Federated learning for violence incident prediction in a simulated cross-institutional psychiatric setting

Citation for published version (APA):

Document license:
CC BY

DOI:
10.1016/j.eswa.2022.116720

Document status and date:
Published: 01/08/2022

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
Federated learning for violence incident prediction in a simulated cross-institutional psychiatric setting

Thomas Borger a,b, Pablo Mosteiro a,∗, Heysem Kaya a, Emil Rijcken c, d, Albert Ali Salah a, f, Floortje Scheepers d, Marco Spruit c, g, a

a Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
b Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey
c Department of Public Health & Primary Care, Leiden University Medical Centre, Leiden, The Netherlands
d Jheronimus Academy of Data Science, Eindhoven University of Technology, ’s-Hertogenbosch, The Netherlands
e KPMG N.V., Amstelveen, The Netherlands
f Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands
g Eindhoven University of Technology, ’s-Hertogenbosch, The Netherlands

A R T I C L E I N F O

Keywords:
Federated learning
Violence prediction
Neural networks
Psychiatry
Clinical notes

ABSTRACT

Inpatient violence is a common and severe problem within psychiatry. Knowing who might become violent can influence staffing levels and mitigate severity. Predictive machine learning models can assess each patient’s likelihood of becoming violent based on clinical notes. Yet, while machine learning models benefit from having more data, data availability is limited as hospitals typically do not share their data for privacy preservation. Federated Learning (FL) can overcome the problem of data limitation by training models in a decentralised manner, without disclosing data between collaborators. However, although several FL approaches exist, none of these train Natural Language Processing models on clinical notes. In this work, we investigate the application of Federated Learning to clinical Natural Language Processing, applied to the task of Violence Risk Assessment by simulating a cross-institutional psychiatric setting. We train and compare four models: two local models, a federated model and a data-centralised model. Our results indicate that the federated model outperforms the local models and has similar performance as the data-centralised model. These findings suggest that Federated Learning can be used successfully in a cross-institutional setting and is a step towards new applications of Federated Learning based on clinical notes.

1. Introduction

Inpatient violence is a serious problem in clinical psychiatry, causing short- and long-term damage to property as well as people (Havaei, MacPhee, & Lee, 2019; Inoue, Tsukano, Muraoka, Kaneko, & Okamura, 2006; van Leeuwen & Harte, 2017; Nijman, Bowers, Oud, & Jansen, 2005). Violence Risk Assessment (VRA) has been used in mental healthcare to inform medical decisions and mitigation strategies (Conroy & Murrie, 2012; Singh et al., 2014). Several manual VRA methods have been proposed and evaluated (Almvik, Woods, & Rasmussen, 2000; Douglas et al., 2014; Ogloff & Daffern, 2006), yet these methods are time-consuming and subjective, and some of them require advanced training to use (Webster, Nicholls, Martin, Desmarais, & Brink, 2006).

Machine Learning (ML) methods promise to address these limitations, developing fast and objective predictions based on patient data present in Electronic Health Records (EHR).

In the psychiatry domain, a particularly promising ML approach is Natural Language Processing (NLP), since EHRs contain large amounts of unstructured clinical notes written by nurses and psychiatrists. The information in these notes could be employed in decision-support systems to aid psychiatrists in predicting aggression, diagnosing patients, predicting side-effects from medication, and predicting suicide attempts, among others. The information is reported in subtle and nuanced ways, and often includes typographical errors, abbreviations and technical terms. Not surprisingly, a common problem encountered by ML researchers in the clinical domain are datasets that are small (Pestian, Nasrallah, Matykiewicz, Bennett, & Leenaars, 2010) or too specific (Suchting, Green, Glazier, & Lane, 2018). Thus, increasing
Multiple Machine Learning (ML) methods have been proposed to tackle the problem of Violence Risk Assessment (VRA). Bader and Evans (2015) attempted to differentiate between patients perpetrating severe and repeated aggression and non-aggressive patients, using common risk factors as predictor variables. In a retrospective study, Raja and Azzoni (2005) found some factors that seemed to correlate with inpatient violence. Menger, Spruit, van Est, Nap, and Scheepers (2019); Le, Montgomery, Kirkby, and Scanlan (2018); and Cook et al. (2016) exploited the abundant free text in EHRs to employ Natural Language Processing (NLP) to this task. Beyond VRA, Pestian et al. (2010) used NLP to classify suicide notes as legitimate or elicited.

Two limiting factors in the development of fair and accurate ML models for the healthcare domain are dataset size and diversity. Of the studies mentioned above, only one had more than a few thousand data points (Le et al., 2018). Its limitation, however, was that they predicted existing VRA instrument scores, not real violence incidents. Suchting et al. (2018) had nearly 30 thousand data points, yet they report being limited both by dataset diversity (due to the nature of their facility) and by dataset size (due to the imbalanced nature of the dataset, as most patients do not engage in violence). Aggregating data from multiple institutions would tackle both problems. However, as medical data often resides in secure data silos across institutions (Lehne, Sass, Essenwanger, Scheipers, & Thun, 2019), aggregating these data is not possible.

Federated Learning (FL) is a novel technique for training ML models on decentralised data (Konečný et al., 2016). It began with the question of how one can train an ML model in a setting where data is unevenly distributed across a large number of devices, and the data cannot be shared among devices or with the central server. FL provides a solution to this question through decentralised training, orchestrated by a central server. The server initialises and sends a model to each participating institution or data silo. Each institution trains the model on their own data, and shares the updated model's parameters with the central server. The server then aggregates all models and creates a new global model. A widely used algorithm for creating a new model is FedAvg (McMahan et al., 2016), which performs a weighted average over the parameters of all models to create a new model. Other algorithms have been proposed to allow the use of adaptive optimisers, such as FedAdagrad, FedYogi, and FedAdam (Reddi et al., 2021).

FL has brought promising results in recent literature, where federated models perform nearly on par with data-centralised models for medical classification tasks, such as brain tumour segmentation (Li et al., 2019; Sheller et al., 2019) and in-hospital mortality prediction (Choudhury et al., 2019). The technique has been applied on private medical data as well by utilising the Personal Health Train (PHT), for classifying post-treatment survival chances in lung cancer patients, by collaborating with eight medical institutions (Deist et al., 2020). PHT is a platform aiming to provide healthcare data from various sources to researchers while ensuring privacy protection. FL has also been used to predict suicidal ideation in online social care texts (Ji et al., 2019). During the literature search conducted at time of study, no applications of FL on models employing clinical texts were identified. Table 1 shows all the aforementioned methods, together with their goals and limitations.

To bring FL to the psychiatric domain, a Natural Language Processing (NLP) task is chosen for this study. Clinical notes have been written about admitted patients on a daily basis for many years across medical institutions. This means that local datasets are available for research at these institutions. Issues with these clinical texts are that they are semi-structured, and sometimes contain thousands up till tens of thousands of words for a single admission period, making feature extraction a difficult task. Menger, Scheipers, and Spruit (2018) compared various methods to convert texts into vectorial representations, including bag-of-words, TF-IDF, Word2Vec and Doc2Vec. Following previous work (Mosteiro et al., 2021), in this paper we use Doc2Vec (Le & Mikolov, 2014), which generates a fixed-length vector for a piece of text of arbitrary length. In this study’s context, a document is the collection of notes of one admission period of a patient. Through this method, the vector representations aims to keep the semantics within each document intact. The representations can then be fed into an ML model such as a neural network for a classification task.
the concatenation of clinical notes of a maximum of 28 days before up until and including the 1st day after admission. Based on the next 27 days following the first day of admission, the data points are labelled by whether the patient is involved in a violence incident during the first 27 days after the first day of the admission period.

### Table 2

<table>
<thead>
<tr>
<th>Nursing ward</th>
<th>Age</th>
<th>Description</th>
<th>Positive</th>
<th>Negative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>&gt; 40</td>
<td>Affective &amp; psychotic disorders</td>
<td>25</td>
<td>734</td>
<td>759</td>
</tr>
<tr>
<td>A3</td>
<td>15–35</td>
<td>Diagnosis &amp; early psychosis</td>
<td>130</td>
<td>696</td>
<td>826</td>
</tr>
<tr>
<td>A2V</td>
<td>&gt; 18</td>
<td>Acute &amp; intensive care</td>
<td>167</td>
<td>1710</td>
<td>1877</td>
</tr>
<tr>
<td>A2J</td>
<td>12–18</td>
<td>Acute &amp; intensive care</td>
<td>103</td>
<td>715</td>
<td>818</td>
</tr>
<tr>
<td>Total</td>
<td>–</td>
<td>–</td>
<td>425</td>
<td>3855</td>
<td>4280</td>
</tr>
</tbody>
</table>

### 3.2. Data processing

To simulate two institutions (A & B) based on one dataset, and to allow for hyper-parameter tuning, a data processing procedure was designed to ensure the split-up datasets meet the following requirements. First, each of the four nursing wards is assigned to either institution A or B, in such a way that makes the numbers of data points in A and B as even as possible. Second, both datasets are split up into a train/validation and test set. The train/validation set is split up into 5 folds for cross-validation (CV). Third, between cross-validation folds themselves, and between the train/validation set and the testing set, no patient IDs may overlap; overlapping patient IDs could result in validating/testing on training data. This overlap sometimes occurs when a patient is moved to a different nursing ward, and the new nursing ward copies the notes taken from the previous nursing ward. Fourth, it should be possible to combine the folds between institutions to form patient-independent folds for federated and data-centralised training. Fifth, both testing sets may only include new data based on the admission timestamp, to ensure we test the final models on new data points exclusively. These requirements are visualised as a top-down procedure illustrated in Fig. 1.

### 3.3. Treatment design

In this study, four treatments are designed and compared to test all scenarios derived from the research goals mentioned in Section 1, based on Wieringa’s design cycle (Wieringa, 2014). Each treatment performs a grid search with 5-fold cross-validation (CV) to search for the best hyper-parameters for training a neural network on their respective dataset. Based on this outcome, each treatment delivers a final model by training on the data from all five folds, and is tested against a held-out testing set. These four final models are compared as part of the statistical difference-making experiment.

The difference between each treatment lies within the data it is applied on, and the training method. Two treatments are trained on data from the two simulated institutions A and B. The other two treatments, data-centralised and federated, train on data from both institutions. The data-centralised treatment trains on all data without restrictions, to show how performance would be if privacy regulations could be ignored. Therefore, it acts as a gold standard in terms of performance, as we expect the nonrestrictive training environment to deliver the best performance. The federated treatment trains a neural network on both institutional data sets through FL.

### 3.3.1. Classification model

The classification model used across treatments is a feed-forward neural network, consisting of an input layer, one hidden layer, and
output layer. The size of the input layer corresponds to the number of elements in the Doc2Vec vectors in the dataset (300). The hidden layer size is given by the variable \( h \), whose values are optimised through hyper-parameter tuning. Furthermore, the hidden layer uses the Rectified Linear Unit (ReLU) activation function, chosen for its fast computation time. The output layer has a single neuron with a sigmoid activation function for providing the classification.

The model uses the Binary Cross Entropy (BCE) with logit loss function to compute its gradients. We use mini batch gradient descent. Eq. (1) is the average loss per data point for a single batch \( n \), given input \( x \) and outcomes \( y \).

\[
l(x,y)_n = \frac{1}{T} \sum_{i=1}^{T} [p_{y_i} \cdot \log(\sigma(x_i)) + (1 - y_i) \cdot \log(1 - \sigma(x_i))] \tag{1}
\]

Batch \( n \) contains \( T \) data points. For each data point \( i \), we apply a sigmoid activation function \( \sigma \) to the input \( x_i \). To mitigate the issue of class imbalance, when the binary outcome \( y_i \) is positive, we multiply it by a weight \( p \) equal to the ratio of negative to positive samples in the dataset.

An exponential learning rate scheduler is used for training, which updates the learning rate through \( lr = lr_0 \cdot e^{\gamma d_n} \), where \( lr_0 \) is the starting learning rate, \( \gamma \) is the amount of decay, and \( n \) is the number of the current epoch. A \( \gamma \) of 0.975 is used for the experiment. This value causes the learning rate to approximately be divided by 10 at epoch 100. It is a relatively quick drop, but as there is a computational constraint in the amount of epochs we can use, we aim for models with an initial quick convergence, and use the remaining epochs for more fine-grained model updates. The maximum number of epochs is 120.

An early stopping mechanism keeps track of the validation loss of the model at each epoch. The mechanism saves a checkpoint of the model whenever the validation loss decreases since the last overall decrease in validation loss. It means that if the validation loss has not decreased in the last seven epochs, the early stopping mechanism kicks in and stops the training. It will then load the model checkpoint which has the lowest validation loss. This checkpoint model is then used for model evaluation.

### 3.3.2. Hyper-parameter tuning

Each treatment follows the hyper-parameter tuning cycle and testing procedure as shown in Fig. 2. This aims to result in hyper-parameter values optimal for training a treatment’s final model. The tuning happens through a process known as grid search in steps 1 through 4 in the Figure, where for each possible combination from a fixed set of hyper-parameters, a model is trained to reveal its respective performance. To ensure a good error estimate, the grid search is performed through 5-fold CV. Thus, for each hyper-parameter combination, five models are trained. To compute a performance measure of a combination, the ground truth labels and the predicted labels from the five models are concatenated and used as input for performance measure calculations.

The following set of hyper-parameters are used during the grid search, resulting in 36 unique combinations. These values were chosen during data exploration.

- **Hidden Layer Size:** [64, 128, 256, 512]
- **Learning rate:** [0.005, 0.001, 0.0005]
- **Weight decay:** [1e-3, 1e-4, 1e-5]

The performance measure used for fine-tuning is the F1-score, which assigns importance to correctly classifying the positive class. As the violence risk assessment dataset is strongly imbalanced and as correctly classifying patients exhibiting violence is deemed to be more important than correctly classifying patients who do not exhibit such behaviour, being able to accurately evaluate true positives among the positive predictions is key. When the combination with the best F1-score is determined in step 5, the final model can be trained. It uses the best hyper-parameters to train on the data from all 5-folds in step 6. After training, it is ready to be evaluated on the held out testing data at step 7.

### 3.4. Treatment validation

Each treatment is validated by testing its final model on the held out testing data containing only new data points based on the admission timestamp. It corresponds to the final step in the hyper-parameter tuning cycle (Fig. 2). This is done by feeding the testing data from institutions A, B, and the combination of the two sets into each treatment’s final model. From here, performance measures per treatment are computed. At each treatment uses the same set of testing datasets, and each data point is fed in an identical order to each treatment, the performance measures as well as the raw predictions can directly be compared.

For each performance measure, confidence intervals are calculated by bootstrapping the test set for 10,000 bootstraps. Bootstrapping is a method for estimating the sample distribution for a certain statistic. It is implemented by sampling the test set with replacement to produce a new test set with a different distribution of the same size, this is done 10,000 times. From here a bootstrapped mean and confidence intervals can be calculated. The 95% confidence intervals are computed for each performance measure, treatment, and test set combination. These intervals are computed through percentiles, which is known as the percentile bootstrap method, and are compared between treatments and test sets, to provide insights into relative performance.

Each treatment’s bootstrapped performance measures are compared against the federated measures by calculating the difference between the measure scores for each bootstrapped sample. Computing this difference is a method adapted from (Cumming & Finch, 2005). This too will yield a distribution with 95% confidence intervals. If the confidence interval whiskers exclude zero, then the difference is statistically significant. An important side-note for this method is that bootstrapping is ideally performed on the training set. Due to computational limitations, we performed it on the test set to see how much our specific trained models vary in their performance, when the test set is modified slightly through bootstrapping.
3.5. Federated learning implementation

The Python library PySyft is used for simulating a federated setting on a single device. We simulate two institutional devices and a central server. The training/validation algorithm implemented using PySyft is shown below in Algorithm 1. First, a central model is initialised on the server and a copy is sent to both institutions. Each institution trains the model in batches of their full local dataset. For each batch passing through the institutional models, the models are updated accordingly. After a single pass over the full dataset, the resulting institutional models are sent to the central server. Here they are aggregated by averaging the weights and biases of each layer in the neural network. The resulting averaged model is validated at each epoch to track the validation loss and performance measures. This happens by sending a copy of the averaged model back to the institutions and by feeding it the local validation sets. This results in a set of predictions and ground truth values from both institutions, which are in turn shared with the central server. The central server then computes the performance measures over the concatenation of the predicted and ground truth values from both institutions.

Algorithm 1: Federated Learning implementation  
Result: Model, performance

model = initialise_model()
for nᵢ in n_epochs do
  for inst in institutions do
    updated_model_inst = local_train_model(model, inst)
    model = average({updated_model_inst})
    (performance, loss) = validate(model)
    should_stop = early_stopping(loss) // Section 3.3.1
    if should_stop then
      break
  end for
return model, performance

Function validate(model):
  for inst in institutions do
    predictions_inst, labels_inst = local_validate_model(model, inst)
    predictions = concatenate({predictions_inst})
    labels = concatenate({labels_inst})
    performance, loss = get_performance_and_loss(predictions, labels)
  end for
return (performance, loss)

Function local_train_model(model, inst):
  updated_model = model
  for nᵢ in n_batches do
    updated_model = train(updated_model, dataset(inst))
  end for
return updated_model

Function local_validate_model(model, inst):
  predictions, labels = model(dataset(inst))
return (predictions, labels)

4. Experimental results

4.1. Data splitting

The violence risk assessment dataset contains a total of 4005 data points after removing overlapping patients from the training/validation set. The first split assigns 856 points to the testing set, and 3149 to the training/validation set. The test set is distributed to institutions A & B based on nursing wards, giving institution A 347 and institution B 509 testing data points. Distributing the training/validation set based on the same nursing ward split assigns 1410 (of which 114 positive) and 1739 (of which 184 positive) data points to institutions A and B, respectively. Table 3 displays the distribution across the cross-validation folds illustrating the class imbalance based on the training/validation set. Nursing ward A2V is appointed to institution A, and nursing wards A1, A2J, and A3 to institution B.

<table>
<thead>
<tr>
<th>Fold 1</th>
<th>Fold 2</th>
<th>Fold 3</th>
<th>Fold 4</th>
<th>Fold 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>Positive</td>
<td>257</td>
<td>277</td>
<td>265</td>
<td>263</td>
</tr>
<tr>
<td>Negative</td>
<td>26</td>
<td>22</td>
<td>18</td>
<td>17</td>
</tr>
</tbody>
</table>

Total: 1296 positive, 114 negative.

4.2. Grid search cross validation

Table 4 displays the relationship between cross-validation F1-scores and the F1-scores of applying the treatments to the held-out test set.

4.3. Performance measures

We observe in Table 5 that the federated and data-centralised models perform much alike regardless of the performance measure on our testing set. This indicates that the FL process has had no large impact on the performance of the model. When comparing the federated model to the local institutional models, we see a large gap in terms of the FL-score for all testing sets; the federated model’s FL-score was remarkably higher for each set. In addition, the federated model achieves higher scores for most performance measures regardless of the testing set compared to the local models, while achieving similar scores compared to the data-centralised model. The data-centralised ROC-AUC score on the combined test set is consistent with that found in previous work (Mosteiro et al., 2021).

4.4. Bootstrapped F1-scores comparison

For a given performance measure, the confidence intervals are calculated in two ways. Both methods rely on bootstrapping of the
Table 5
Performance measures for each treatment tested on each testing set. Values in bold are the highest among a measure given a specific test set across the four treatments. F1-score, recall, and precision use a classification threshold of 0.5.

<table>
<thead>
<tr>
<th>Test set</th>
<th>Test set Institution A</th>
<th>Test set Institution B</th>
<th>Test set Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROC-AUC</td>
<td>0.803</td>
<td>0.744</td>
<td>0.777</td>
</tr>
<tr>
<td>PR-AUC</td>
<td>0.278</td>
<td>0.274</td>
<td>0.293</td>
</tr>
<tr>
<td>F1</td>
<td>0.351</td>
<td>0.339</td>
<td>0.419</td>
</tr>
<tr>
<td>Recall</td>
<td>0.459</td>
<td>0.757</td>
<td>0.703</td>
</tr>
<tr>
<td>Precision</td>
<td>0.283</td>
<td>0.219</td>
<td>0.299</td>
</tr>
</tbody>
</table>

Fig. 3. Comparison of Confidence Intervals (95%) based on bootstrapped F1-scores. The x-axis refers to the test set. The centre points represent the mean of the bootstrapped F1-scores.

Table 6
Confusion matrices of each treatment applied to the combined test set.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Neg</th>
<th>Pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neg</td>
<td>678</td>
<td>79</td>
</tr>
<tr>
<td>Pos</td>
<td>63</td>
<td>36</td>
</tr>
</tbody>
</table>

(a) Institution A

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Neg</th>
<th>Pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neg</td>
<td>560</td>
<td>197</td>
</tr>
<tr>
<td>Pos</td>
<td>39</td>
<td>60</td>
</tr>
</tbody>
</table>

(b) Institution B

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Neg</th>
<th>Pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neg</td>
<td>615</td>
<td>142</td>
</tr>
<tr>
<td>Pos</td>
<td>41</td>
<td>58</td>
</tr>
</tbody>
</table>

(c) Federated

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Neg</th>
<th>Pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neg</td>
<td>621</td>
<td>136</td>
</tr>
<tr>
<td>Pos</td>
<td>41</td>
<td>58</td>
</tr>
</tbody>
</table>

(d) Data-centralised

4.5. Prediction comparisons

To provide a more in-depth comparison between each treatment’s predictions, the confusion matrices and contingency tables are displayed in Tables 6 and 7, respectively. We observe significant differences between the predictions of the two local models and the data-centralised and federated model. Comparing the predictions of the data-centralised and federated models alone, reveals highly similar predictions. Table 8 provides the extent to which all models agree on each test sample. We observe that all models often agree with one another on the same test samples. An interesting observation is that the models also commonly misclassify a significant number of similar test samples.

To provide more insights into the data structure and the classifications, t-SNE and PCA analyses were performed on the testing dataset of the federated model. Fig. 4 shows the result of these analyses coloured by the classification of the federated model. It reveals the difficulty of the classification task as both the t-SNE and PCA show that the true positive samples are scattered across the figures.

5. Discussion

5.1. Statistical significance

When tested on the combined testing data, the federated model achieved an F1-score of 0.388 and the data-centralised achieved 0.397. This is in line with our expectations that both models would perform on par with one another. We also report on Table 5 the areas under
the Receiver Operating Characteristic (ROC-AUC) and precision-recall curve (PR-AUC). These measures indicate performance without choosing a classification threshold. On the combined test set, both models had the same ROC-AUC (0.765) and PR-AUC (0.288). Furthermore, there was not enough evidence to reject the null hypothesis that the data-centralised and federated model were similar, with a significance threshold of 0.05. While we cannot conclusively state that the federated and data-centralised model perform on par, they did so on our test dataset.

The F1-score of institution A, as observed on its own test set (0.351), was much lower than the federated model (0.419). Even though the observed differences from the experiment were large, the bootstrapped confidence intervals could not prove a statistically significant difference, given a significance threshold of 0.05. However, we can state that the federated model was better on our testing data. Analysing the F1-scores we observe that the federated model outperformed the local model, but we cannot state with statistical significance that federated model is better.

5.2. Model differences

The confusion matrices of all models as displayed in Table 6 reveal the differences in correct and incorrect predictions between each model. Institution A delivered the most accurate model giving the lowest number of incorrect predictions (79 + 63 = 142). However, the model also yielded the highest number of false negatives (63) and lowest number of true positives (36), which is why it had the lowest F1-score out of all models. The data-centralised model and federated model give an identical number of false negatives (41) and true positives (58), while differing only by six samples for true negatives and false positives. The model of institution B yielded the highest number of true positives (60) at the expense of the highest number of false positives (197). The differences between institutions A and B remind us that the F1-score is not the end of the story. It will also depend on whether a practitioner favours a high number of false negatives over a high number of false positives, or vice versa.

Contingency tables reveal the relative performance and agreement of each model compared to the federated model on the combined data. Tables 7a & 7b show that there is a significant disagreement between the two local models compared to the federated model. Table 7c shows that the highest level of agreement is between the data-centralised and federated model, disagreeing only on 36 (21 + 15) data points.

To see the extent to which all models agree with one another, Table 8 displays the number of commonly correctly classified and the commonly misclassified test samples. Common, within this context, is defined as having all models agreeing upon a single outcome. The confusion matrices of all models as displayed in Table 6 reveal the differences in correct and incorrect predictions between each model. Institution A delivered the most accurate model giving the lowest number of incorrect predictions (79 + 63 = 142). However, the model also yielded the highest number of false negatives (63) and lowest number of true positives (36), which is why it had the lowest F1-score out of all models. The data-centralised model and federated model give an identical number of false negatives (41) and true positives (58), while differing only by six samples for true negatives and false positives. The model of institution B yielded the highest number of true positives (60) at the expense of the highest number of false positives (197). The differences between institutions A and B remind us that the F1-score is not the end of the story. It will also depend on whether a practitioner favours a high number of false negatives over a high number of false positives, or vice versa.

5.3. Limitations

The first limitation of this study is concerning the Doc2Vec model. The model was trained on all clinical notes present in the violence risk assessment dataset. There are two issues with this approach. First, it would have been more realistic to train this Doc2Vec model using FL; currently it was trained with a data-centralised approach to create the best Doc2Vec model we could with the limited size of the data set. For a real life scenario, adding federated Doc2Vec training to the pipeline...
is a prerequisite, for the same reasons data-centralised training is not allowed cross-institutionally. However, Doc2Vec is not compatible with PySyft at the moment. The second issue is that the Doc2Vec model has also been trained on clinical notes occurring in the testing set. When making real life predictions, it is unusual to retrain the Doc2Vec model to include the testing clinical notes and retrain the classification model.

Another limitation is that other privacy preserving technologies were not investigated in this study. Models trained through FL can be attacked like other machine learning models, and an attacker might be able to infer details about the training data of the model. FL can be combined with other privacy preserving techniques such as differential privacy (Dwork, 2006), homomorphic encryption (Gentry, 2009), and secure multiparty computation (Yao, 1982). These techniques might alleviate additional privacy concerns, but could also negatively impact model performance. To guarantee a high level of privacy to admitted patients, combining FL with these techniques might be required.

Lastly, we observed a large variance in the performance measures and believe this can be attributed to the small test set with a low number of positive samples. Because of this, there is a high probability that a bootstrapped sample contains a skewed class distribution, which has a high impact on the variance of F1-scores.

6. Conclusions

Violence Risk Assessment (VRA), like many other clinical tasks, can be tackled with Machine Learning methods. In the psychiatry domain, NLP methods are particularly interesting thanks to the abundance of clinical notes containing valuable information. NLP models benefit enormously from bigger and more diverse datasets, as can be acquired by working across multiple institutions. Since data sharing among institutions is not possible, we have developed a Federated Learning (FL) pipeline for training an algorithm for VRA. We found no performance loss from using FL, as opposed to a data-centralised approach. Also, FL seems to improve the performance of locally trained models tested on a different dataset. To the best of our knowledge, this is the first application of FL and NLP on clinical texts.

The results suggest that there are benefits to using federated models and this should be investigated further with cross-institutional datasets. Not only would this provide insights into real-life deployment, it would also lead to more data points for training and testing and could help to decrease performance measure variance.

In future work, we plan to train document embeddings in a federated environment. Furthermore, we will investigate how FL can help solve other clinical tasks, such as text de-identification. Finally, we plan to investigate the possibility of adding other additional privacy-preserving technologies, such as differential privacy.

CRediT authorship contribution statement

Thomas Borger: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization.

Pablo Mosteiro: Methodology, Software, Investigation, Data curation, Supervision, Project administration, Writing – review & editing.

Heysem Kaya: Methodology, Supervision, Writing – review & editing.

Emil Rijcken: Software, Investigation, Writing – review & editing.

Albert Ali Salah: Methodology, Supervision, Writing – review & editing.

Floortje Scheepers: Investigation, Resources, Funding acquisition.

Marco Spruit: Conceptualization, Supervision, Project administration, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Utrecht University, The Netherlands, the PsyData team from UMC Utrecht, The Netherlands, and KPMG, The Netherlands n.v. Useful discussions were held with Sebastian Mildner and Laurens Breij.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2022.116720. For each treatment's final model, the receiver operator characteristic (ROC) curve has been computed. The supplementary data provides these curves for each test set.

References


