weights and bias of the shared model | (Optional) Differential privacy for training the shared model

the model generally needs to be shared with the cloud. Noting that functional mechanism used in [34] might need some adaptations of the deep model, while the technique in [35] uses negative value vectors to make the noise cancelable, which makes the impact of a participant's dropout high, as it prevents global gradient from being revealed.

Shared model Although the solution based on shared model concept can reach high accuracy, it requires first training a shared model at the cloud, using a set of non-private training data (such as voluntarily shared data). Moreover, a client needs sufficient local samples in order to personalize its local model, which leads to a high overhead in terms of computation. Besides, the dropout of the client has no impact on the training process once the shared model is received, since it is fine-tuned locally and independently from the cloud. As for privacy, the training data are not shared, and reversing the shared model does not reveal any private information. However, structure of the shared model and its parameters are hosted and trained at the cloud, and only personalized trained parameters are protected.

Key concepts comparison By going through the reviewed solutions, we summarize in Table 4 the key concepts adequacy with the PHM environment mainly with respect to effectiveness, client-side efficiency and privacy guarantees. The main potential limitations are underlined in the table.

As for individual learning, both HE and SM concepts require sharing the model structure, although it is not necessarily considered as a privacy issue. In case of relatively small deep networks, SM might be the most interesting concept, as it may offer the best effectiveness, provided that enough local training data is available. More deeper networks may not be supported by SM because of the required local training performed on constrained client devices. On the other hand, HE-based reviewed solutions show good performances, although accuracy loss depends on polynomial approximation of activation functions. Besides, HE-based solutions still need to address the challenge of low client-side overhead.

Regarding collaborative learning, the different reviewed key concepts face challenges in terms of trade-off between accuracy and privacy due to the perturbation impact. Moreover, achieving low client-side overhead remains challenging with the required local training, unless deep networks are enough small to be supported by the client devices. Besides, among the different reviewed solutions, only those based on HE could successfully protect the local model.

5.2 Privacy-preserving training of a remote model

Table 5 summarizes the main characteristics of the reviewed privacy-preserving solutions for training a remote model, which are classified according to four main technologies, representing the key base concepts that are used by these solutions, namely, homomorphic encryption (HE), partial sharing, transformation of sensitive information, and model splitting between the client and the remote side.

Table 6 evaluates the above solutions against the criteria defined in Sect. 4.3. Although these solutions are designed for training local models, some of them can be however used to train a remote model, but with some privacy gaps, as shown in Table 6. The table cells, which are highlighted in gray, show the main criteria that are not fully satisfied, along with their limitations that are underlined.

Table 3 Solutions for training a local model *vs* evaluation criteria

Ref	Effectiveness	Efficiency		Privacy guarantees and limitations		Notes
	Accuracy	OVC	DIT	TD*	LM	
[27] ind	*Can be high but with loss(1)*	*High(4)(5)*	None(7)	TD [and weights] encrypted	weights encrypted *but (12)*	AFR: *APX*
[28] ind	*Can be high but with loss(1)*	*High(4)(5)*	None(7)	TD [and weights] encrypted	weights encrypted *but (12)*	AFR: *APX*
[29] col	SAN	*High(4)(5)(6)*	None(8)	TD not shared [and local shared weights encrypted] *but weak against attacks between participants* [13, 38]	not shared	ADV: *honest participants*
[30] col	CAN *but (2)*	*High(4)(5)(6)*	*High*	TD not shared [and shared gradients perturbed and encrypted, *but (10)*]	not shared	-
[31] col	Quite high *but (2)*	*High(4)(5)(6)*	None(8)	TD not shared [and shared gradients perturbed and encrypted, *but (10)*]	not shared	ADV: tolerate collusion of server with multiple users
[32] col	CAN *but (2) and criticized in* [29]	*High(5)(6)*	None(8)	TD not shared [and only fraction of local parameters shared perturbed, *but (10)*	not shared *but (11) < might be a privacy concern if server is not trusted*	ADV: trusted server
[33] col	RR ~ 90% *but (2)*	*High(5)(6)*	Round-robin: *High* Asynchronous: None(8)	TD not shared [and only fraction of local parameters shared, *but (10)*]	not shared, *but (11)*	- *can be concerned by the privacy critics reported in* [29, 39–41]
[34] col	MRE High *but (2)*	*High(5)(6)*	None(8)	TD not shared [and objective functions (thus gradients) perturbed, *but (10)*]	*shared > might be a privacy concern if server is not honest*	ADV: *honest server*, active & passive participants AFR/MOR: *may require adaptations*
[35] col	N/A	*High(5)(6)*	*High*	TD not shared [and local gradients perturbed with cancelable noise]	*shared*	ADV: malicious server, honest participants > = 2 - *anonymization network required*
[36] col	Up to High *but (2)*	*High(5)(6)*	[None(8) | *High*] (9)	TD not shared [and local parameters obtained from mixup input data, *but (10)*]	*shared*	-
[37] ind	Up to High *but (3)*	*High(6)*	None	TD not shared +resistant to model inversion	*(12)*	-

ind Individual training, *col* Collaborative Accuracy training mainly on the basis of authors evaluations, *SAN* can reach Same As Non-private model, *CAN* can reach Close to/Almost Non-private model, *RR* Reconstruction Rate, *MRE* Mean Relative Error, *OVC* overhead is relative to the non-private model, *DIT [col] high* if the training round stops for all participants, none, if the training round is not affected, *[ind] high* if the participant training process stops, *none* if the process does not stop, *ADV* Adversary model, *AFR* Activation Function Restrictions, *MOR* Model Restrictions, *APX* Approximated

(1) Due to polynomial approximation.

(2) Trade-off with privacy

(3) Provided that a shared model can be trained at the cloud, and that clients have enough samples to personalize their local models

(4) Due to Homomorphic encryption.

(5) Due to iterative interaction between client and server

(6) Due to local training

(7)Training round only needs local data to be transferred

(8) However, user's local training is not considered in the global model until transferred

(9) Depends on the server round policy, i.e., wait for late users? Indefinitely, or for a certain period, …etc.

(10) Trade-off with accuracy

(11) Global gradients are not protected

(12) Structure shared

* Privacy guarantees and limitations relative to indirect leakage are distinguished between square brackets []

Table 4 Comparison of key concepts for training a local model

Key concept	Learning	Accuracy	Client overhead	Dropout impact	Data privacy	Local model privacy
HE	IND—COL	**IND**: *loss* **COL**: *trade-off*[a]	*high*	**IND**: *none* **COL**: *high*[a]	**IND**: *private* **COL**: *trade-off*[b]	**IND**: *structure shared* **COL**: *not shared*
PS	COL	*trade-off*	*high*	*none* | *high*[a]	*tradeoff*	*clear global gradients*
TRA	COL	*trade-off*	*high*	*none* | *high*[a]	*tradeoff*	*Model shared*
SM	IND	*depends on local training resources*	*high*	*none*	*private*	*structure shared*

HE Homomorphic encryption, *PS* Partial sharing, *TRA* Transformation, *SM* Shared model, *IND* Individual learning *COL* Collaborative learning, *Dropout impact—COL high*, if the training round stops for all participants; *none*, if the training round is not affected, *IND high,* if the participant training process stops, *none*, if the process does not stop, *trade-off* between accuracy and privacy

[a] in case the coordination or a threshold of participants and/or their transmitted information are required in the process at each round

[b] in case of using perturbation against inter-participants protection

• Discussion

Homomorphic encryption (HE) HE-based solutions are characterized by high overhead at the client-side, especially if fully HE is employed. Besides, they provide high accuracy when activation functions are computed without polynomial approximation [42]. However, this requires the use of alternative techniques like outsourcing the computation to the client, which leads to more client-side overhead. Moreover, if the outsourcing method is used, the training round that is related to a dropped client is stopped until it reconnects. As for privacy, training data and intermediate results are protected through encryption, while the model is not shared with participants except for its activation functions, due to the outsourcing method. In general, HE-based solutions do not make restrictions on deep model or activation function. However, an adaptation of the model to the HE domain is needed as explained in [43].

Partial sharing (PS) PS-based solutions are characterized by high client-side overhead due to local training. They also make a trade-off between accuracy and privacy, which is controlled by the fraction of shared parameters and the level of perturbation. Besides, the impact of a participant's dropout on the training round process is low, and no restrictions are made on the deep model and activation functions. As for privacy, only a fraction of local parameters is revealed to the cloud. However, the remote model needs to be shared with the participants.

Transformation Solutions based on transformations can reach high accuracy. However, they make a trade-off between privacy and accuracy due to the applied perturbations [44]. Moreover, solutions that rely on local training are characterized by high overhead at the client-side, and require to share the remote model with participants. As for the impact of a participant dropout, the solution [35] requires the feedback of all the participants in order to reveal the global gradient, which blocks the training round process.

Table 5 Privacy-preserving solutions for training a remote model

Key concept	Ref	Main characteristics
Homomorphic Encryption	Q. Zhang et al. [45]	Partially Paillier HE | Outsourcing non-linear computations to the client
Partial sharing	Shokri and Shmatikov [32]	Partial sharing of parameters | Laplace differential privacy | Sparse vector technique
	Liu et al. [33]	Partial sharing of parameters
Transformation	Lyu et al. [46]	Repeated Gompertz (RG) for data perturbation | Row-orthogonal random projection (RP) matrix for projecting high-dimensional data to lower dimension
	Zhao et al. [34]	Functional exponential mechanism | Polynomial approximation for objective function | Cryptography and hashing against eavesdrop attacks
	Hartmann and West [35]	Cancelable noise (differential privacy) | Anonymization network (such as Tor)
	Fu et al. [36]	Mixup data augmentation
Model splitting	Yu et al. [47]	1st convolutional layer on local | Step-wise activation functions | CNN
	Abuadbba et al. [48]	Part of layers on local | differential privacy
	Dong et al. [49]	1st layer on local | Dropping connections and activation outputs | Dropout and Dropconnect

Table 6 Solutions for training a remote model *vs* evaluation criteria

Ref	**Effectiveness**	**Efficiency**		**Privacy** guarantees and limitations		**Notes**
	Accuracy	**OVC**	**DIT**	**TD***	**RM**	
[45]	WLS(1)	*High(3)(4)*	*Low*	TD [and intermediate results] encrypted	Only*activation functions shared*	AFR/MOR: tested on DNN and CNN
[32]	CAN *but (2) and criticized in*[29]	*High(4)(5)*	None(7)	TD not shared [and only fraction of local parameters shared perturbed,*but (9)*]	*shared*	ADV:*trusted server*
[33]	*RR ~ 90%, but (2)*	*High(4)(5)*	Round-robin:*High* Asynchronous: None(7)	TD not shared [and only fraction of local parameters shared,*but (9)*]	*shared*	*- can be concerned by the privacy critics reported in*[29, 39–41]
[46]	*< 5% loss & (2) & evaluated using a custom proposed model*	Low communication *but computation needs evaluation*	None(6)	TD perturbed and projected to lower dimension,*but (9)*	not shared	-
[34]	MRE High *but (2)*	*High(4)(5)*	None(7)	TD not shared [and objective functions (thus gradients) perturbed,*but (9)*]	*shared*	ADV:*honest server,* active & passive participants AFR/MOR:*may require adaptations*
[35]	N/A	*High(4)(5)*	*High*	TD not shared [and local gradients perturbed with cancelable noise]	*shared*	ADV: malicious server, *honest participants > = 2* - *anonymization network required*
[36]	Up to High *but (2)*	*High(4)(5)*	[None(7)|*High*](8)	TD not shared [and local parameters obtained from mixup input data,*but (9)*]	*shared*	-
[47]	*Up to Good but (2)*	Low	None(10) | *Low(11)*	TD not shared [and local output perturbed through step-wise local activation function,*but (9)*]	Only*1stlayer shared*	ADV: supports malicious attacks AFR:*step-wise activation functions*
[48]	- With differential privacy (DP):*(2)* - No DP: WLS	- With DP: low - No DP:*depends on local partition*	None(10) | *Low(11)*	- With DP: TD not shared, [and local output perturbed,*but (9)*] - No DP: TD not shared,*but (12)*	*local-side layers shared* - No DP:*(12)*	MOR: 1-dimension CNN
[49]	N/A	Low	None(10) | *Low(11)*	Local outputs protected by droppings*but criticized in* [50]	Only*1stlayer shared*	-

Accuracy mainly on the basis of authors evaluations.

WLS Without Loss, *CAN* can reach Close to/Almost Non-private model, *RR* Reconstruction Rate, *MRE* Mean Relative Error, *OVC* overhead is relative to the non-private model, *DIT* high, if the training round stops for all participants, *low* if the training round stops for only the dropped participant, *none* if the training round is not affected, *ADV* Adversary model, *AFR* Activation Function Restrictions, *MOR* Model Restrictions

(1) No approximation is involved

(2) Trade-off with privacy

(3) Due to Homomorphic encryption

(4) Due to iterative interaction between client and server

(5) Due to local training

(6) Training round only needs local data to be transferred

(7) but user's local training not considered in the global model until transferred

(8) Depends on the server round policy: wait for late users? Indefinitely, for a certain period, …

(9) Trade-off with accuracy

(10) After the local partition has been executed, and local output transferred

(11) If local output has not been transferred

(12) Trade-off between the privacy of the model an the privacy of data

* Privacy guarantees and limitations relative to indirect leakage are distinguished between square brackets []

Table 7 Comparison of key concepts for training a remote model

Key concept	Accuracy	Client overhead	Dropout impact	Data privacy	Remote model privacy
HE	*without loss*	*high*	*low*	*private*	*activation functions shared*
PS	*trade-off*	*high*	*none* | *high*[a]	*trade-off*	*model shared*
TRA	*theoretically without loss*[c] | *trade-off*	*high*[b] | *low*	*high*[a] | *low*	*private*[c] | *trade-off*	*model shared*[b] | *not shared*
MS	*trade-off*	*low*	*low*[d]	*trade-off*	*part of layers shared*

HE Homomorphic encryption, *PS* Partial sharing, *TRA* Transformation, *MS* Model splitting,*Dropout impact—high*, if the training round stops for all participants, *low* if it stops only for the dropped participant, *none* if it is not affected, *trade-off:* between accuracy and privacy

[a] in case the coordination or a threshold of participants and/or their transmitted information are required in the process at each round

[b] in case of distributed (federated) learning

[c] in case perturbation is cancelable

[d] if the local output has not been transferred, otherwise *none*

Model splitting The client-side overhead, under the model splitting concept, depends on the local model partition depth and complexity. In [47], only the first convolutional layer is migrated to the client, which ensures a low client-side overhead. The solution also ensures a low impact of a participant's dropout on the training process. However, to preserve privacy, perturbation is applied on the local output, which makes a trade-off between accuracy and privacy. Model privacy is ensured partially, and depends on the model partition depth migrated to the client-side. Generally, model splitting solutions do not make restrictions on the deep model. However, activation functions in [47] require to be step-wise in order to perturb the client-side output and preserve privacy, while some approaches focus on specific models such as [48], which addressed 1-dimension CNN models.

Key concepts comparison By going through the reviewed solutions, we summarize in Table 7 the key concepts adequacy with the PHM environment mainly with respect to effectiveness, client-side efficiency and privacy guarantees. The main potential limitations are underlined in the table.

It is observed that HE-based reviewed solutions show the best overall performances among other concepts. In fact, HE solutions can meet most of the PHM requirements, except for the client-side overhead which is still challenging. Moreover, it is also observed that all concepts require a trade-off between accuracy and privacy, except for HE that can ensure both high accuracy and privacy. However, if the perturbation introduced under TRA concept is cancelable, data privacy can also be ensured without comprising accuracy, but such solutions (under federated learning) suffer from a high client-side overhead due to local training. Besides, PS and TRA (under federated learning) concepts do not consider model privacy. On the other hand, HE concept only needs to share the used activation functions, while MS solutions require to migrate a part of the model layers to client, which may incur in some solutions a trade-off between model privacy and data privacy. TRA solutions not relying on federated learning successfully keep the remote model private.

5.3 Privacy-preserving remote inference

Table 8 summarizes the main characteristics of the reviewed privacy-preserving solutions for remote inference, which are classified according to four main technologies, representing the key base concepts that are used by these solutions, namely, homomorphic encryption (HE), secure multi-party computation (SMC), transformation of sensitive data, and model splitting between the client and the remote side.

Table 9 evaluates the solutions against the criteria defined in Sect. 4.3. The table cells, that are highlighted in gray, show the main criteria that are not fully satisfied, along with their limitations that are underlined.

- **Discussion**

Homomorphic encryption (HE) HE-based solutions are characterized by high overhead at the client-side, especially if fully HE is employed. Moreover, solutions, which rely on the the client participation to address HE noise growth,[1] further increase the client-side overhead. As for effectiveness, HE-based solutions can reach close and up to the same accuracy

[1] "*When operations such as addition and multiplication are applied to encrypted data, the noise in the result may be larger than the noise in the inputs; this is referred to as noise growth*". If this noise grows too much, the ciphertext becomes impossible to decrypt even using the correct private key [61].

Table 8 Privacy-preserving solutions for remote inference

Key concept	Ref	Main characteristics
Homomorphic Encryption	Gilad-Bachrach et al. [23]	Leveled YASHE HE | Polynomial approximation of activation function
	Baryalai et al. [26]	Partially Paillier HE | Non-colluding dual clouds | Diffie-Hellman key exchange | Random salt | Classification
	Chabanne et al. [51]	Fully BGV HE | Low degree polynomial approximation of activation function | Batch normalization | Classification | CNN with depth > 2
	Hesamifard et al. [52, 53]	Leveled HE | Polynomial approximation: derivative of ReLU based approach and Sigmoid, Tanh, over a symmetric interval | CNN
	Zhu and Lv [43]	Partially Paillier HE | Interactive protocol between client and server for ReLU computation
	Vizitiu et al. [54]	Fully MORE HE
SMC	Huang et al. [55]	Additive secret-sharing | Secure computations | CNN feature extractor | Non-colluding dual edge servers
	Ma et al. [56]	Secret sharing | Partially El Gamal HE | Low-degree polynomial approximation of activation function | Non-colluding dual servers
	Li et al. [57]	Secret sharing | Triplet generation | Fully YASHE HE | Two non-colluding servers | Asynchronous computation | Garbled circuits | CNN
Transformation	Leroux et al. [58]	Generative Adversarial Networks | Neural-network-based obfuscation
	Raval et al. [21]	Generative Adversarial Networks | Neural-network-based obfuscation
	Xu et al. [16]	Neural-network-based obfuscation
Model splitting	Osia et al. [59]	Feature extractor on local | Siamese architecture | Dimensionality reduction: PCA and auto-encoder | Symmetric gaussian noise
	Chi et al. [60]	Bipartite model | Interactive adversarial deep networks
	Yu et al. [47]	1st convolutional layer on local | Step-wise activation functions | CNN
	Dong et al. [49]	1st layer on local | Dropping connections and activation outputs | Dropout and Dropconnect

as the non-private models when activation functions are computed without polynomial approximation [42]. However, this requires the use of alternative techniques like outsourcing the computation to the client, which leads to more client-side overhead. Moreover, as these methods rely on the client participation, the impact of a client dropout on the inference process becomes high. In [26], a non-colluding two-servers architecture is introduced to mitigate the client overhead by delegating the computation of activation functions to an intermediate server. However, the client is only partially discharged from the cryptographic operations. As for privacy, input data are protected from the cloud through encryption, while the model is not shared with the clients, except for its activation functions, due to the outsourcing method. In general, HE-based solutions do not make restrictions on deep model or activation function. However, some solutions focused on the CNN model, while some others addressed specific activation functions for which they investigated polynomial approximations. Besides, an adaptation of the model to the HE domain is needed as explained in [43].

Secure Multiparty Computation (SMC) SMC-based solutions can ensure low impact of client dropout on the inference process, and preserve accuracy without incurring high overhead at the client-side. As for privacy, input data and inferences are protected from the servers, and the remote model is not shared with the clients. However, SMC-based solutions require the composition of adapted layers for the different phases of the neural network, while the model needs to be hosted partially or totally on both servers. Moreover, some solutions require using a trustworthy third party to initialize the random shares. In [56], HE is introduced at the client-side, to encrypt input data instead of splitting it into shares, which allows to eliminate the trust initializer. However, encryption increases the client-side overhead, and leads to the use of polynomial approximation of activation functions, which may incur accuracy loss.

Transformation Transformation-based solutions ensure that once the client obfuscates its data and transmits it, its dropout will not impact the inference process. However, in [21], up to 17% of accuracy loss was incurred, which shows that obfuscating input data may have a high impact on accuracy. Moreover, the overhead at the client-side might be high, as it depends on the obfuscator network and its output. As for privacy, inferences are not protected, and input data, although obfuscated, could allow leakage. In [21] for example, close to 17% accuracy of inferring private information could be reached.

Table 9 Remote inference solutions *vs* evaluation criteria

Ref	Effectiveness	Efficiency		Privacy guarantees and limitations			Notes
	Accuracy	**OVC**	**DII**	**IN***	**IF**	**MO**	
[23]	Up to 99%, *but very low if non-linear layers > 2*	*High(5)*	None	encrypted	encrypted	not shared	AFR: *use the square function (the lowest-degree non-linear polynomial function)*
[26]	N/A	*High(5)*	None	IN encrypted, [*but not intermediate results*[46]]	Encrypted, masked with random salt	not shared	AFR: *softmax (last layer)* - *non-colluding dual clouds required* - *random salt mechanism not detailed*
[51]	CTN	*High(5)*	None	encrypted	encrypted	not shared	AFR: *APX- new learned ReLU approximation* MOR: focus on CNN - *training is required after adding batch normalization layer*
[52]	CTN *but potential loss(1)*[57]	*High(5)(6,11)*	None | *High(6,11)*[50]	encrypted	encrypted	not shared	AFR: *APX—ReLU, sigmoid, tanh with new investigated approximations* MOR: focus on CNN
[43]	CTN	*High(5)(6)*	*High*	encrypted	encrypted	not shared	AFR: *ReLU* MOR: focus on CNN
[54]	Up to ITN	*High(5)*	None	encrypted	encrypted	not shared	- *Train over encrypted data* - *MORE scheme is weak against chosen plain-text attacks*
[55]	WLS	Low	None	encrypted: split into two shares	encrypted into two shares	*held by both edge servers*	ADV: *third party honest and trusted* AFR: focus on ReLU MOR: *CNN, and composed adapted layers* - *Edge servers independent/ non-colluding*
[56]	N/A *but potential loss*(1)	*High(5)*	None	encrypted	encrypted into two shares	*each server holds a share of the model*	AFR: *APX* + focus on ReLU, sigmoid, and tanh - *Cloud servers non-colluding*
[57]	Preserves high accuracy	Low	None	encrypted: split into two shares	encrypted into two shares	*each server holds a share of the model*	AFR/MOR: focus on CNN *with special designed protocols* - *Special designed garbled circuits for SMC*

Table 9 (continued)

Ref	Effectiveness	Efficiency		Privacy guarantees and limitations			Notes
	Accuracy	**OVC**	**DII**	**IN***	**IF**	**MO**	
[58]	*Accuracy loss (~5%)*	*DOB*	None	obfuscated	*not protected*	*obfuscator need to back-propagate through the main model*	- obfuscator and deobfuscator trained competitively - *obfuscator retraining is required in case the main model is non static (retrained)* - *obfuscation of multi-modal data might need more investigation as in* [19]
[21]	*Accuracy loss (maximum drop of 17%) and (4)*	*DOB*	None	obfuscated *(accuracy of inferring private info < 17%) and (9)*	*not protected*	not shared	- obfuscator and deobfuscator trained competitively, and with the help of of the main classifier - obfuscation of multi-modal data needs investigation
[16]	*Accuracy drop* (evaluations showed a max. of less than 3%, *but (2)(3)*)	*DOB*, but the overhead was considered as low by authors	None	obfuscated *(each edge device receives a unique set of obfnets)*	*not protected*	not shared	ADV: edge devices *honest*, but may collude with backend AFR: *many-to-one activation functions* - *obfnets trained at backend (thus known) while concatenated to the inference model, and using original training data* - *number of obfnets can be huge for large number of users*
[59]	Acceptable (according to authors) *but (4)*	*DLP* [16] *estimated the computation overhead as considerable*	None(7) | *High(8)*	IN not shared, [and fine-tuned features are output instead, *but (9)* although considered acceptable by authors]	*not protected*	*feature extraction layers shared*	MOR: focus on CNN, *RNN is for eg. set as a future direction* - *Model needs to be fine-tuned using Siamese*
[60]	Up to high *but (4)*	*DLP*	None(7) | *High(8)*	IN not shared, [and reversibility of local output strengthened, *but (9)*]	*not protected*	*local-side layers shared, and (10)*	ADV: adversary can have access to the remote party, and intermediate states - Bipartite model needs to be trained concurrently with the defender

Table 9 (continued)

Ref	Effectiveness	Efficiency		Privacy guarantees and limitations			Notes
	Accuracy	**OVC**	**DII**	**IN***	**IF**	**MO**	
[47]	Up to good *but (4)*	Low	None(7) | *High(8)*	IN not shared, [and local output from step-wise activation function, *but (9)*]	*not protected*	Only *1st layer shared*	ADV: supports malicious attacks AFR: *Output needs to be perturbed*
[49]	No noticeable loss	Low	None(7) | *High(8)*	IN not shared, [and local output protected by a dropping strategy, *but criticized in* [53]]	*not protected*	Only *1st layer shared*	-

Accuracy mainly on the basis of authors evaluations

WLS Without Loss, *CTN* Close To Non-private model, *ITN* Identical To Non-private model, *OVC* overhead is relative to the non-private model, *DOB* Depends on the obfuscator network and its output, *DLP* Depends on the local partition, *DII* high, if the inference process stops, *none* if the process does not stop, *ADV* Adversary model, *AFR* Activation Function Restrictions, *MOR* Model Restrictions, *APX* Approximated

(1) Due to polynomial approximation of activation functions

(2) Tested using only non-heavyweight inference models

(3) Different obfnets may achieve different accuracy results, the effectiveness can be non-stable as it may differ from user to user or from an obfnet to another for a same user

(4) Trade-off with privacy

(5) Due to Homomorphic encryption

(6) Due to iterative interaction between client and server

(7) After the local partition has been executed, and local output transferred

(8) If the local output has not been transferred

(9) Trade-off with accuracy

(10) Trade-off between the privacy of the model an the privacy of data

(11) When addressing noise growth

* Privacy guarantees and limitations relative to indirect leakage are distinguished between square brackets []

Table 10 Comparison of key concepts for remote inference

Key concept	Accuracy	Client overhead	Dropout impact	Data privacy	Inference privacy	Model privacy
HE	*high but potential loss*[a]	*high*	*none* | *high*[c]	*IR shared*[f] | *private*	*private*	*not shared*
SMC	*high but potential loss*[a]	*low* | *high*[b]	*none*	*private*	*private*	*model (or a share of it) shared with the 2nd server*
TRA	*loss*	*depends on obfuscator network*	*none*	*private but potential trade-off*	*not protected*	*not shared* | *back-propagated*[e]
MS	*trade-off*	*depends on local partition*	*high*[d]	*trade-off*	*not protected*	*local partition*

Dropout impact - high, if the inference process stops; none, if the process does not stop. trade-off: between accuracy and privacy

HE Homomorphic encryption, *SMC* Secure Multiparty Computation, *TRA* Transformation, *MS* Model splitting

[a] if activation function are approximated and depending on the polynomial approximation use

[b] if homomorphic encryption is used

[c] in case refreshing noise and/or the computation of activation functions are performed by the client

[d] if the local output has not been transferred, otherwise non

[e] in case of obfuscators that need during their training to back-propagate through the main mode

[f] in order to compute activation functions, Intermediate Results (IR) are shared with the introduced 2nd server without protection

Model splitting Under the model splitting concept, the client-side overhead and sometimes data privacy, depend on the local model partition depth and complexity. In fact, in [47], only the first convolutional layer is migrated to the client, which ensures a low client-side overhead. However, in [59], the client-side computation overhead was described as considerable [16], due to a more complex local partition, representing the feature extractor. In model splitting-based solutions, the dropout of a participant does not impact the inference process, once the local output is transmitted to the server-side. As for privacy preservation, techniques like perturbation of the local output [47] or adversarial training [59] are used, which makes a trade-off between accuracy and privacy. Moreover, inferences protection is not considered.

Key concepts comparison By going through the reviewed solutions, we summarize in Table 10 the key concepts adequacy with the PHM environment mainly with respect to effectiveness, client-side efficiency and privacy guarantees. The main potential limitations are underlined in the table.

Similarly to remote model training scenario, it is observed that HE-based solutions generally meet almost all PHM requirements, except for the client-side overhead, which is still challenging. SMC concept is also promising, but requires to share the model (or part of it) to a non-colluding second server. Moreover, the use of HE for privacy purposes in some SMC-based solutions leads to a high client-side overhead. Besides, TRA and MS concepts still need to address a number of challenges in order to support PHM environment. In fact, the two concepts do not consider inferences protection, while data privacy might be in trade-off with the accuracy. Moreover, the remote model needs to be partially migrated to the client in MS solutions, while some TRA-based solutions require to have access to the whole model.

6 Open research

Many efforts were deployed in order to design solutions for privacy-preserving deep learning. However, many of the existing solutions do not consider specific target environment constraints. As previously discussed, in the context of pervasive health monitoring, the different key concepts of privacy preservation require more investigation in order to address the identified limitations and cope with the client-constrained environment.

This section outlines, for each key concept, a set of recommendations for future research directions within each of the privacy-preserving deep-learning-based scenarios of PHM.

6.1 Privacy-preserving training of a local model

Two main future investigation paths can be recommended in order to optimize Homomorphic Encryption-based solutions to the PHM environment in a training of a local model scenario:

- The mitigation of the client-side overhead in terms of computation and communication incurred by the heavy

cryptographic operations and local training. Investigated solutions should take into consideration accuracy preservation and privacy of both data and model.
- The protection of the deep model structure in the individual training scenario.

As for Partial sharing and Transformation approaches, two common open research paths might be followed:

- The improvement of local training, or the introduction of alternatives methods, in order to mitigate the client-side overhead without compromising the privacy of local data and model.
- The improvement of the trade-off between accuracy and privacy using more efficient perturbations that can combine high accuracy and strong sensitive data protection. In this direction, proposed perturbations need to be adaptive to the input data type of the target application, and consider potential heterogeneity of medical data.

However, transformation-based solutions require to reveal the model to the cloud, which may represent a serious limitation in the local model training scenario if the privacy of the model is important. On another hand, the protection of aggregated model parameters from the cloud need to be investigated in partial sharing solutions.

Finally, as the shared model concept mainly relies on local fine tuning, further investigations are recommended in order to introduce more efficient methods in terms of the client-side overhead. Such methods need also to take into consideration the privacy of both local samples and the model. Moreover, due to the cloud-based training step, the structure of the deep model is shared between the client and cloud, which may constitute a serious issue if its privacy is considered as important.

6.2 Privacy-preserving training of a remote model

Two main open research paths need to be investigated in HE-based solutions for training a remote model under the PHM environment:

- The mitigation of the client-side overhead in terms of computation and communication incurred by the heavy cryptographic operations, without compromising the privacy of local data.
- The introduction of approximation-free techniques for the computation of activation functions, and which do not rely on the client-side, and do not compromise the privacy of the local data and remote model.

Partial sharing and Transformation approaches require some improvements, particularly:

- The improvement of local training, or the introduction of alternatives methods, in order to mitigate the client-side overhead and provide a certain level of privacy to the remote model, without compromising the privacy of local data.
- The improvement of the trade-off between accuracy and privacy using more efficient perturbations that can combine high accuracy and strong sensitive data protection. In this direction, proposed perturbations need to be adaptive to the input data type of the target application, and consider potential heterogeneity of medical data.

As for the model splitting approach, a more efficient trade-off that could balance between the different requirements of the PHM environment need to be investigated. More specifically, such a trade-off needs to consider:

- The client-side overhead, controlled by the local partition depth and the perturbation complexity.
- The privacy of training data, controlled by the local partition depth, and the perturbation effectiveness.
- The privacy of the deep model, controlled by the local partition depth.
- The accuracy, controlled by the impact of the perturbation.

6.3 Privacy-preserving remote inference

Two main open research paths are recommended to address the limitations of Homomorphic Encryption-based privacy-preserving solutions under the PHM environment:

- The mitigation of the client-side overhead in terms of computation and communication incurred by the heavy cryptographic operations, without compromising the privacy of local data. In this context, the non-colluding two-servers architecture used in [26] should be more investigated in combination with other mechanisms.
- The introduction of approximation-free techniques for the computation of activation functions, as well as techniques for addressing homomorphic noise growth, which do not rely on the client-side. Moreover, introduced techniques should take into consideration the privacy of the remote model and local data, including preventing the intermediate results from leaking sensitive information.

As for Secure Multiparty Computation-based solutions, a set of identified limitations need to be addressed in order to cope with the target environment, mainly:

- The model is either split or shared between the non-colluding servers, which compromises its privacy.

- A number of modifications are necessary at the different stages of the neural network in order to adapt it to the SMC approach.
- Some solutions require a trust initializer in order to split the input data into shares. Others rely on HE encryption at the client-side, which increases the local overhead.

As for transformation-based solutions, future investigation directions include:

- The design of more efficient obfuscator networks, or transformation methods that can combine high accuracy, strong privacy, and low local overhead.
- The design of methods that protect inferences from the cloud.

Finally, as for the model splitting-based solutions, two main open research paths can be recommended:

- Investigating a more efficient trade-off that could balance between the different requirements of the target environment, considering:
  - The client-side overhead, controlled by the local partition depth and the local privacy-preserving method (method applied on local to prevent leakage) complexity.
  - The privacy of the input data, controlled by the local partition depth, and the local privacy-preserving method effectiveness.
  - The privacy of the model, controlled by the local partition depth.
  - The accuracy, controlled by the impact of the local privacy-preserving method.
- Designing solutions that protect inferences from the cloud.

7 Conclusion

This paper studies the adequacy of existing privacy-preserving deep learning solutions to pervasive heath monitoring (PHM) applications. To this end, privacy-requiring scenarios are defined, and a number of recent solutions for privacy-preserving deep learning are discussed according to criteria derived from constraints of the environment and requirements of the target solution.

The analysis of the PHM deep learning-based scenarios shows that the inference phase as well as the training phase, including training local and remote models, are all subject to privacy concerns. In order to design privacy-preserving solutions for PHM, the following specific constraints of the environment need to be taken into consideration: (*a*) the client-side devices in terms of limited resources, (*b*) input data in terms of heterogeneity, and (*c*) communication network in terms of unreliability and high cost.

Accordingly, in order to assess privacy-preserving deep learning solutions with the PHM environment, the following derived set of criteria are defined: (*a*) *effectiveness*, in terms of high accuracy, (*b*) *efficiency*, in terms of low computation and communication overhead at the client-side, as well as the impact of a client's dropout on the training round or inference process, and (*c*) *privacy*, in terms of the protection of input data, deep model, and inferences.

Existing solutions are subsequently classified according to key concepts, and evaluated against defined criteria. The evaluation study and the impact of introducing privacy to deep learning-based PHM applications are then discussed.

We summarize the main findings and conclusions of the present study, according to the privacy-requiring scenarios, as follows:

- **Local model training** HE-based solutions in individual learning do not protect the deep model structure, and incur high client-overhead due to the cryptographic operations. In collaborative learning, the adequacy of privacy-preserving solutions to the PHM environment is particularly restricted by local training and its impact on the client-side overhead, which is furthermore increased in HE-based solutions. Future alternatives or optimizations of local training and HE operations need to be investigated, taken into consideration the privacy of both local model and training data. Moreover, solutions based on transformation and partial-sharing concepts need to investigate more efficient perturbations mechanisms in order to improve the trade-off made between accuracy and privacy.
- **Remote model training** Mitigating the client-side overhead in HE-based solutions require the introduction of new techniques for the computation of activation functions, as well as new methods to discharge the client from the cryptographic load. In transformation-based solutions, efficient alternatives or optimizations of local training need to be investigated. On the other hand, the trade-off made between data privacy and accuracy in transformation as well as model splitting-based solutions depends on the target application requirements. In transformation-based solutions, more efficient perturbations considering the heterogeneity of medical data need to be investigated. Ultimately, future HE-based solutions might be most likely to suit the PHM environment, knowing moreover that current solutions can provide high accuracy and privacy without making a trade-off between them.
- **Remote inference** HE-based solutions suffer from a high client-side overhead, which can be mitigated by the

introduction of new techniques for the computation of activation functions, as well as new methods to discharge the client from the cryptographic load. As these solutions can provide high accuracy and privacy without making a trade-off between them, future HE-based solutions might be most likely to suit the PHM environment. SMC-based solutions, on the other hand, present some limitations essentially regarding the privacy of the deep model, and the need of a trustworthy initializer. Lastly, solutions based on transformation and model splitting concepts do not provide protection for the inferences. Moreover, they make a trade-off between different parameters, combining privacy, accuracy, and efficiency, and which may depend on the target application requirements.

Funding No funding.

Data availability Not applicable.

Code availability Not applicable.

Declarations

Conflicts of interest Authors declare that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Research involving human and animal participants Additional declarations for articles in life science journals that report the results of studies involving humans and/or animals.

References

1. Yao ZJ, Bi J, Chen YX. Applying deep learning to individual and community health monitoring data: A survey. Int J Autom Comput. 2018;15(6):643–55.
2. AI Aids DOD in Early Detection of COVID-19, https://www.defense.gov/Explore/News/Article/Article/2356086/ai-aids-dod-in-early-detection-of-covid-19/. (Last access: 11/2/2021)
3. Fedorin I, Slyusarenko K, Nastenko M. Respiratory events screening using consumer smartwatches. In Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers 2020;25–28.
4. Jindal, V. Integrating mobile and cloud for PPG signal selection to monitor heart rate during intensive physical exercise. In Proceedings of the International Conference on Mobile Software Engineering and Systems 2016;36–37.
5. Zhao P, Quan D, Yu W, Yang X, Fu X. Towards deep learning-based detection scheme with raw ECG signal for wearable telehealth systems. In2019 28th International Conference on Computer Communication and Networks (ICCCN) 2019 Jul 29 (pp. 1-9). IEEE.
6. Hassantabar S, Stefano N, Ghanakota V, Ferrari A, Nicola GN, Bruno R, Marino IR, Hamidouche K, Jha NK. Coviddeep: Sars-cov-2/covid-19 test based on wearable medical sensors and efficient neural networks. IEEE Transactions on Consumer Electronics. 2021. arXiv preprint.
7. Otoom M, Otoum N, Alzubaidi MA, Etoom Y, Banihani R. An IoT-based framework for early identification and monitoring of COVID-19 cases. Biomedical signal processing and control. 2020;62:102149.
8. Imran A, Posokhova I, Qureshi HN, Masood U, Riaz MS, Ali K, John CN, Hussain MI, Nabeel M. AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. Informatics in Medicine Unlocked. 2020. arXiv preprint.
9. Boulemtafes A, Khemissa H, Derki MS, Amira A, Djedjig N. Deep learning in pervasive health monitoring, design goals, applications, and architectures: An overview and a brief synthesis. Smart Health. 2021;22:100221.
10. Boulemtafes A, Derhab A, Challal Y. A review of privacy-preserving techniques for deep learning. Neurocomputing. 2020;384:21–45.
11. Zhang D, Chen X, Wang D, Shi J. A survey on collaborative deep learning and privacy-preserving. In 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC) 2018.
12. Chang S, Li C. Privacy in Neural Network Learning: Threats and Countermeasures. IEEE Network. 2018;32(4):61–7.
13. Tanuwidjaja HC, Choi R, Kim K. A survey on deep learning techniques for privacy-preserving. InInternational Conference on Machine Learning for Cyber Security 2019 Sep 19 (pp. 29-46). Springer, Cham.
14. Riazi MS, Rouani BD, Koushanfar F. Deep learning on private data. IEEE Security & Privacy. 2019.
15. Zheng M, Xu D, Jiang L, Gu C, Tan R, Cheng P. Challenges of privacy-preserving machine learning in IoT. In Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things 2019 Nov 10 (pp. 1-7).
16. Xu D, Zheng M, Jiang L, Gu C, Tan R, Cheng P. Lightweight and unobtrusive data obfuscation at IoT edge for remote inference. IEEE Internet of Things J. 2020. arXiv preprint.
17. Varshney U. Pervasive healthcare and wireless health monitoring. Mobile Networks and Applications. 2007;12(2–3):113–27.
18. Huzooree G, Kumar Khedo K, Joonas N. Pervasive mobile healthcare systems for chronic disease monitoring. Health Informatics J. 2019;25(2):267–91.
19. Banerjee A, Verma S, Bagade P, Gupta SK. Health-dev: Model based development pervasive health monitoring systems. In 2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks 2012;85–90
20. Chowdhury MA, Light J, McIver W. A framework for continuous authentication in ubiquitous environments. In 2010 Sixth International conference on Wireless Communication and Sensor Networks 2010 Dec 15 (pp. 1-6). IEEE.
21. Raval N, Machanavajjhala A, Pan J. Olympus: Sensor Privacy through Utility Aware Obfuscation. Proceedings on Privacy Enhancing Technologies. 2019;2019(1):5–25.
22. Chen C, Zhang P, Zhang H, Dai J, Yi Y, Zhang H, Zhang Y. Deep learning on computational-resource-limited platforms: a survey. Mobile Information Systems. 2020;2020.
23. Gilad-Bachrach R, Dowlin N, Laine K, Lauter K, Naehrig M, Wernsing J. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In International conference on machine learning 2016.
24. Prapas I, Derakhshan B, Mahdiraji AR, Markl V. Continuous Training and Deployment of Deep Learning Models. Datenbank-Spektrum. 2021;21(3):203–12.

25. Pianykh OS, Langs G, Dewey M, Enzmann DR, Herold CJ, Schoenberg SO, Brink JA. Continuous learning AI in radiology: implementation principles and early applications. Radiology. 2020;297(1):6–14.
26. Baryalai M, Jang-Jaccard J, Liu D. Towards privacy-preserving classification in neural networks. In 2016 14th annual conference on privacy, security and trust (PST) 2016.
27. Zhang Q, Yang LT, Chen Z. Privacy Preserving Deep Computation Model on Cloud for Big Data Feature Learning. IEEE Trans Comput. 2016;65(5):1351–62.
28. Bu F, Ma Y, Chen Z, Xu H. Privacy preserving back-propagation based on BGV on cloud. In 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems 2015.
29. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-Preserving Deep Learning: Revisited and Enhanced. Applications and Techniques in Information Security Communications in Computer and Information Sci. 2017;100–110.
30. Zhang X, Ji S, Wang H, Wang T. Private, yet practical, multiparty deep learning. In2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS) 2017.
31. Hao M, Li H, Xu G, Liu S, Yang H. Towards efficient and privacy-preserving federated deep learning. InICC 2019-2019 IEEE International Conference on Communications (ICC) 2019 May 20 (pp. 1-6). IEEE.
32. Shokri R, Shmatikov V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security 2015.
33. Liu M, Jiang H, Chen J, Badokhon A, Wei X, Huang MC. A collaborative privacy-preserving deep learning system in distributed mobile environment. In 2016 International Conference on Computational Science and Computational Intelligence (CSCI) 2016.
34. Zhao L, Wang Q, Zou Q, Zhang Y, Chen Y. Privacy-preserving collaborative deep learning with unreliable participants. 2019. *arXiv preprint*.
35. Hartmann V, West R. Privacy-preserving distributed learning with secret gradient descent. 2019. *arXiv preprint*.
36. Fu Y, Wang H, Xu K, Mi H, Wang Y. Mixup based privacy preserving mixed collaboration learning. In 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE) 2019 Apr 4 (pp. 275-2755). IEEE.
37. Servia-Rodríguez S, Wang L, Zhao JR, Mortier R, Haddadi H. Privacy-preserving personal model training. In 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI). IEEE. arXiv preprint, 2017.
38. Duan J, Zhou J, Li Y. Privacy-Preserving distributed deep learning based on secret sharing. Inf Sci. 2020;527:108–27.
39. Phan N, Wu X, Hu H, Dou D. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In 2017 IEEE International Conference on Data Mining (ICDM) 2017 Nov 18 (pp. 385-394). IEEE. arXiv preprint 2017.
40. Phan N, Wu X, Dou D. Preserving differential privacy in convolutional deep belief networks. Mach Learn. 2017;106(9–10):1681–704.
41. Papernot N, Abadi M, Erlingsson U, Goodfellow I, Talwar K. Semi-supervised knowledge transfer for deep learning from private training data. arXiv preprint 2016.
42. Vizitiu A, Niţă CI, Puiu A, Suciu C, Itu LM. Applying deep neural networks over homomorphic encrypted medical data. Computational and mathematical methods in medicine. 2020.
43. Zhu Q, Lv X. 2p-dnn: Privacy-preserving deep neural networks based on homomorphic cryptosystem. arXiv preprint arXiv:1807.08459. 2018. arXiv preprint.
44. Yin X, Zhu Y, Hu J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Computing Surveys (CSUR). 2021;54(6):1–36.
45. Zhang Q, Wang C, Wu H, Xin C, Phuong TV. GELU-Net: A Globally Encrypted, Locally Unencrypted Deep Neural Network for Privacy-Preserved Learning. In IJCAI 2018 Jul 13 (pp. 3933-3939).
46. Lyu L, He X, Law YW, Palaniswami M. Privacy-preserving collaborative deep learning with application to human activity recognition. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management 2017 Nov 6 (pp. 1219-1228).
47. Yu CH, Chou CN, Chang E. Distributed layer-partitioned training for privacy-preserved deep learning. In 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) 2019 Mar 28 (pp. 343-346). IEEE.
48. Abuadbba S, Kim K, Kim M, Thapa C, Camtepe SA, Gao Y, Kim H, Nepal S. Can we use split learning on 1d cnn models for privacy preserving training?. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security 2020 Oct 5 (pp. 305-318).
49. Dong H, Wu C, Wei Z, Guo Y. Dropping activation outputs with localized first-layer deep network for enhancing user privacy and data security. IEEE Transactions on Information Forensics and Security. 2017 Oct 17;13(3):662-70.
50. Tan X, Li H, Wang L, Xu Z. Comments on "Dropping Activation Outputs with Localized First-Layer Deep Network for Enhancing User Privacy and Data Security." IEEE Trans Inf Forensics Secur. 2020;15:3938–9. https://doi.org/10.1109/TIFS.2020.2988156.
51. Chabanne H, De Wargny A, Milgram J, Morel C, Prouff E. Privacy-preserving classification on deep neural network. Cryptology ePrint Archive. 2017.
52. Hesamifard E, Takabi H, Ghasemi M. Cryptodl: Deep neural networks over encrypted data. arXiv preprint 2017.
53. Hesamifard E, Takabi H, Ghasemi M, Jones C. Privacy-preserving machine learning in cloud. InProceedings of the 2017 on cloud computing security workshop 2017.
54. Vizitiu A, Niţă CI, Puiu A, Suciu C, Itu LM. Towards privacy-preserving deep learning based medical imaging applications. In 2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA) 2019 Jun 26 (pp. 1-6). IEEE.
55. Huang K, Liu X, Fu S, Guo D, Xu M. A lightweight privacy-preserving CNN feature extraction framework for mobile sensing. IEEE Transactions on Dependable and Secure Computing. 2019.
56. Ma X, Chen X, Zhang X. Non-interactive privacy-preserving neural network prediction. Inf Sci. 2019;481:507–19.
57. Li M, Chow SS, Hu S, Yan Y, Chao S, Wang Q. Optimizing privacy-preserving outsourced convolutional neural network predictions. IEEE Transactions on Dependable and Secure Computing. 2020. arXiv preprint.
58. Leroux S, Verbelen T, Simoens P, Dhoedt B. Privacy aware offloading of deep neural networks. 2018. arXiv preprint.
59. Osia SA, Shamsabadi AS, Sajadmanesh S, Taheri A, Katevas K, Rabiee HR, Lane ND, Haddadi H. A hybrid deep learning architecture for privacy-preserving mobile analytics. IEEE Internet of Things J. 2020;7(5):4505-18.
60. Chi J, Owusu E, Yin X, Yu T, Chan W, Tague P, Tian Y. Privacy partitioning: Protecting user data during the deep learning inference phase. arXiv preprint arXiv:1812.02863. 2018. arXiv preprint.
61. Chen H, Gilad-Bachrach R, Han K, Huang Z, Jalali A, Laine K, Lauter K. Logistic regression over encrypted data from fully homomorphic encryption. BMC Med Genomics. 2018;11(4):3–12.