Ensembling Neural Networks for Improved Prediction and Privacy in Early Diagnosis of Sepsis

Shigehiko Schamoni SCHAMONI@CL.UNI-HEIDELBERG.DE
Michael Hagmann HAGMANN@CL.UNI-HEIDELBERG.DE
Stefan Riezler RIEZLER@CL.UNI-HEIDELBERG.DE
Department of Computational Linguistics &
Interdisciplinary Center for Scientific Computing (IWR)
Heidelberg University, Germany

Abstract

Ensembling neural networks is a long-standing technique for improving the generalization error of neural networks by combining networks with orthogonal properties via a committee decision. We show that this technique is an ideal fit for machine learning on medical data: First, ensembles are amenable to parallel and asynchronous learning, thus enabling efficient training of patient-specific component neural networks. Second, building on the idea of minimizing generalization error by selecting uncorrelated patient-specific networks, we show that one can build an ensemble of a few selected patient-specific models that outperforms a single model trained on much larger pooled datasets. Third, the non-iterative ensemble combination step is an optimal low-dimensional entry point to apply output perturbation to guarantee the privacy of the patient-specific networks. We exemplify our framework of differentially private ensembles on the task of early prediction of sepsis, using real-life intensive care unit data labeled by clinical experts.

1. Introduction

Ensembling describes a family of algorithms that train multiple learners to solve the same problem, and exploit their heterogeneous properties to perform a committee-based prediction that achieves higher accuracy than any single component learner. These techniques are well-tried in machine learning practice and have led to theoretically well-founded algorithms such as stacking (Wolpert, 1992), boosting (Freund and Schapire, 1995), or bagging (Breiman, 1996). Research on ensembling has very early tackled the problem of reducing variance of neural networks while keeping bias low at the same time. In the wide spectrum of approaches, ranging from sophisticated techniques to jointly train component networks (Liu and Yao, 1999; Buschjäger et al., 2020) to building ensembles from model parameters of a single training trajectory (Huang et al., 2017; Izmailov et al., 2018), we are specifically interested in approaches where component models are trained independently and then smartly combined.

A key insight in this area, first formulated in Perrone and Cooper (1992), is that the generalization error of the weighted average of predictions of individual component networks can be formalized as the weighted correlation between the component neural networks participating in the ensemble. This formulation opens several possibilities for efficient and effective machine learning: First, the bulk of the machine learning cost, namely the cost of training individual component networks, can be trivially parallelized or even be done asynchronously, thus providing an efficient way of enhancing the representational power of the ensemble by training multiple classifiers at once. Second, optimizing combination weights to minimize the weighted correlation between component networks provides a direct avenue to minimize the generalization error of the ensemble, or to build a sparse ensemble from the optimal subset of component networks with small error and small correlation with other component networks.

A further advantage of weighted-averaging ensembles that has been investigated much less than their generalization performance is the possibility to seamlessly integrate privacy protection into machine learning. In the case of machine learning models trained on medical data, the privacy to be protected might concern the membership of patient-specific data in the training data for a particular disease. As argued by Dinur and Nissim (2003), removal of "identifying" attributes such as patients' names is not enough, but instead random perturbations have to be applied to the outputs in order to protect privacy even in the simplest case of "statistical" queries such as averages over databases. The framework of differential privacy (Dwork and Roth, 2014) allows giving strong guarantees on the information derivable from private training data when querying a machine learning algorithm. We show that weighted-averaging ensembles do possess small sensitivity by tightly bounded output ranges and do not accumulate privacy budget via iterative training, thus they are ideally suited for privacy protection at small noise scales. Furthermore, we prove that uniform weights are optimal to protect privacy in a weighted-averaging ensemble.

Specifying guarantees on privacy protection is of increasing importance for medical research. National laws and regulations such as the US HIPAA Privacy rule[1] require measures to protect the privacy of health information. On the hospital level, protecting a patient's privacy is crucial especially when information is shared across institutions. Our method demonstrates the benefit of *output sharing* where hospitals keep their in-house model in a secured area and only share the output with other institutions, thus avoiding the challenges and difficulties of *model sharing* techniques such as federated learning (Rieke et al., 2020). On the patient level, a recent survey has shown that more than 30% of the participants are comfortable with sharing their electronic health data for personalized healthcare, while less than 5% are very uncomfortable with sharing (Garett and Young, 2022). This means more than 60% do not have a strong opinion on this topic, thus we hope that an increasing number of people will share their data if stronger privacy guarantees can be given.

Generalizable Insights about Machine Learning in the Context of Healthcare

Expert labels and neural networks are a powerful combination for early sepsis prediction. However, patient data for this task is scarce as expert labels are difficult to obtain, while the protection of privacy is crucial to encourage patients to contribute with their personal private data. We show how to train individual personalized models and how to combine a small number of patient models in an ensemble that has more desirable properties in the field of medical data analysis than a standard full model, i.e., a single model that is trained on all available patient data.

[1] www.hhs.gov/hipaa/for-professionals/privacy/ (accessed 07/06/2022)

- We present theoretical results that an ensemble of models which was trained on a fraction of the available data can be better than a full model, and we verify this empirically.
- Our training method not only exposes fewer patients in the predictor than a full model, but also protects the privacy better: we apply a strong membership attack and show that the ensemble successfully prevents privacy leakage.
- We show that an ensemble of several models is favorable to a single model due to its reduced sensitivity in theory, and we experimentally verify that our ensemble maintains its accuracy at privacy budgets almost two orders of magnitude smaller than a full model.

Furthermore, our ensemble can be easily updated by model-growing without the need of retraining the whole system when new patient's data becomes available.

2. Related Work

Ensembles of neural networks have been researched at least since Hansen and Salamon (1990), and are now a standard tool of deep learning. The spectrum of approaches ranges from joint training of component models under ensemble objectives such as negative correlation learning (Liu and Yao, 1999; Buschjäger et al., 2020) to approaches to efficiently build ensembles by combining snapshots of model parameters along the training trajectory of a single network by averaging in model space (Huang et al., 2017), or weight space (Izmailov et al., 2018). Even well-known staples such as dropout can be seen as ensembles of subnetworks (Srivastava et al., 2014). For a recent overview over ensemble deep learning, see Ganaie et al. (2021).

Traditionally, ensemble methods are often used in medical data science. Recent examples can be found in the area of early prediction of sepsis: Goh et al. (2021) use a voting ensemble of a logistic regression model and a random forest trained on same data; Moor et al. (2021) use a max-score ensemble of four machine learning models trained on four different datasets. The privacy preserving aspects of ensemble methods in health care, however, have only been investigated recently. Fritchman et al. (2018) describe a framework for privacy preserving inference using cryptographic protocols. Adams et al. (2022) demonstrate secure training of ensembles in a multi-party computation scenario. Both works are based on decision tree ensembles, while our ensemble strategy can be combined with any machine learning model.

Differential privacy has become a de-facto standard for privacy protection at least since Dwork (2006). This framework has been applied to the case of privacy protection in machine learning where the goal is to protect the privacy of training data when publicly releasing a machine learning model. To our knowledge, despite their natural fit to protect privacy in the combination of patient-specific models, differentially private ensembles have not yet been widely used in medical data science. Instead, the paradigm of cryptography still seems to be going strong in the area of collaborative learning on health data (Gong et al., 2015).

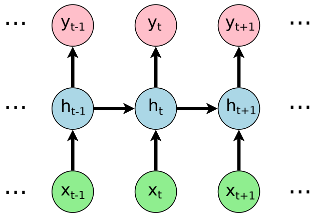


Figure 1: Schematic view on the RNN architecture. For each input $\mathbf{x}$, the RNN is in a hidden state $\mathbf{h}$ that is also conditioned on the previous hidden state and generates an output y.

3. Methods

In this section, we first describe the basic machine learning model that is used throughout our experiments. We then explain how we combine multiple models in an ensemble that improves prediction accuracy, and we also show how ensembles can enhance differential privacy. Based on the theoretical results, we finally design an algorithm for greedy ensemble growing.

3.1. Recurrent Neural Networks over Time Series

Our basic machine learning models are recurrent neural network (RNN) architectures that predict a severity score y_t for each time step t (see Figure 1). We use discretized 30 minute steps as input for our models and predict a sepsis-related score. The predicted score at each time step is an expert label ranging from 0 to 4. The motivation for using an RNN-based system is the intuition that a recurrent network is able to model a dynamic system over time (Durstewitz et al., 2021) (i.e., the development of the patient) with its feedback loop connections. We implement a special form of RNNs, namely LSTMs (Hochreiter and Schmidhuber, 1997), due to their ability to model both short and long term dependencies in time series. However, our ensemble growing strategy can easily be adapted to other sequence-to-sequence models such as Transformers (Vaswani et al., 2017), or even to time-agnostic models such as feed-forward nets or decision trees. Details on the architecture and meta-parameters of our model are given in Appendix A.

We train two different types of models with a regression objective. First, a *full model* trained on all available data, and secondly, an *ensemble* of individual models that are each trained on the data of a single patient. The training data for the full model consist of 638 or 637 patient's timelines depending on the data fold (see Section 4.1). It is trained for a maximum of 200 epochs with an early stopping criterion on the validation set. We originally optimized the general model architecture on this model type and use the same architecture throughout our experiments.

3.2. Weighted Averaging Ensemble

In our setup, we are combining models that were trained on individual patients, which makes each model an "expert" (Jacobs et al., 1991) of sepsis prediction for a specific patient. We then combine selected prototypical patient models by weighting their predictions, motivated by the insight that a physician uses past experience to make future decisions. Such a weighted-average ensemble has very useful properties. Specifically, the ensemble's mean squared error (MSE) can be decomposed as a linear combination of the MSE and the covariances of the errors of the individual components models. Furthermore, a closed-form solution for optimal weights is completely determined by the variance-covariance matrix of the model's errors on a held-out set. Similar results have been shown by Perrone and Cooper (1992) for the case of cross-validation, a setup that is usually applied to prevent overfitting of hyperparameters. We use cross-validation in our experiments particularly due to the fact that our medical data is small and thus our learned models easily overfit and hence the test-set performance shows high variance. Zhou et al. (2002) also exploit this decomposition of the MSE to improve generalization when selecting models and determining weights by using a genetic algorithm as a heuristic.

Definition 1 *Let* $f, \hat{f}_i : \mathbb{R}^m \to \mathbb{R}, i = 1, \ldots, N$*. Further let* $\mathfrak{F} = \{\hat{f}_i(x) := \widehat{\mathbb{E}[y|x]}_i\}$ *be a set of regression estimates of a regression function* $f(x) := \mathbb{E}[y|x]$ *and* $w_i \in \mathbb{R}$ *subject to* $\sum_{i=1}^N w_i = 1$*. Then*

$$\hat{f}_{em}(x) := \sum_{i=1}^{N} w_i \hat{f}_i(x)$$

is called the weighted-average ensemble estimator*.*

When comparing two models' predictions to decide which is best, it is useful to define a measure of the length of the misfit vector $\mathbf{m}$. Each element in this vector is given by the error of the model's prediction f_i for input x, i.e., the misfit $m_i(x) := f_i(x) - f(x)$ of function $f_i(x)$ with respect to the true value $f(x)$. A norm defined on the vector data space then returns the length of the misfit vector, which is in its simplest form the inner product $M = \mathbf{m}^\intercal \mathbf{m}$ or the ℓ_2-norm of $\mathbf{m}$. Conveniently, the length of the misfit vector is equal to the squared error of f_i in this case. A comparison of the ℓ_2-norm of the misfits of two models is thus a comparison of the squared error, and the resulted ranking of models is equivalent to a ranking w.r.t. their MSE.

To measure how different two models are, it is again useful to compare two models' predictions $f_i(x)$ and $f_j(x)$ by looking at the misfits of both functions and calculate their covariance $\sigma_{ij}^2 = \frac{1}{N}\mathbf{m}_i^\intercal \mathbf{m}_j$. The covariance of functions that generate more similar predictions is above 0, of functions that generate more opposite predictions below 0, and of functions that generate orthogonal predictions it is 0. We are specifically interested in the last case, as a system of orthogonal functions improves generalization if combined with a suitable ensemble growing strategy (Perrone and Cooper, 1992; Zhou et al., 2002).

Summarizing, the covariance of the errors of a set of models can be represented by a quadratic matrix $\mathbf{C}$ where each element is given by the covariance of the prediction errors of two individual models or by the mean squared error of a single model on the diagonal:

$$\mathrm{C}_{ij} := \mathbb{E}_X[(\hat{f}_i(X) - f(X))(\hat{f}_j(X) - f(X))] = \int (\hat{f}_i(X) - f(X))(\hat{f}_j(X) - f(X)) d\mathrm{P}_X$$

is symmetric, $C_{ij} = C_{ji}$, and

$$\mathrm{MSE}(\hat{f}_i) := \mathbb{E}_X[(\hat{f}_i(X) - f(X))^2] = \mathrm{C}_{ii}$$

The following theorem shows that the MSE of the ensemble estimator can be expressed in terms of the C_{ij}.

Lemma 2 *Let $\hat{f}_{em}(x)$ be the ensemble estimator constructed from $\mathfrak{F} = \{\hat{f}_i : i = 1, \ldots, N\}$ and $\mathbf{C} = [C_{ij}]_{i,j=1,\ldots,N}$ be the covariance matrix of $\mathfrak{F}$. Then*

$$\begin{aligned} MSE(\hat{f}_{em}(x)) &= \mathbf{w}^\intercal \mathbf{C} \mathbf{w} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j C_{ij} \\ &= \sum_{i=1}^{N} w_i^2 C_{ii} + 2 \sum_{i=2}^{N} \sum_{j=1}^{i-1} w_i w_j C_{ij} \\ &= \sum_{i=1}^{N} w_i^2 MSE(\hat{f}_i) + 2 \sum_{i=2}^{N} \sum_{j=1}^{i-1} w_i w_j C_{ij} \end{aligned}$$

for all $\mathbf{w} = [w_1, \ldots, w_N]^\intercal \in \mathbb{R}^N$.

This fact also facilitates an optimality result with respect to the weights that is given in Appendix B. An important consequence of this result is the fact that if the weights are chosen in an optimal way, then $\mathrm{MSE}(\hat{f}_{em}) \leq \min_{j=1,\ldots,N} \mathrm{MSE}(\hat{f}_j)$.

3.3. Differentially Private Ensembling

Differential privacy for machine learning is commonly conceptualized as protecting the privacy of training data by randomized optimization algorithms that output learned weights. For example, during stochastic gradient descent (SGD) training of machine learning models, privacy-preserving noise can be added to the trained weights, to the learning objective, or the gradient-based weight updates (Jayaraman and Evans, 2019). While convex optimization algorithms like logistic regression allow for output perturbation or objective perturbation (Chaudhuri et al., 2011), non-convex optimization for deep neural networks requires iterative gradient perturbation in addition to gradient clipping (Shokri and Shmatikov, 2015; Abadi et al., 2016; McMahan et al., 2016). Although these strategies provide privacy guarantees for model sharing in theory, there exist several disadvantages in practice:

1. The required privacy budget ϵ for deep learning can be proportional to the size of the target deep learning model, leading to ϵ values in the order of hundreds of thousands or more. Jayaraman and Evans (2019) show that the large privacy budgets that are required in practice in non-convex optimization severely undermine the value of privacy guarantees provided by differential privacy.

2. Bounding the gradient norm in order to restrict the sensitivity of SGD training for deep neural networks is limited by a tradeoff between privacy protection and network performance, where training speed and prediction accuracy are lowered if the bound on the gradient norm is too tight.

3. Iterative learning procedures such as SGD do not scale to large numbers of training iterations due to the fundamental composition theorem of Dwork and Roth (2014) that causes the required privacy budget to accumulate across iterations.

In this work, we take an approach where models are shared by ensembling, and only the final predictions are made public. This allows circumventing most of the above mentioned problems.

1. Ensembling allows privacy protection by output perturbation via a Laplace mechanism. This mechanism is independent of the size of the component networks.

2. In most application cases, outputs of component networks that contribute to an ensemble are either naturally bounded in a certain range or can be thresholded without loss of generality.

3. Ensembling is a one-step process that is applied to the outputs of component networks. It is thus not affected by an accumulation of required privacy budgets across training iterations.

The basic mathematical details of differential privacy are described in Appendix C. For our work, we especially build upon work on differentially private mean estimators (Epasto et al., 2020; Bun and Steinke, 2019) and formalize a privacy-protected weighted averaging algorithm for ensembling as follows. Let $\hat{f}_i : \mathcal{D} \rightarrow \mathbb{R}, i = 1, \ldots, N$ be functions approximated by N component neural networks. Furthermore, assume bounded outputs $f_i \in [0, B], i = 1, \ldots, N$. Lastly, assume an ensembling technique that combines components by weighted averaging, with weights $w_i \in \mathbb{R}_{\geq 0}, i = 1, \ldots, N$ and $\sum_{i=1}^{N} w_i = 1$. Then Algorithm 1 protects the privacy of an ensemble of N neural networks simply by applying the Laplace mechanism for output perturbation in the averaging phase:

```
Algorithm 1: Private Weighted Averaging Ensemble
Input: Outputs of component networks $\hat{f}_1, \ldots, \hat{f}_N$, combination weights $w_1, \ldots, w_N$
$\hat{f} \leftarrow w_1 \hat{f}_1 + \cdots + w_N \hat{f}_N$
$\tilde{f} \leftarrow \hat{f} + \mathrm{Lap}\left(\frac{B \cdot \max_{i=1,\ldots,N} w_i}{\epsilon}\right)$
return $\tilde{f}$
```

Lemma 3 *For every set of component networks f_i and weights w_i, $i = 1, \ldots, N$, Algorithm 1 is $(\epsilon, 0)$-differentially private.*

Proof The ℓ_1-sensitivity $\Delta \hat{f}$ of the weighted averaging function $\hat{f}$ is $B \cdot \max_{i=1,\ldots,N} w_i$. This allows applying a Laplace mechanism to construct a randomized algorithm $\mathcal{A} = \tilde{f}$ with noise drawn from $\mathrm{Lap}(\Delta \hat{f} / \epsilon)$. By Theorem 3.6 of Dwork and Roth (2014), we know that the Laplace mechanism preserves $(\epsilon, 0)$-differential privacy. ■

As can be seen from the use of a Laplace mechanism, sensitivity of the ensemble output is minimized by choosing uniform weights $w_i = 1/N$. Uniform weights effectively reduce the ℓ_1-sensitivity $\Delta\hat{f}$ of an ensemble $\hat{f}$ by a factor of $1/N$ compared to single models ($w_i = 1$), including the full model that is trained on all available training data. Furthermore, these weights guarantee privacy protection at perturbation with minimal variance. This theoretical advantage of ensembles in privacy protection is confirmed in the experiments presented in Section 4.

3.4. Algorithm

The method implemented in our work can be characterized as a "bucket of models" where we select a pool of individual learners that performed best on a validation set. The idea is to identify prototypical models that represent certain types of patients whose properties can be transferred well to other patients. Our bucket of models consists of a number of individual models that were selected based on the criterion defined by Perrone and Cooper (1992). By looking at the *misfit* of function $\hat{f}_i$, which is the deviation from the true solution, $m_i := f(x) - \hat{f}_i(x)$, the algorithm adds a new candidate model f_{new} to the ensemble $\hat{f}_{em}$ if the candidate satisfies the following inequality:

$$(2N+1)\text{MSE}[\hat{f}_{em}] > 2\sum_{i\neq\text{new}} \mathbb{E}[m_{\text{new}}m_i] + \mathbb{E}[m_{\text{new}}^2] \tag{1}$$

The left part of the RHS's sum expresses that the candidate has to be reasonably different to already included models while the right part of the sum makes sure that the candidate has low error on the validation set, hence the inclusion of $\hat{f}_{new}$ improves generalization and reduces error of the ensemble. The total number of models in the ensemble is not fixed and depends on the performance of the trained models, their diversity, and the validation set.

Our final algorithm is listed in Algorithm 2. This is a greedy algorithm which adds a new model in each step until Inequality 1 cannot be satisfied by any remaining model. It should be noted that a greedy algorithm does not guarantee to return the best performing ensemble, which can only be determined by an exhaustive search over all $2^N - 1$ model combinations.

Our pool of models contains sepsis and non-sepsis patients' models, however, non-sepsis models are often more similar to each other and have a low error, because they usually predict a severity score between 0 and 1 and the number of non-sepsis patients exceeds the number of sepsis patients by more than a factor of 3 (see Table 2). At the same time, our main objective is sepsis prediction, thus we prioritize the selection of sepsis models in function `nextModel` by first going through the list of sepsis models and only if no suitable sepsis model is found, we then go through the list of non-sepsis models. We omitted this prioritization strategy in Algorithm 2 for reasons of clarity.

We also compare two different weighting schemes for our ensemble. In the uniformly weighted case, the weights w_i are always set to $\frac{1}{N}$:

$$y_T = \sum_{i=1}^{N} w_i \cdot y^{(i)} \tag{2}$$

The prediction y at time point T is simply the arithmetic mean of all predictions of the M individual models. As demonstrated in Section 3.3, this weighting scheme delivers the

Algorithm 2: Greedy Ensemble Growing

```
Input: List of patient models $P$ sorted non-decreasing by MSE
Output: Final ensemble $\hat{f}_{em}$
$\hat{f}_{em} \leftarrow$ initialize ensemble
repeat
    $f_{new} \leftarrow$ nextModel($\hat{f}_{em}, P$)
    if $f_{new}$ is found then
        $\hat{f}_{em} \leftarrow \hat{f}_{em} + f_{new}$
    end
until $\hat{f}_{em}$ stops growing

Function nextModel($\hat{f}_{em}$, $P$):
    for $f_{new}$ in $P$ do
        if $(2N+1)MSE[\hat{f}_{em}] > 2\sum_{i \neq new} \mathbb{E}[m_{new} m_i] + \mathbb{E}[m_{new}^2]$ then
            $P \leftarrow P - f_{new}$                          // remove $f_{new}$ from $P$
            return $f_{new}$
        end
    end
return not found
```

best tradeoff between privacy and accuracy by employing privacy protection with minimal variance.

We additionally employ a more sophisticated method for combining learners that uses a weighting scheme based on the model predictions and previous expert labels. Here, the weights of each individual model is determined by the accuracy at which the model was able to predict the patient's previous labels. This method is connected to the *mixture of experts* strategy (Jacobs et al., 1991), but instead of using a gating mechanism for selecting the best experts we apply a soft weighting scheme to get the optimal combination. Here, the weights w_i for Equation 2 are calculated using the following expression:

$$w_i^{(T)} = \frac{1}{C} \sum_{t=1}^{T-1} \frac{1}{1 + |y_t^{(i)} - \hat{y}_t|^2}$$

In words, the weight $w_i^{(T)}$ reflects the accuracy by which model i predicted the label in previous time steps (1 to $T-1$). The value $1/C$ is a normalization factor such that $\sum w_i^{(T)} = 1$. The idea of this weighting scheme is motivated by our label collecting method: On each day, the senior physicians assign labels to each patient in the intensive care unit (ICU). Thus, a theoretical online-learning algorithm has access to previous expert labels and this information can be exploited to tune weights without retraining the model.

The code for training, inference, and evaluation of the sepsis prediction model as well as the implementation of the ensemble growing strategy described in Algorithm 2 is available on `github`.[2]

[2] https://github.com/StatNLP/sepens/

4. Experiments

In this section, we define the patient cohort and how features and labels were obtained for the sepsis prediction task. We compare the fully trained model and the ensemble in terms of prediction accuracy in AUROC, and we empirically show the ensemble's insensitivity to privacy leakage during a membership attack, and evaluate its prediction accuracy with respect to a given privacy budget.

4.1. Data Cohort

Our data is based on a PDMS system running at the University Medial Centre in Mannheim, Germany (UMM). The UMM is a 1,352-bed tertiary care center operating a 22-bed interdisciplinary surgical ICU. The hospital is a center of the Acute Respiratory Distress Syndrome (ARDS) Network Germany. Timelines of clinical measurements were extracted from the Intellispace Critical Care and Anesthesia (ICCA) system by Philips (Eindhoven, Netherlands). Additional demographic patient data as well as ICU admission and discharge times were extracted from a HIS system by SAP (Walldorf, Germany).

Timelines of 42 features were extracted from the ICCA system, and 1 demographic feature, namely age, was extracted from the HIS system. Other demographic features such as gender did not improve performance of our predictive models. See Table 5 in Appendix D for the list of features we use for training our models.

At the beginning of an admission many clinical measurements are not available. Such measurements are set to standard default values defined by a clinical expert. To account for varying intervals of clinical measurements during hospital stay, we apply a carry-forward strategy where the most recent value is "carried forward" until a new value is available. Based on the time lines of varying intervals, we discretize the time lines into uniform steps of 30 minutes during the patient's ICU stay. All values are standardized by calculating z-scores, i.e. $z = \frac{x-\mu}{\sigma}$, where μ is the mean and σ is the standard deviation of the population.

4.1.1. Expert Labels

Sepsis is a complex concept with a wide range of clinical symptoms. Established definitions aim to operationalize this concept by combining an suspected or existing infection with clinical conditions such as SIRS (Sepsis-1/2) or SOFA (Sepsis-3). These definitions are very important in clinical practice, however, they can introduce problems of circularity for machine learning models if the criteria defining a condition are used to predict the very same condition. The problem of circularity in machine learning has been discussed in a broader context in Riezler and Hagmann (2022), and for the specific problem of sepsis prediction in Schamoni et al. (2019). We thus established a questionnaire that collects expert opinions on a daily basis (Lindner et al., 2022). The questionnaire concerns several aspects in ICU practice and was developed in close cooperation with the senior physicians at the ICU. The main goal of the questionnaire is to capture expert opinions that are often based on complex clinical concepts and are thus not fully reflected by established operationalizations. On every day, we ask the senior intensivists to assign a current working diagnosis that is not based purely on clinical criteria, but on their experience and their own opinion. The working diagnoses are put on a 5-point scale where 0 stands for "Neither SIRS nor Sepsis", and 4 stands for "Septic Shock". See Table 1 for the complete list of working diagnoses.

Table 1: List of possible expert labels for current working diagnoses.

Value	**Working diagnosis**
0	Neither SIRS nor Sepsis
1	SIRS
2	Sepsis
3	Severe Sepsis
4	Septic Shock

The questionnaire is filled out on a daily basis at 2 p.m. At the beginning of the survey, the labels were assigned for the preceding 24h window, i.e., from 2 p.m. on the preceding day to 2 p.m. on the current day. Later in the survey, the senior intensivists were asked to set a 6h window of change if a given label differs from the previous label. These new 6h-intervals divide the 24 h window previously in use. To balance the error introduced by the difference of the true sepsis onset to the sepsis labeling time, we set the assumed time of sepsis onset to the center of the interval, e.g., to 2 a.m. for the 24h window, 5 a.m. for the 2 a.m.–8 a.m. window, 11 a.m. for the 8 a.m.–2 p.m. window, 5 p.m. for the 2 p.m.–8 p.m. window, and 11 p.m. for the 8 p.m.–2 a.m. window.

4.1.2. Data Filtering and Splitting

Our goal is to learn a model that is able to make timely and accurate predictions on a patient's sepsis outcome. As we are interested in prediction, we excluded on-admission sepsis cases where we defined an on-admission case as a patient who received the first sepsis expert label within the first 48h after ICU admission. We also removed patients that stayed less than 16h in the ICU. Both filtering steps reduced the total number of patients in our cohort from 1,961 to 1,275. To address the problem of having very different distributions in train, validation and test splits, we employed a sampling scheme where we first sort non-sepsis and sepsis patients by length of stay and sepsis onset, respectively, and then sample from groups of four consecutive patients to randomly assign them to four partitions, namely A, B, C, and D. The results of our sampling scheme are listed in Table 2.

Table 2: Distribution of hospital stay times and sepsis onset in hours for sepsis and non-sepsis patients across the four partitions.

	Patients	**Non-sepsis pat. stay [h]**	**Sepsis pat. onset [h])**
Partition	**(Non-/Sepsis)**	**Median/Min/Max**	**Median/Min/Max**
A	319 (245/74)	56.5/15.5/2253.0	159.0/50.0/1394.0
B	319 (245/74)	55.5/15.5/1218.5	162.0/49.0/972.5
C	319 (245/74)	56.0/15.5/1618.5	161.0/51.0/1257.0
D	318 (244/74)	56.0/15.5/831.5	161.5/48.5/1269.5
Total	1275(979/296)	–	–

In our experiments, we applied a 4-fold cross validation scheme where we used two splits as train data, one as validation data, and the final one as test data. In detail, the train data consists of partitions A+B, B+C, C+D, D+A, and the validation and test sets are partitions C and D, D and A, A and B, and B and C, respectively. Table 2 lists statistics of patients and hospital stay times in the partitions. The preprocessed data splits are available for download.[3]

4.1.3. Ethics

Ethics approval was obtained from the Medical Ethics Commission II of the Medial Faculty Mannheim, Heidelberg University, Germany (reference number, 2016-800R-MA).

4.2. Experimental Results

In this section, we compare prediction accuracy and privacy of ensemble and full models. We show how various privacy budgets affect accuracy loss, and how ensembles successfully prevent privacy leakage in the context of a membership attack.

4.2.1. Prediction Accuracy of Ensemble and Full Model

In our first experiment, we compare a model that was trained on all available data to an ensemble grown using Algorithm 2. Table 3 lists the resulting ensemble sizes and the ratio of sepsis and non-sepsis patients in our final ensembles per split. While the ensembles are of size 40.5 on average, the numbers range from 20 for split-1 to 65 for split-2, which is 50% less and 60% more than the average, respectively. The ratio between non-sepsis and sepsis patient models shows a similar high variance ranging from 0.333 to 0.85 depending on the data split. This illustrates that although we tried to make the data splits as similar as possible, the number of patients is still small and individual patients can have a large influence on the composition of the final model.

Table 3: Comparison of total data and resulting ensemble sizes using the growing strategy described in Algorithm 2.

	split-0	**split-1**	**split-2**	**split-3**	**average**
Total data					
# train patients	638	638	637	637	–
# dev patients	319	318	319	319	–
# test patients	318	319	319	319	–
Ensemble sizes					
# total models	37	20	65	40	40.50
# non-sepsis models	17	5	30	15	16.75
# sepsis models	20	15	35	25	23.75
non-sepsis/sepsis ratio	0.85	0.33	0.85	0.60	0.71

The metric to compare the predictive performance of our models is the area under the receiver operator characteristic curve (AUROC). The AUROC represents the curve of sensitivity-specificity pairs at particular decision thresholds. In the case of time series with

[3] https://www.cl.uni-heidelberg.de/statnlpgroup/sepsisexp/

prediction intervals, a common practice of calculating the AUROC is to consider sensitivity and specificity at all timesteps of interest, that is the interval itself and the preceding time of hospital stay. We follow the standard procedure of evaluating only the first sepsis episode, as subsequent episodes have very different properties due to interventions such as medication, administration of fluids, etc. Significance levels were computed using a two-sample t-test over the means of the two populations. Means are calculated over four cross-validation runs.

When comparing the predictive performance in terms of AUROC, the ensembles are remarkably better than the full model. For the uniform ensemble (*ensemble-u*), the difference is significant according to a t-test at $p < 0.05$ for all but one case, that is predicting sepsis 12h before onset using the uniform model (see Table 4). We attribute this to the growing strategy of Algorithm 2 which guarantees to improve generalization based on theoretical results discussed in Section 3.2. The ensemble with weights adjusted due to the history of prediction performance (*ensemble-w*), we observe larger gains for the shorter prediction times. When moving further away from sepsis onset, the prediction performance of the uniform and the weighted ensemble becomes more similar. This might be influenced by the fact that the expert labels have limited time resolution (24h and 6h) such that the prediction interval is closer to the real onset time than the interval values indicate.

Table 4: AUROC of the ensemble for predicting sepsis at different times before onset. Here, the ensemble is generated using Algorithm 2 in Section 3.4. The preceding * denotes statistically significant difference ($\alpha = 0.05$) compared to the full model. Values in parenthesis are standard deviations across data splits.

	4h	**8h**	**12h**	**12h–8h**	**24h–12h**
full model	70.80 (1.84)	69.48 (1.56)	67.99 (2.08)	68.75 (1.85)	67.13 (2.10)
ensemble-u	*76.15 (2.66)	*73.47 (2.62)	70.51 (3.02)	*72.12 (2.86)	*70.77 (2.93)
ensemble-w	*78.13 (1.07)	*74.61 (1.67)	*70.67 (2.00)	*72.77 (1.90)	*70.63 (1.93)

4.2.2. Privacy and Accuracy: Attacks and Defenses

Differential privacy has become a privacy standard for privacy-preserving machine learning. However, there exist many forms of differential privacy with different theoretical privacy guarantees. Jayaraman and Evans (2019) provide a detailed overview on various differentially private machine learning methods in practice. They empirically evaluate two types of privacy attacks, *membership inference* and *attribute inference*. We are mostly interested in the former attack, *membership inference*, as it is most relevant for our method of ensembling patient-specific models.

We apply a simple but effective membership inference attack that has been described by Yeom et al. (2018). Here, it is assumed that the attacker has access to the average training loss. While this is not the case in general, an attacker might obtain this single number by a security breach. The attacker could also estimate the average training loss if there exists some knowledge about the original training data distribution. To infer membership of a data point, the attacker feeds the data to the model and receives the corresponding prediction. Then, by comparing the prediction to the gold label, the attacker calculates the error (or loss) on this example. Finally, if the calculated loss of the example is smaller than the average training loss, then the example is considered to have been part of the training set.

In our first experiment on membership attack, we compare the *privacy leakage* (Jayaraman and Evans, 2019) of the full model and the uniform ensemble at different privacy budgets ϵ. The privacy leakage, also known as *attacker's advantage* (Yeom et al., 2018), is defined as the difference of the true positive rate (TPR) and false positive rate (FPR) of a membership attack. For each evaluated privacy budget, we calculate FPR as the ratio of false positives from the unseen test set, and TPR as the ratio of true positives from the training set. To get additional error estimates, we keep the smaller group fixed and sample sets of an equal size 1,000 times from the larger group, i.e., we fix the test set (negatives) and sample from the training set (positives) 1,000 times to calculate our statistics. Figure 2 illustrates that the full model causes privacy leakage at $\epsilon > 10$ while the uniform ensemble is unaffected in terms of membership inference success on all evaluated levels ($10^{-3} \leq \epsilon \leq 10^{3}$).

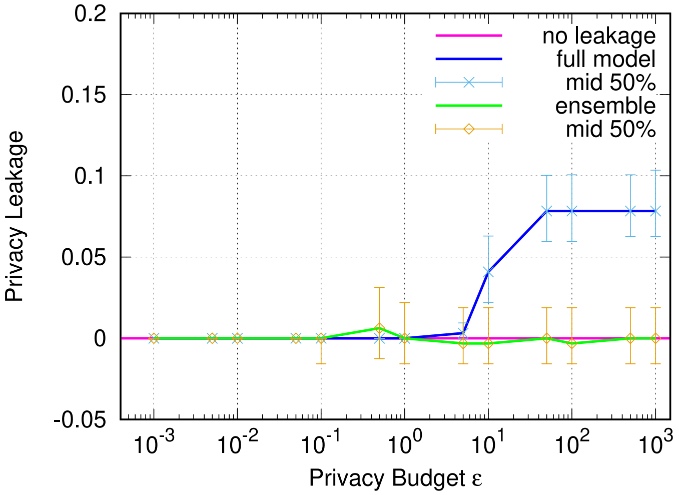


Figure 2: Privacy leakage for membership inference attacks on the full model and on the uniform ensemble at different ϵ-levels. Privacy leakage is defined as TPR − FPR, thus it can get values below 0. The full model shows privacy leakage at values $\epsilon > 10$, while the ensemble preserves privacy at all evaluated ϵ-levels. The vertical bars denote the 2^{nd} and 3^{rd} quartile and the median.

In our second experiment, we empirically evaluate the tradeoff between privacy and accuracy which has been discussed in Section 3.3. Privacy is achieved by injecting noise to the model's prediction. This, however, decreases accuracy, so it is important to know how much noise can be injected without sacrificing too much accuracy. We adopt the idea of accuracy loss by Jayaraman and Evans (2019) and extend it to our AUROC metric. While accuracy typically ranges from 0 to 1, the AUROC ranges from 0.5 (random guessing) to 1.0 (every prediction correct). We thus employ an AUROC-adapted accuracy loss calculation:

$$Accuracy\ loss_{AUROC} = 1 - \frac{(2 \cdot AUROC\ of\ Private\ Model) - 1}{(2 \cdot AUROC\ of\ Non\text{-}Private\ Model) - 1}$$

For each ϵ-level and for each prediction interval, we calculate the accuracy loss for both the full model and the uniform ensemble. Throughout all intervals, our experiments indicate that the ensemble's accuracy degrades at privacy budget levels approximately two orders of magnitude smaller than the full model's accuracy. For smaller intervals (Figure 3 a,b,c) the variance between different data splits is high, while for wider prediction intervals (Figure 3 d,e) the variance is smaller, quantile markings are closer, and curves are smoother.

5. Discussion and Conclusion

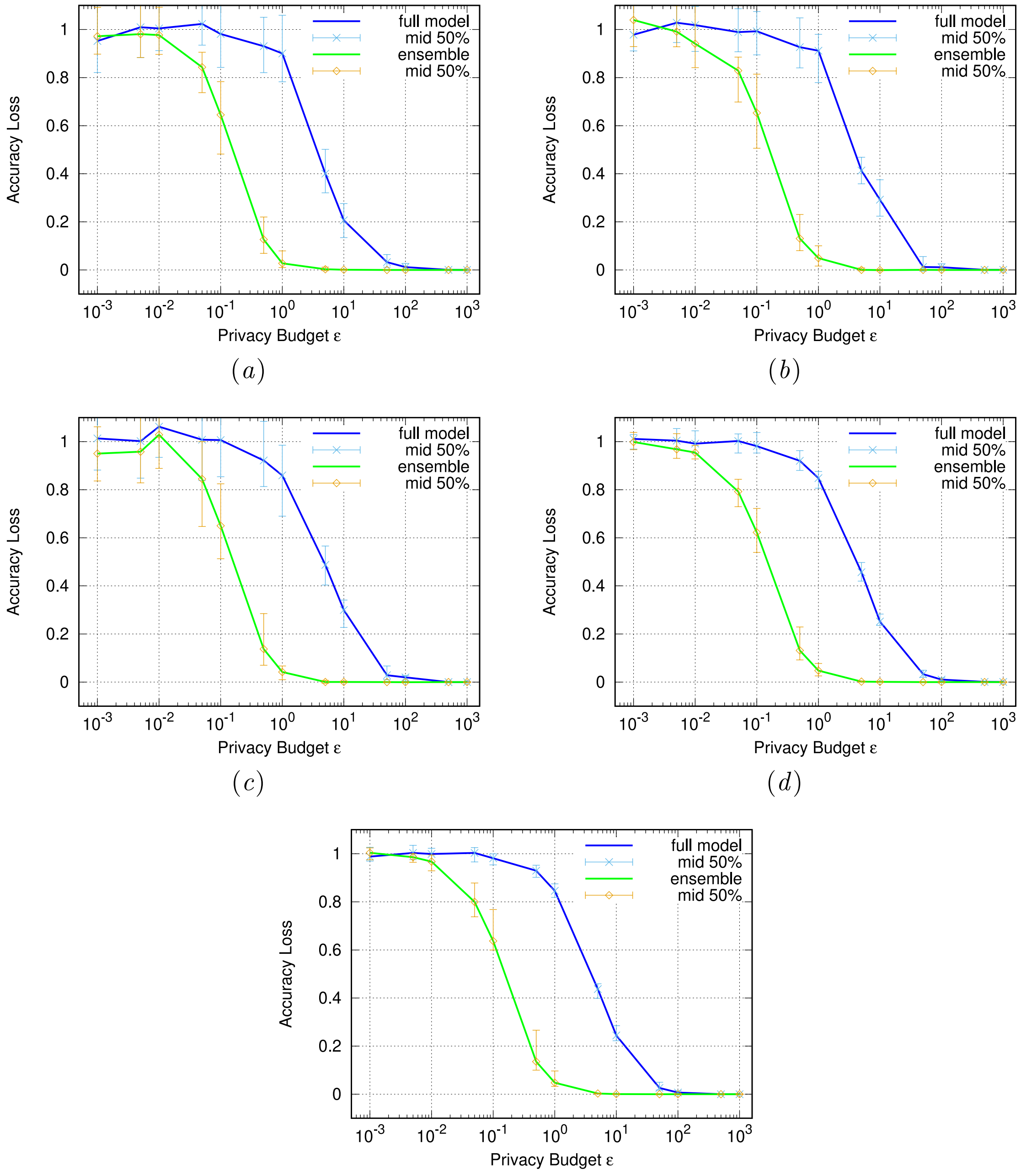
We have shown how to train and combine individual personalized models in an ensemble that has highly desirable properties in the field of medical data analysis. We presented theoretical and empirical results that show that an ensemble of models which were trained on a fraction of the available data can be better than a baseline model that was trained on all data. We applied a strong membership attack and showed that the ensemble successfully prevents privacy leakage while maintaining its accuracy at privacy budgets almost two orders of magnitude smaller than a single fully trained baseline model.

An important type of attack that we did not explicitly evaluate is attribute inference. The techniques for attribute inference, however, are indeed based on membership inference. In attribute inference, the attacker constructs multiple variants of a candidate patient with and without the attribute in question. Then, similar to membership inference, the prediction error of the model tells the attacker which variant was used in the training dataset. Thus, models that successfully prevent membership inference are also able to prevent attribute inference.

In the field of medical data analysis, ensembling is a promising method for prediction tasks. Explicit sharing of models across different medical communities is desirable and is performed for example in federated learning. However, these techniques are in practice difficult to implement due to various legal constraints and other circumstances such as incompatible interfaces. Protecting privacy in federated learning usually requires very careful injection of noise in the gradients or in the models to not sacrifice performance, which often results in high privacy costs (Jayaraman and Evans, 2019). Sharing models implicitly by ensembling and publicly sharing only outputs, where the models remain in a secured, inaccessible area, is a more realistic setup where privacy can be controlled efficiently.

One limitation of our privacy preserving approach is that we still consume a small privacy budget for each query. Thus, given a fixed privacy budget, subsequently querying our ensemble will eventually use up all the privacy budget available. This problem is addressed by the Private Aggregation of Teacher Ensembles (PATE) approach (Papernot et al., 2017), where an ensemble of teacher models is used to annotate incomplete public data and train a student model on the annotated data. In doing so, the consumed privacy budget does not increase once the student model is trained. In principle, our ensemble strategy could be easily extended to a PATE-like scenario.

In this work, we follow the standard mechanism for privacy protection by adding randomness at certain locations in the machine learning model. Another way of improving privacy could be to exploit the intrinsic randomness of ensembling algorithms such as subsample-and-aggregate (Nissim et al., 2007; Jordon et al., 2019) or bagging (Liu et al., 2021). An interesting direction for future research is an investigation of data augmentation



(a) (b) (c) (d) (e)

Figure 3: Accuracy loss of the full model and of the uniform ensemble at different ϵ-levels and prediction times before Sepsis onset. Subfigures (a), (b), and (c) show the accuracy loss for the 4h, 8h, and 12h prediction time, subfigures (d) and (e) for the 12h–8h and the 24h–12h prediction intervals, respectively. The vertical bars denote the 2^{nd} and 3^{rd} quartile and the median.

techniques for privacy protection (Yun et al., 2019; Lam et al., 2022). In these approaches, data are randomly cut and recombined to generate artificial training data that have the potential to protect the privacy of the original data.

The dataset we use throughout our experiments is quite imbalanced, i.e., the ratio between sepsis and non-sepsis patients is 1:3.3 (see Table 2). This is mainly due to our filtering where we remove a large amount of short term ICU non-sepsis patients. We found that the ensemble growing strategy we propose does handle imbalances at the ratio mentioned above very well. In reality, however, ICUs often observe much stronger imbalanced data and whether our method degrades at higher ratios or not is an open question. One aspect that implicitly reduces such imbalances is the fact that in general sepsis patients have longer hospital stay times than non-sepsis patients and thus provide more data points to the model.

The ensemble growing strategy described in Algorithm 2 provably improves generalization of the ensemble, but as a greedy algorithm it cannot guarantee to return the optimal model combination. Other methods that jointly train an ensemble (Buschjäger et al., 2020) or apply a non-greedy strategy (Zhou et al., 2002) might return better ensembles at the cost of increased complexity. Our simple growing strategy, however, means that our ensembles can be easily updated when new patients' data becomes available: a new model needs to be trained on the new data, and Algorithm 2 will integrate the model in the ensemble if it performs well on the validation set and if it is reasonably different to the existing models.

Privacy protection is of increasing importance in the growing field of medical data science. Machine learning models highly benefit from increasing amounts of data, which can potentially compromise the patients' rights if techniques are applied without the privacy aspect in mind. There is a lot of active research in model sharing techniques such as federated learning, however, we demonstrate that output sharing such as ensembling is a simple and effective method to provide strong privacy guarantees without sacrificing performance. We don't propose that model sharing and output sharing should be mutual exclusive; at some levels, model sharing might be better applicable, for example in protected in-house scenarios. In other scenarios, where privacy is defined by differing regulations or laws, for example in a national or international context, output sharing might provide an avenue that is simpler to implement and at the same time provides very strong privacy guarantees.

6. Acknowledgements

This research has been conducted in project SCIDATOS (Scientific Computing for Improved Detection and Therapy of Sepsis), funded by the Klaus Tschira Foundation, Germany (Grant number 00.0277.2015).

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Vienna, Austria, 2016. doi: 10.1145/2976749.2978318.

Samuel Adams, Chaitali Choudhary, Martine De Cock, Rafael Dowsley, David Melanson, Anderson C. A. Nascimento, Davis Railsback, and Jianwei Shen. Privacy-preserving training of tree ensembles over continuous data. *Proc. Priv. Enhancing Technol.*, 2022 (2):205–226, 2022. doi: 10.2478/popets-2022-0042.

Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996. doi: 10.1007/BF00058655.

Mark Bun and Thomas Steinke. Average-case averages: Private algorithms for smooth sensitivity and mean estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 181–191, Vancouver, Canada, 2019.

Sebastian Buschjäger, Lukas Pfahler, and Katharina Morik. Generalized negative correlation learning for deep ensembling. *CoRR*, abs/2011.02952, 2020.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069—-1109, 2011.

Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM Symposium on Principles of Database Systems (PODS)*, San Diego, California, 2003. doi: 10.1145/773153.773173.

Daniel Durstewitz, Quentin J. M. Huys, and Georgia Koppe. Psychiatric illnesses as disorders of network dynamics. *Biol Psychiatry Cogn Neurosci Neuroimaging*, 6(9):865–876, 2021. doi: 10.1016/j.bpsc.2020.01.001.

Cynthia Dwork. Differential privacy. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, Venice, Italy, 2006.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Alessandro Epasto, Mohammad Mahdian, Jieming Mao, Vahab Mirrokni, and Lijie Ren. Smoothly bounding user contributions in differential privacy. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, virtual, 2020.

Yoav Freund and Robert E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the European Conference on Computational Learning Theory (EuroCOLT)*, Barcelona, Spain, 1995.

Kyle Fritchman, Keerthanaa Saminathan, Rafael Dowsley, Tyler Hughes, Martine De Cock, Anderson Nascimento, and Ankur Teredesai. Privacy-preserving scoring of tree ensembles: A novel framework for ai in healthcare. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2413–2422, 2018. doi: 10.1109/BigData.2018.8622627.

Mudasir A. Ganaie, Minghui Hu, Mohammad Tanveer, and Ponnuthurai N. Suganthan. Ensemble deep learning: A review. *CoRR*, abs/2104.02395, 2021.

Renee Garett and Sean D. Young. Ethical views on sharing digital data for public health surveillance: Analysis of survey data among patients. *Frontiers in Big Data*, 5, 2022. ISSN 2624-909X. doi: 10.3389/fdata.2022.871236.

Kim Huat Goh, Le Wang, Adrian Yong Kwang Yeow, Hermione Poh, Ke Li, Joannas Jie Lin Yeow, and Gamaliel Yu Heng Tan. Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. *Nature Communications*, 12(711), 2021.

Yanmin Gong, Yuguang Fang, and Yuanxiong Guo. Privacy-preserving collaborative learning for mobile health monitoring. In *IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, USA, 2015. doi: 10.1109/GLOCOM.2015.7417841.

L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990. doi: 10.1109/34.58871.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John Hopcroft, and Kilian Weinberger. Snapshot ensembles: Train 1, get m for free. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence (UAI)*, Monterey, CA, USA, 2018.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *Proceedings of the 28th USENIX Conference on Security Symposium (SEC'19)*, Santa Clara, CA, USA, 2019.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Differentially private bagging: Improved utility and cheaper privacy than subsample-and-aggregate. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, pages 4325–4334, Vancouver, Canada, 2019.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. Sample, translate, recombine: Leveraging audio alignments for data augmentation in end-to-end speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 245–254, Dublin, Ireland, 2022. ACL.

H. A. Lindner, S. Schamoni, T. Kirschning, C. Worm, B. Hahn, F. S. Centner, J. J. Schoettler, M. Hagmann, J. Krebs, D. Mangold, S. Nitsch, S. Riezler, M. Thiel, and V. Schneider-Lindner. Ground truth labels challenge the validity of sepsis consensus definitions in critical illness. *Journal of Translational Medicine*, 20(6):27, 2022. doi: 10.1186/s12967-022-03228-7.

Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. On the intrinsic differential privacy of bagging. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2730–2736, Virtual Event / Montreal, Canada, 2021. ijcai.org. doi: 10.24963/ijcai.2021/376.

Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10): 1399–1404, 1999. doi: https://doi.org/10.1016/S0893-6080(99)00073-8.

H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

Michael Moor, Nicolas Bennett, Drago Plecko, Max Horn, Bastian Rieck, Nicolai Meinshausen, Peter Bühlmann, and Karsten M. Borgwardt. Predicting sepsis in multi-site, multi-national intensive care cohorts using deep learning. *CoRR*, abs/2107.05230, 2021.

Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 75–84, San Diego, CA, USA, 2007. ACM. doi: 10.1145/1250790.1250803.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, 2017.

Michael P. Perrone and Leon N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In *Artificial Neural Networks for Speech and Image Processing*, pages 126–142. Chapman-Hall, New York, 1992.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. *npj Digital Medicine*, 3(1), December 2020. ISSN 2398-6352. doi: 10.1038/s41746-020-00323-1.

Stefan Riezler and Michael Hagmann. *Validity, Reliability, and Significance: Empirical Methods for NLP and Data Science*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2022. ISBN 9781636392714. doi: 10.2200/S01137ED1V01Y202110HLT055.

Shigehiko Schamoni, Holger A. Lindner, Verena Schneider-Lindner, Manfred Thiel, and Stefan Riezler. Leveraging implicit expert knowledge for non-circular machine learning in sepsis prediction. *Artif. Intell. Medicine*, 100, 2019. doi: 10.1016/j.artmed.2019.101725.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 53rd Annual Allerton Conference on Communication, Control, and Computing (CCS'15)*, Allerton, IL, USA, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, Long Beach, CA, USA, 2017.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *31st IEEE Computer Security Foundations Symposium (CSF 2018)*, pages 268–282, Oxford, United Kingdom, 2018. IEEE Computer Society. doi: 10.1109/CSF.2018.00027.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6022–6031, Seoul, Korea (South), 2019. IEEE. doi: 10.1109/ICCV.2019.00612.

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137(1-2):239–263, 2002. doi: 10.1016/S0004-3702(02)00190-X.

Appendices

A. Model Architecture and Meta-Parameters

The details of our general model architecture are as follows. Our model uses LSTM-cells (Hochreiter and Schmidhuber, 1997) to model long and short dependencies in the time series data. Each patient's stay is divided into 24h windows with 12h of overlap. For example, a 48h stay will be divided into three 24h windows during training. The motivation for using overlapping windows here is that important changes in clinical measurements should not solely occur on one end of a window, but also in the middle of such a sequence so the model has access to more context. Our model has 2 hidden LSTM-layers with 200 units each. The input layer takes a 43-dimensional feature vector, the output is the severity score. We train the model with a minibatch size of 20 with mean squared error as the optimization metric. We also apply gradient clipping of 0.25 and set dropout to 0.2 for the hidden layers during training.

B. Optimality in Weighted Averaging Ensembling

Lemma 4 *Let $\mathfrak{F} = \{\hat{f}_i : i = 1, \dots, N\}$ be a set of regression estimates with covariance matrix $\mathbf{C} = [C_{ij}]_{i,j=1,\dots,N}$. Then choosing the weights according to*

$$\mathbf{w}^* = \frac{\mathbf{C}^{-1}\mathbf{1}_N}{\mathbf{1}_N^\intercal \mathbf{C}^{-1}\mathbf{1}_N}$$

where $\mathbf{1}_N$ is an N-vector whose components are all 1 yields the ensemble estimator with the lowest possible MSE for $\mathfrak{F}$.

Note:
If $\mathbf{C}$ is a diagonal matrix then the optimal weights are

$$w_i^* = \frac{\frac{1}{\mathrm{MSE}(\hat{f}_i)}}{\sum_{j=1}^{N} \frac{1}{\mathrm{MSE}(\hat{f}_j)}} \ .$$

If further $\mathrm{MSE}(\hat{f}_1) = \ldots = \mathrm{MSE}(\hat{f}_N)$ the optimal weights are

$$w_i^* = \frac{1}{N} \ .$$

An important consequence of this Lemma is the fact that if the weights are chosen in an optimal way, then $\mathrm{MSE}(\hat{f}_{em}) \leq \min_{j=1,\ldots,N} \mathrm{MSE}(\hat{f}_j)$.

Summary:

1. The MSE of an ensemble estimator of a collection $\mathfrak{F}$ is completely determined by the covariance matrix $\mathbf{C}$.

2. In practice $\mathbf{C}$ is unknown and must be replaced by an estimator $\hat{\mathbf{C}}$. In the usual setting where training, test and validation data are drawn from the same distribution the covariances C_{ij} can be estimated by validation set sample means.

3. Regarding the optimal weights, the invertibility of $\mathbf{C}$ is of direct importance. Given that in practice we need to base our calculation on $\hat{\mathbf{C}}$ which should aim at choosing $\mathfrak{F}$ in such a way that the inversion of $\mathbf{C}$ (and in consequence $\hat{\mathbf{C}}$) is numerically stable and well conditioned. One way to achieve this is to choose $\mathfrak{F}$ in such a way that $\mathbf{C}$ is a diagonal matrix.

C. Differential Privacy

Differential privacy (Dwork, 2006; Dwork and Roth, 2014) is based on guarantees that a randomized algorithm behaves similarly on similar input databases, i.e., on databases differing in one data point. Let $D, D' \in \mathcal{D}$ be two data sets that are obtained from one another by removing one data point, called neighboring data sets, and denoted by $D \sim D'$. Furthermore, let $\mathcal{A}$ be a randomized algorithm producing an output in the space $\mathcal{O}$ on input data in $\mathcal{D}$.

Definition 5 *A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all data sets $D, D' \in \mathcal{D}$, and all subsets of outcomes $S \subseteq \mathcal{O}$,*

$$Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in S] + \delta,$$

where ϵ is the privacy budget, and δ is the failure probability.

The concept of differential privacy implies that the level of protection of data is always lowered when a model trained on this data is queried. The privacy budget ϵ is thus reduced by each model query. A common way to lower this reduction is to add noise to the model's answer of the query.

Let us consider deterministic functions $f : \mathcal{D} \to \mathbb{R}^d$ as fundamental types of database queries. The amount of noise that is required to preserve privacy of a function f is proportional to its sensitivity, i.e., to the maximum change in the output of f over all possible inputs:

Definition 6 *The ℓ_1-sensitivity of a function $f : \mathcal{D} \to \mathbb{R}^d$ is*

$$\Delta f = \max_{D \sim D'} \left\| f(D) - f(D') \right\|_1.$$

A standard technique to achieve differential privacy is the Laplace mechanism that perturbs each coordinate of a function with noise drawn from a Laplace distribution, with variance proportional to the sensitivity of the function (divided by ϵ):

Definition 7 *Given a function $f : \mathcal{D} \to \mathbb{R}^d$, the Laplace mechanism is defined as $f(D) + (W_1, \ldots, W_d)$ where W_i are i.i.d. random variables drawn from* $\mathrm{Lap}(\Delta f/\epsilon)$*, and* $\mathrm{Lap}(s)$ *is a Laplace distribution with mean* 0 *and variance* $2s^2$*.*

D. Features for Sepsis Prediction Task

Table 5: List of the 43 features we used for training our models for the Sepsis prediction task. Features can be readings from vital monitors (e.g., heart rate, blood pressure), lab results (e.g., bilirubin, creatinine), or static demographic features (age).

Age	Arterial pH	Urine output	Procalcitonin (PCT)
Heart rate	Leukocytes	Blood glucose	Δ Temperature
Lactate	Bicarbonate	Stroke volume	Alanine transaminase
Creatinine	Base excess	Horowitz index	BUN/Creatinine ratio
Bilirubin	Lymphocytes	Partial CO_2	Aspartate transaminase
Sodium	Net balance	Respiratory rate	Oxygenation saturation
Potassium	Quick score	Calcium (ionized)	C-reactive protein (CRP)
Hemoglobin	Systolic BP	Heart time volume	Respiratory minute volume
Chloride	Temperature	Oxygen saturation	Fraction of inspired O_2
SVRI	Diastolic BP	Pancreatic lipase	Partial pressure art. O_2
Mean BP	Thrombocytes	Blood urea nitrogen	