Decentralized Neural Network Modeling from Heterogeneous Data Sources: A Feature Mapping Approach

Kwabena Frimpong Marfo
University of Silesia in Katowice
Katowice, Poland *kwabena.marfo@us.edu.pl*

Małgorzata Przybyła-Kasperek
University of Silesia in Katowice
Katowice, Poland
Constantine the Philosopher University in Nitra
Nitra Slovakia *malgorzata.przybyla-kasperek@us.edu.pl*

Abstract

This paper presents a privacy-preserving framework for distributed neural network modeling across heterogeneous data sources, where local datasets differ in both objects and attributes. To enable collaborative learning without sharing raw data or model parameters, each local decision table is independently transformed into a unified feature space using multiple dimensionality reduction techniques – Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and Uniform Manifold Approximation and Projection (UMAP). Various types of neural networks – Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Simple Recurrent Network (SIMPLE), Multilayer Perceptron (MLP) and the Radial Basis Function Network (RBF) – are trained locally, and their outputs are aggregated using soft voting (simple average) to generate final predictions. Experimental results on benchmark datasets confirm the approach's effectiveness, scalability, and robustness in decentralized learning settings.

Keywords: decentralized learning, heterogeneous data, feature transformation, neural networks, dispersed data.

1. Introduction

Modern machine learning increasingly operates in distributed environments, such as hospitals banks or research institutions, where data is siloed across different locations due to privacy, security or regulatory constraints. A growing challenge in such settings is the need to learn predictive models from fully heterogeneous data, where each local repository stores its own dataset with unique features and distinct records. These locally stored datasets – hereafter referred to as local tables – often differ not only in their content but also in structure, rendering traditional distributed or federated learning approaches inadequate.

Federated learning addresses some aspects of decentralized data by enabling collaborative model training without sharing raw data. However, it generally assumes a shared feature space across all local datasets and requires parameter synchronization during training. These assumptions break down in scenarios where local tables are structurally different, such as hospitals recording patient data with differing diagnostic tools, or environmental sensors deployed in diverse geographic regions. In such real-world cases, a more flexible and privacy-preserving approach is required.

This paper introduces a novel framework for decentralized neural network modeling over structurally heterogeneous data sources. Each local table is independently transformed into a compatible feature space using dimensionality reduction techniques, enabling interoperability without exposing original data structures. Unlike conventional federated learning, our method avoids

parameter exchange and ensures that the architecture and internal workings of local models remain opaque, reducing vulnerability to reverse-engineering or inference attacks.
The framework is general and supports a variety of transformation techniques. In this study, we use PCA, SVD and UMAP to project each local table into a k-dimensional feature space. Independent neural network models are trained locally using different architectures – MLP, SIMPLE, GRU, LSTM, and RBF networks. Their predictions are then aggregated via soft voting (simple average), avoiding raw data exchange and model synchronization.
We evaluate the proposed method on benchmark datasets from the UCI Machine Learning Repository, which were artificially partitioned into non-overlapping, heterogeneous local tables. Results demonstrate the scalability, robustness, and privacy-preserving characteristics of our approach, making it suitable for multi-institutional collaboration in sensitive domains.
The main contributions of this work are as follows:

1. A novel framework for decentralized learning over fully heterogeneous data, addressing structural disparities that traditional federated learning cannot handle.

2. Use of dimensionality reduction (PCA, SVD, UMAP) to project heterogeneous decision tables into a shared feature space while preserving privacy.

3. An architecture-agnostic training approach where each center independently trains neural network models, preserving local data confidentiality without sharing model parameters.

The rest of the paper is structured as follows: Section 2 reviews related work. Section 3 outlines the proposed framework, including feature transformation, model training and aggregation strategy. Section 4 presents experimental validation using datasets from the UC Irvine (UCI) Machine Learning Repository. Finally, Section 5 concludes with a discussion of applications, limitations and directions for future research.

2. Literature review

Distributed machine learning has advanced rapidly in recent years, with significant focus on handling data heterogeneity, ensuring privacy preservation and improving computational efficiency. Federated learning (FL), established a framework for training global models without centralizing data, thereby preserving user privacy [14]. However, standard FL approaches often assume a shared feature space across clients and struggle with non-independent and identically distributed (non-IID) data [14], [20]. Several works have proposed solutions to address these challenges, including personalized FL [23] and communication-efficient protocols [9].
Privacy-preserving techniques have been further developed to mitigate risks such as attribute inference or model inversion attacks. Differential privacy [1] and secure multi-party computation [21] have been integrated into FL frameworks to strengthen data confidentiality. Nonetheless, these methods generally assume consistent feature sets and require complex cryptographic operations, limiting scalability in heterogeneous environments [4].
Feature extraction and dimensionality reduction techniques play a pivotal role in addressing data heterogeneity. PCA, SVD, and nonlinear methods such as UMAP have been widely employed to reduce feature dimensionality while preserving relevant information [3]. Recent studies have explored distributed implementations of such techniques to enable privacy-preserving feature extraction, such as the work by Fontenla-Romero et al. [7] which uses local SVD computations in decentralized anomaly detection scenarios.
Ensemble learning and voting mechanisms have been extensively used to improve robustness and accuracy in distributed systems. Techniques such as soft voting, bagging, and boosting aggregate predictions from multiple local models to form a consensus decision without requiring parameter sharing [18], [25]. This paradigm not only enhances predictive performance but also contributes to privacy preservation by avoiding the need to exchange raw data or model

parameters [5]. Despite these advances, a critical gap remains in effectively handling fully heterogeneous datasets where neither the object sets nor the attribute spaces overlap across local data centers. Existing FL methods like FedAvg [14] assume partial alignment of data or require costly synchronization that may leak sensitive information [15]. Moreover, most feature extraction approaches presuppose consistent features across clients, limiting their applicability in realistic multi-institutional collaborations where data heterogeneity is intrinsic.
The proposed framework addresses these challenges by unifying structurally incompatible local datasets through dimensionality reduction (PCA, SVD, UMAP) and integrating independent neural network models using soft voting (simple average). While the main objective and contributions are detailed in the introduction, this work uniquely addresses the limitations of existing FL systems by combining transformation-based compatibility with model-level aggregation in a privacy-preserving, structure-independent setting.

3. Methods and models

Let $D_i = (U_i, A_i, d)$, for $i = 1, \ldots, n$, represent a set of heterogeneous decision tables from distributed data centers, where U_i is a set of objects, A_i is a set of attributes, and d is the decision attribute. These tables may differ in both objects and features – a situation common in domains like healthcare, where a patient may visit multiple hospitals with distinct but partially overlapping diagnostic attributes.
Each local dataset is transformed into a uniform feature space using multiple feature extraction techniques. Given a fixed dimensionality k, each transformation (PCA, SVD, UMAP) maps D_i into a set $\mathcal{H}_i^k = \{(U_i, \mathcal{A}_i^k, d)_1, \ldots, (U_i, \mathcal{A}_i^k, d)_T\}$. These are then horizontally concatenated to form a unified table $\mathcal{C}_i^k$, while maintaining local independence and privacy – no raw data or transformation parameters are exchanged between sites.
PCA and SVD [2], [13] provide linear projections to reduce noise and highlight global structure, while UMAP [10] captures nonlinear patterns by preserving local and global topology. Their combined use enhances representational diversity, making the transformed features robust and expressive. Each $\mathcal{C}_i^k$ is used to train an independent neural network. We evaluate five architectures: MLP, RBF, GRU, LSTM, and Simple RNN. MLPs serve as baselines for tabular data [12]; RBFs provide local sensitivity [22]; GRUs and LSTMs capture temporal patterns, with GRUs being more computationally efficient whiles Simple RNNs offer a lightweight benchmark [16]
At inference, a test sample x is transformed using the same feature extractors, yielding $\mathcal{X}^k = [x_{PCA}^k, x_{SVD}^k, x_{UMAP}^k]$ and input into each of the n local models. Their outputs are combined via soft voting (simple average) to produce the final prediction. This two-stage framework – transformation followed by local modeling and decentralized voting ensures privacy, accommodates data heterogeneity, and avoids synchronization overhead.
While transforming local tables into a uniform feature space may incur some degree of information loss, this is mitigated by using a diverse set of transformation maps (PCA, SVD, UMAP). Each method emphasizes different structural aspects of the data (linear variance, latent structure, non-linear manifold). As such, the concatenation strategy recovers complementary information, which increases the likelihood that essential patterns are preserved across maps. Figure 1 illustrates the entire pipeline.

4. Experiments

The experimental assessment of the framework is achieved by conducting tests on three separate and distinct UCI datasets, namely the Landsat Satellite – 36 attributes, 6 decision classes and 6,435 objects (4,435, training, 2,000 test) [19]; Crowdsourced Mapping – 28 attributes, 6 decision classes and 10,844 objects (7,590 training, 3,254 test) [11] and Anuran Calls – comprising 22 attributes, 4 decision classes and 7,195 objects(5,036 training, 2,159 test) [6].

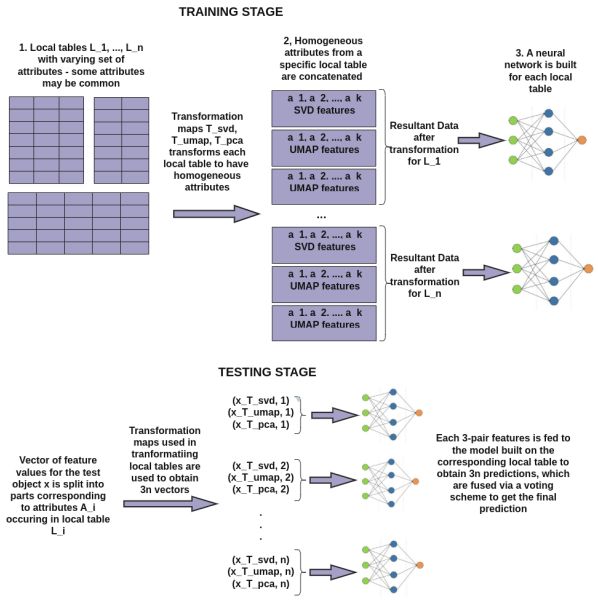

Fig. 1. Model generation and prediction for test object stages.

Each of the datasets is initially available in a non-dispersed form, with all data organized within a single decision table. The training sets are then dispersed, considering varying degrees of dispersion. Each individual data set is transformed into five different dispersed versions, with the dispersions consisting of $3, 5, 7, 9$, and 11 local tables, respectively. During the creation of these local tables, a subset of attributes from the original data set is chosen for each table. The number of attributes in each local table is significantly reduced compared to the original decision table, although certain attributes are repeated across different local tables to ensure that some overlap exists among them. This overlapping is crucial because it allows for the possibility that some local tables may share common attributes. Three transformation maps (PCA, SVD, UMAP) and five neural network models (GRU, LSTM, MLP, RBF and SIMPLE) were considered in the experiments.

Experiments varied model architectures and hyperparameters, including hidden layer depths (1-3) and the number of principal components k (ranging from 1 to the feature count of the smallest local table). Hidden layer sizes were scaled relative to the input dimension (I) – for 1 layer – $\{4, 6, 9, 12, 20\} \times I$; 2 layers – $\{(4,2), (5,3), (6,3), (8,4), (10,5), (12,6), (15,9), (20,10)\} \times I$; 3 layers – $\{(4,3,2), (6,5,4), (7,5,3), (8,4,2), (10,7,3), (12,6,3), (15,9,4), (20,10,5)\} \times I$.

Tested models included RBF (1 layer), GRU, LSTM, SIMPLE (1-2 layers), and MLP (1-3 layers). Architectural choices reflect each model's structural limitations and capabilities: recurrent models were restricted to shallow configurations to mitigate overfitting and gradient instability, while MLPs were explored more extensively due to their capacity for deep feature learning. These settings were selected to balance expressiveness and stability across architectures.

Model performance is evaluated using standard metrics: accuracy, precision, recall, and F1-score, with the latter providing a balanced measure under class imbalance. Each experiment was repeated three times, and average results are reported. Full results for MLP are provided in Table 1, while Tables 2–5 present selected outcomes for other architectures. Reported values correspond to optimal network configurations for each k, with best accuracy results highlighted in bold.

As can be seen, MLP and SIMPLE networks consistently achieved the highest accuracy and F-measure scores, particularly on the Satellite and Anuran datasets. For instance, MLP reached 0.855 accuracy on Satellite and 0.914 on Anuran, while SIMPLE achieved similar or better

Table 1. Results of prec., rec., F-m, acc, for MLP; config denotes neurons in the hidden layers.

Data set	No. tables	k	Performance metrics				config	k	Performance metrics				config
			prec.	*rec.*	F-m	*acc.*			*prec.*	*rec.*	F-m	*acc.*	
Satellite	3	1	0.515	0.51	0.477	0.51	(15;9)×I	6	0.839	0.832	0.825	0.832	(20;10;5)×I
		2	0.709	0.734	0.694	0.734	(10;7;3)×I	7	0.851	0.843	0.836	0.843	(15;9;4)×I
		3	0.822	0.825	0.811	0.825	(2;10;5)×I	8	0.857	0.853	0.847	0.853	(20;10;5)×I
		4	0.837	0.832	0.821	0.832	(20;10;5)×I	9	0.857	0.855	0.847	**0.855**	(20;10;5)×I
		5	0.833	0.828	0.817	0.828	(20;10;5)×I	10	0.856	0.855	0.847	**0.855**	(15;9;4)×I
	5	1	0.546	0.539	0.489	0.539	(20;10)×I	5	0.853	0.852	0.843	**0.852**	(15;9;4)×I
		2	0.677	0.708	0.655	0.708	(15;9;4)×I	6	0.835	0.835	0.825	0.835	(15;9;4)×I
		3	0.826	0.818	0.801	0.818	(15;9)×I	7	0.849	0.852	0.843	**0.852**	(15;9;4)×I
		4	0.84	0.836	0.821	0.836	(20;10)×I	8	0.839	0.836	0.817	0.836	20×I
	7	1	0.53	0.517	0.442	0.517	(15;9)×I	4	0.82	0.818	0.788	0.818	(15;9;4)×I
		2	0.711	0.77	0.719	0.77	(20;10;5)×I	5	0.841	0.836	0.816	**0.836**	(20;10;5)×I
		3	0.825	0.808	0.768	0.808	(20;10;5)×I						
	9	1	0.54	0.542	0.476	0.542	(20;10)×I	3	0.803	0.798	0.756	**0.798**	(20;10;5)×I
		2	0.71	0.768	0.717	0.768	(20;10;5)×I						
	11	1	0.411	0.474	0.404	0.474	(15;9)×I	3	0.821	0.813	0.773	**0.813**	(20;10;5)×I
		2	0.71	0.767	0.715	0.767	(20;10;5)×I						
Crowd sourced	3	1	0.479	0.692	0.567	0.692	4×I	6	0.782	0.778	0.711	0.778	(20;10;5)×I
		2	0.479	0.692	0.567	0.692	4×I	7	0.793	0.79	0.736	0.79	(20;10;5)×I
		3	0.588	0.732	0.635	0.732	(20;10;5)×I	8	0.756	0.784	0.725	0.784	(15;9;4)×I
		4	0.734	0.761	0.675	0.761	20×I	9	0.813	0.8	0.753	0.8	(15;9;4)×I
		5	0.712	0.768	0.692	0.768	(15;9;4)×I	10	0.8	0.802	0.757	**0.802**	(20;10;5)×I
	5	1	0.479	0.692	0.567	0.692	4×I	4	0.686	0.712	0.605	0.712	(20;10;5)×I
		2	0.479	0.692	0.567	0.692	4×I	5	0.699	0.707	0.6	0.707	(15;9;4)×I
		3	0.566	0.7	0.584	0.7	(15;9;4)×I	6	0.73	0.736	0.656	**0.736**	(20;10;5)×I
	7	1	0.479	0.692	0.567	0.692	4×I	4	0.572	0.704	0.591	0.704	(20;10;5)×I
		2	0.479	0.692	0.567	0.692	4×I	5	0.739	0.757	0.677	**0.757**	(20;10;5)×I
		3	0.567	0.699	0.582	0.699	(15;9)×I						
	9	1	0.479	0.692	0.567	**0.692**	4×I	3	0.479	0.692	0.567	**0.692**	4×I
		2	0.479	0.692	0.567	**0.692**	4×I						
	11	1	0.479	0.692	0.567	0.692	4×I	3	0.573	0.693	0.568	**0.693**	(20;10;5)×I
		2	0.479	0.692	0.567	0.692	4×I						
Anuran	3	1	0.684	0.65	0.542	0.65	(15;9)×I	5	0.896	0.904	0.898	0.904	20×I
		2	0.758	0.817	0.777	0.817	(6;3)×I	6	0.893	0.895	0.883	0.895	(15;9;4)×I
		3	0.828	0.842	0.814	0.842	(20;10)×I	7	0.906	0.914	0.908	**0.914**	12×I
		4	0.867	0.871	0.855	0.871	(15;9;4)×I	8	0.897	0.902	0.891	0.902	(12;6)×I
	5	1	0.677	0.687	0.614	0.687	(20;10)×I	4	0.847	0.861	0.847	0.861	(20;10;5)×I
		2	0.703	0.752	0.704	0.752	(20;10;5)×I	5	0.868	0.875	0.864	**0.875**	(20;10;5)×I
		3	0.798	0.823	0.787	0.823	(12;6)×I						
	7	1	0.68	0.619	0.479	0.619	(20;10;5)×I	3	0.755	0.79	0.747	0.79	20×I
		2	0.755	0.792	0.749	**0.792**	(20;10;5)×I						
	9	1	0.377	0.614	0.467	0.614	4×I	3	0.725	0.742	0.687	0.742	(20;10)×I
		2	0.742	0.771	0.725	**0.771**	(20;10;5)×I						
	11	1	0.377	0.614	0.467	0.614	4×I	2	0.742	0.764	0.715	**0.764**	(8;4;2)×I

Table 2. Results of prec., rec., F-m, acc, for LSTM; config denotes neurons in the hidden layers.

Data set	No. tables	k	Performance metrics				config	k	Performance metrics				config
			prec.	*rec.*	F-m	*acc.*			*prec.*	*rec.*	F-m	*acc.*	
Satellite	3	4	0.813	0.811	0.794	**0.811**	(20;10)×I	9	0.807	0.798	0.773	0.798	20×I
	5	1	0.545	0.541	0.467	0.541	20×I	5	0.818	0.82	0.799	**0.82**	20×I
	7	1	0.548	0.55	0.48	0.55	(20;10)×I	4	0.721	0.796	0.75	**0.796**	(20;10)×I
	9	1	0.434	0.546	0.47	0.546	20×I	3	0.713	0.762	0.718	**0.762**	20×I
	11	1	0.426	0.49	0.416	0.49	9×I	3	0.729	0.784	0.737	**0.784**	20×I
Crowd sourced	3	2	0.656	0.72	0.624	0.72	(10;5)×I	7	0.712	0.775	0.715	0.775	4×I
		5	0.669	0.768	0.684	0.768	4×I	10	0.836	0.831	0.795	**0.831**	20×I
	5	3	0.595	0.705	0.593	0.705	(10;5)×I	6	0.632	0.748	0.659	**0.748**	4×I
	7	2	0.479	0.692	0.567	0.692	4×I	5	0.634	0.736	0.652	**0.736**	(8;4)×I
	9	1	0.479	0.692	0.567	0.692	4×I	3	0.519	0.695	0.573	**0.695**	(4;2)×I
	11	1	0.479	0.692	0.567	0.692	4×I	3	0.515	0.693	0.568	**0.693**	(5;3)×I
Anuran	3	1	0.708	0.696	0.619	0.696	(18;4)×I	5	0.833	0.839	0.815	**0.839**	9×I
	7	1	0.511	0.659	0.557	0.659	(12;6)×I	3	0.757	0.813	0.776	**0.813**	(12;6)×I
	9	1	0.525	0.636	0.513	0.636	(15;9)×I	3	0.744	0.807	0.767	**0.807**	(4;2)×I
	11	1	0.479	0.692	0.567	0.692	(10;5)×I	2	0.515	0.693	0.568	**0.693**	(12;6)×I

Table 3. Results of prec., rec., F-m, acc, for GRU; config denotes neurons in the hidden layers.

Data set	No. tables	k	Performance metrics				config	k	Performance metrics				config
			prec.	*rec.*	F-m	*acc.*			*prec.*	*rec.*	F-m	*acc.*	
Satellite	3	5	0.757	0.787	0.757	0.787	20×I	10	0.86	0.855	0.843	**0.855**	20×I
	5	3	0.806	0.803	0.785	0.803	20×I	7	0.842	0.837	0.82	**0.837**	20×I
	7	2	0.683	0.723	0.676	0.723	(20;10)×I	5	0.82	0.806	0.778	**0.806**	20×I
	9	1	0.572	0.559	0.486	0.559	(15;9)×I	3	0.754	0.776	0.731	**0.776**	20×I
	11	1	0.427	0.486	0.413	0.486	20×I	3	0.731	0.79	0.741	**0.79**	(20;10)×I
Crowd sourced	3	3	0.604	0.741	0.645	0.741	(10;5)×I	8	0.819	0.817	0.781	**0.817**	20×I
		4	0.732	0.783	0.709	0.783	6×I	9	0.817	0.817	0.777	**0.817**	20×I
	5	2	0.511	0.702	0.589	0.702	(8;4)×I	5	0.727	0.774	0.703	**0.774**	4×I
	7	2	0.479	0.692	0.567	0.692	4×I	5	0.741	0.766	0.685	**0.766**	(10;5)×I
	9	1	0.479	0.692	0.567	0.692	4×I	3	0.525	0.706	0.593	**0.706**	(8;4)×I
	11	1	0.479	0.692	0.567	0.692	4×I	3	0.523	0.694	0.57	**0.694**	(10;5)×I
Anuran	3	3	0.853	0.857	0.841	0.857	9×I	7	0.884	0.884	0.876	**0.884**	12×I
	5	2	0.685	0.745	0.69	0.745	(5;3)×I	5	0.855	0.846	0.838	**0.846**	(15;9)×I
	7	2	0.764	0.83	0.791	**0.83**	(10;5)×I						
	9	1	0.666	0.695	0.633	0.695	(15;9)×I	3	0.762	0.825	0.786	**0.825**	(12;6) ×I
	11	1	0.677	0.692	0.599	0.692	(10;10)×I	3	0.771	0.826	0.794	**0.826**	(20;10)×I

performance in some configurations. GRU also performed strongly, whereas LSTM generally trailed slightly in both accuracy and computational efficiency. RBF networks showed solid results, especially at higher dimensionalities. Statistical analysis were conducted using the F-measure to compare five neural network types across 77 conditions (dataset, dispersion version, and k values). Due to the non-normal distribution of ratio-scaled data, the Friedman test was applied, revealing a significant difference among the models ($\chi^2(4, 76) = 42.70, p = 0.00001$) with MLP, SIMPLE, and GRU ranking highest as shown in Figure 2 The dimensionality pa-

Table 4. Results of prec., rec., F-m, acc, for SIMPLE; config denotes neurons in the hidden layers.

Data set	No. tables	k	Performance metrics				config	k	Performance metrics				config
			prec.	*rec.*	F-m	*acc.*			*prec.*	*rec.*	F-m	*acc.*	
Satellite	3	3	0.836	0.829	0.812	0.829	(20;10)×I	8	0.868	0.866	0.857	**0.866**	(20;10)×I
	5	1	0.511	0.542	0.467	0.542	20×I	5	0.861	0.861	0.855	**0.861**	20×I
	7	2	0.71	0.768	0.719	0.768	20×I	5	0.86	0.851	0.837	**0.851**	20×I
	9	1	0.428	0.523	0.457	0.523	(12;6)×I	3	0.806	0.797	0.757	**0.797**	(20;10)×I
	11	1	0.329	0.498	0.395	0.498	(10;5)×I	3	0.833	0.824	0.789	**0.824**	(20;10)×I
Crowd sourced	3	2	0.479	0.692	0.567	0.692	4×I	7	0.816	0.806	0.76	0.806	12×I
		3	0.712	0.743	0.652	0.743	12×I	8	0.826	0.836	0.797	**0.836**	(6;3)×I
	5	3	0.519	0.693	0.568	0.693	12×I	6	0.765	0.778	0.724	**0.778**	(20;10)×I
	7	2	0.479	0.692	0.567	0.692	4×I	5	0.761	0.771	0.709	**0.771**	20×I
	9	1	0.479	0.692	0.567	**0.692**	4×I	3	0.479	0.692	0.567	**0.692**	4×I
	11	1	0.479	0.692	0.567	0.692	4×I	3	0.565	0.697	0.577	**0.697**	(15;9)×I
Anuran	3	3	0.828	0.833	0.823	0.833	12×I	7	0.905	0.911	0.906	**0.911**	4×I
	5	2	0.699	0.734	0.68	0.734	(12;6)×I	5	0.881	0.884	0.879	**0.884**	(8;4)×I
	7	1	0.377	0.614	0.467	0.614	4×I	3	0.764	0.816	0.776	**0.816**	12×I
	9	1	0.377	0.614	0.467	0.614	4×I	3	0.84	0.814	0.776	**0.814**	20 ×I
	11	1	0.679	0.617	0.474	0.617	(15;9)×I	3	0.768	0.817	0.778	**0.817**	12×I

Table 5. Results of prec., rec., F-m, acc, for RBF; config denotes neurons in the hidden layers.

Data set	No. tables	k	Performance metrics				config	k	Performance metrics				config
			prec.	*rec.*	F-m	*acc.*			*prec.*	*rec.*	F-m	*acc.*	
Satellite	3	5	0.858	0.853	0.844	**0.853**	20×I	10	0.853	0.84	0.828	0.84	6×I
	5	2	0.701	0.731	0.681	0.731	20×I	6	0.871	0.864	0.853	**0.864**	20×I
	7	1	0.529	0.503	0.425	0.503	20×I	4	0.856	0.838	0.812	**0.838**	20×I
	9	1	0.564	0.562	0.499	0.562	20×I	3	0.837	0.817	0.779	**0.817**	20×I
	11	1	0.315	0.473	0.375	0.473	20×I	3	0.81	0.815	0.776	**0.815**	20×I
Crowd sourced	3	4	0.805	0.786	0.728	0.786	20×I	9	0.778	0.778	0.716	0.778	6×I
		5	0.772	0.783	0.724	0.783	20×I	10	0.785	0.793	0.74	**0.793**	6×I
	5	3	0.642	0.693	0.568	0.693	12×I	6	0.75	0.758	0.69	**0.758**	20×I
	7	2	0.479	0.692	0.567	0.692	4×I	5	0.782	0.761	0.689	**0.761**	20×I
	9	1	0.479	0.692	0.567	**0.692**	4×I	3	0.479	0.692	0.567	**0.692**	4×I
	11	1	0.479	0.692	0.567	0.692	4×I	3	0.711	0.695	0.571	**0.695**	20×I
Anuran	3	1	0.377	0.614	0.467	0.614	4×I	5	0.849	0.85	0.824	**0.85**	9×I
	5	2	0.644	0.696	0.642	0.696	20×I	5	0.781	0.787	0.749	**0.787**	12×I
	7	2	0.747	0.8	0.758	**0.8**	20×I						
	9	2	0.756	0.802	0.76	**0.802**	20×I						
	11	1	0.661	0.621	0.483	0.621	20×I	3	0.764	0.809	0.769	**0.809**	20×I

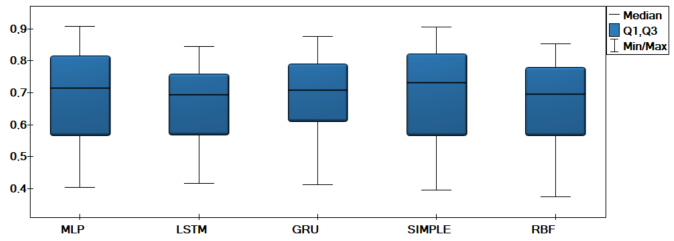

Fig. 2. Comparison of F-measure obtained for all analyzed types of neural network.

rameter k significantly affects model performance. Lower values (e.g., $k = 1, 2$) yield poor results, while performance improves consistently with higher k. This confirms that a higher-dimensional representation captures more discriminative features. To validate this, we applied the Kruskal-Wallis test to F-measure results grouped by k. Group sizes varied from 70 observations for $k \in \{1, 2, 3\}$ to 10 for $k = 9$ and 10. The test confirmed a significant effect of k on performance ($H(5) = 45.33$, $p < 0.00001$). As shown in Figure 3, higher k values correspond to increased median F-measure, indicating that richer, higher-dimensional representations enable better model performance.

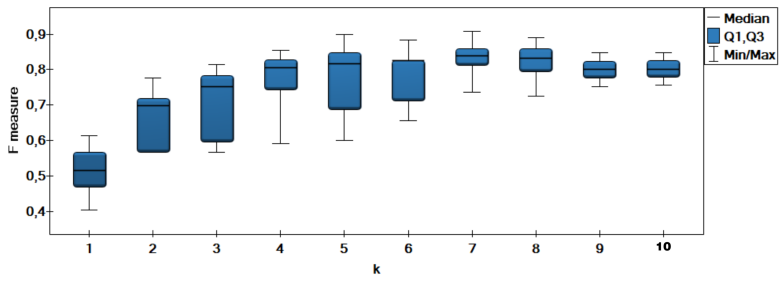

Fig. 3. Comparison of F-measure obtained for all different k values and MLP network.

Data dispersion – the number of local tables was also examined, with results showing the framework's robustness to varying levels of information separation. Performance remained stable across low and high dispersion settings, underscoring its suitability for distributed environments. Comparative analysis with recent methods [17], [24] showed comparable performance. Unlike approaches such as FedFA, which require alignment through shared features or supervised anchors [24], or federated PCA/SVD methods that transmit transformation subspaces [8], our framework avoids global parameter or feature sharing, enhancing privacy and supporting full heterogeneity.

5. Conclusion

This paper introduced a novel framework for decentralized neural network modeling across heterogeneous data sources, addressing disparities in object sets and feature spaces without sharing raw data. By mapping local datasets into equal-dimensional spaces via PCA, SVD, and UMAP, and aggregating independently trained models through soft voting (simple average), the approach maintains both structural and privacy constraints. Experiments on three UCI datasets validate the framework's effectiveness, with strong classification results from MLP, SIMPLE, and GRU models at higher dimensionalities.
However, the results are influenced by specific dataset characteristics, model architectures, and transformation choices, which may limit generalizability. Potential limitations include information loss from dimensionality reduction and computational overhead from parallel training. Future work will explore alternative transformation methods, unified table representations, and stacking-based global model integration.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. p. 308–318. CCS '16, New York, NY, USA (2016)

[2] Baker, K.: Singular value decomposition tutorial. The Ohio State University 24, pp. 22 (2005)

[3] de-la Bandera, I., Palacios, D., Mendoza, J., Barco, R.: Feature extraction for dimensionality reduction in cellular networks performance analysis. Sensors 20(23) (2020)

[4] Cao, F., Liu, B., He, J., Xu, J., Xiao, Y.: Privacy preservation-based federated learning with uncertain data. Information Sciences 678, pp. 121024 (2024)

[5] Chaudhary, N., Gupta, V., Sandhir, K., Gupta, R., Chhabra, S., Singh, A.: Privacy preserving ensemble learning classification model for mental healthcare. pp. 513–518 (11 2022)

[6] Colonna, J., Nakamura, E., Cristo, M., Gordo, M.: Anuran Calls (MFCCs). UCI Machine Learning Repository (2015), DOI: https://doi.org/10.24432/C5CC9H

[7] Fontenla-Romero, O., Pérez-Sánchez, B., Guijarro-Berdiñas, B.: Dsvd-autoencoder: a scalable distributed privacy-preserving method for one-class classification. International Journal of Intelligent Systems 36(1), pp. 177–199 (2021)

[8] Hartebrodt, A., Röttger, R., Blumenthal, D.B.: Federated singular value decomposition for high-dimensional data. Data Mining and Knowledge Discovery 38(3), pp. 938–975 (2024)

[9] He, S., Zheng, J., Feng, M., Chen, Y.: Communication-efficient federated learning with adaptive consensus admm. Applied Sciences 13(9) (2023)

[10] Healy, J., McInnes, L.: Uniform manifold approximation and projection. Nature Reviews Methods Primers 4(1), pp. 82 (2024)

[11] Johnson, B.: Crowdsourced Mapping. UCI Machine Learning Repository (2016), DOI: doi.org/ 10.24432/C56315

[12] Kadra, A., Lindauer, M., Hutter, F., Grabocka, J.: Well-tuned simple nets excel on tabular datasets. arXiv preprint arXiv:2106.11189 (2021)

[13] Kurita, T.: Principal component analysis (pca). In: Computer vision: a reference guide, pp. 1013–1016. Springer (2021)

[14] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)

[15] Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 691–706 (2019)

[16] Mienye, I.D., Swart, T.G., Obaido, G.: Recurrent neural networks: A comprehensive review of architectures, variants, and applications. Information 15(9) (2024)

[17] Przybyła-Kasperek, M., Marfo, K.F.: A multi-layer perceptron neural network for varied conditional attributes in tabular dispersed data. PloS one 19(12), pp. e0311041 (2024)

[18] Rokach, L.: Ensemble-based classifiers. Artificial Intelligence Review 33(1), pp. 1–39 (Feb 2010)

[19] Srinivasan, A.: Statlog (Landsat Satellite). UCI Machine Learning Repository (1993), DOI: doi. org/10.24432/C55887

[20] Tan, Q., Wu, S., Tao, Y.: Privacy-enhanced federated learning for non-iid data. Mathematics 11(19) (2023)

[21] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., Zhou, Y.: A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. p. 1–11. AISec'19, New York, NY, USA (2019)

[22] Yang, Y., Wang, P., Gao, X.: A novel radial basis function neural network with high generalization performance for nonlinear process modelling. Processes 10(1), pp. 140 (2022)

[23] Zhang, G., Liu, B., Zhu, T., Ding, M., Zhou, W.: Ppfed: A privacy-preserving and personalized federated learning framework. IEEE Internet of Things Journal 11(11), pp. 19380–19393 (2024)

[24] Zhou, T., Zhang, J., Tsang, D.: Fedfa: Federated learning with feature anchors to align features and classifiers for heterogeneous data. IEEE Transactions on Mobile Computing PP, pp. 1–12 (01 2023)

[25] Zhou, Z.H.: Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, 1st edn. (2012)