LEVERAGING RELATED DATASETS TO IMPROVE MODEL PERFORMANCE ON AN UNDERREPRESENTED TARGET POPULATION

Master Thesis

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday 6 th July, 2023 at 13:00 o'clock.

by

Maxmillan RIES

Project Duration:	11, 2022 - 07, 2023
Supervisors:	Dr. D.M.J. Tax
Thesis Committee:	Dr. M.J.T. Reinders
	Dr. O.E. Scharenborg

PREFACE

This report represents the culmination of my Master's thesis conducted at Delft University of Technology. I am immensely grateful to Dr. David M.J. Tax for his outstanding guidance and supervision throughout this entire project. I also extend my heartfelt appreciation to Florent Fayette, who served as my secondary daily supervisor, offering constant feedback and unwavering support.

Moreover, I want to express my deep gratitude to my family and friends for their continuous encouragement and support throughout this challenging journey. A special mention goes to my friend Nafie, who took the time to review and provide valuable feedback throughout my thesis.

Completing this thesis has been an immensely rewarding and exhilarating experience. I wish to extend my sincere thanks to everyone who accompanied me on this remarkable journey.

Maxmillan Ries
Delft, July 2023

Leveraging Related Datasets to Improve Model Performance on an Underrepresented Target Population

Maxmillan Ries, David M.J. Tax

Abstract

Training deep learning models for time-series prediction of a target population often requires a substantial amount of training data, which may not be readily available. This work addresses the challenge of leveraging multiple related sources of time series data in the same feature space to improve the prediction performance of a deep learning model for a target population. Specifically, we focus on a scenario where the target dataset, representing the desired target population, is underrepresented, while the source datasets consist of mismatched populations that are sufficiently representative for training a deep learning model. In this study, we explore state-of-the-art techniques, including transfer learning, ensemble learning, and domain adaptation to leverage source datasets towards a target population using real-world medical data. Additionally, we investigate the use of model performance-derived baselines as a heuristic to quantify the magnitude of the distribution mismatch between a source(s) and a target. Our results demonstrate that a set of well-defined baselines can effectively quantify the distribution mismatch and provide insights into the choice of leveraging technique for a given mismatch scenario. Furthermore, our results show that all state-of-the-art techniques can be employed to leverage related source datasets towards the target, though the performance of these techniques varies depending on the characteristics of the distribution mismatch. Eventually, we discuss the applicability of this research to new scenarios, along with avenues for future research.

1 Introduction

Modern machine learning methods, particularly deep learning methods, often require a large volume of training data due to the difficulty of the model's task, the high dimensionality of the input data, and/or the complexity of the model [4]. Furthermore, large datasets for a specific task, such as predictive maintenance or medical diagnostics may not be readily available [53], or it may be desirable to tailor model performance towards a subpopulation with a low sample size. In the medical field for instance, it is important for models used in each hospital to be tailored to the hospital's patient population [28, 50, 52]. Very similar cases can also be found in the industrial field [43, 56]. As training accurate and reliable models can be difficult when specific data is required, it can be useful to employ additional data from related sources.

The process of leveraging data from related sources requires tackling two core challenges, *feature mismatch*, and *distribution mismatch*. A feature mismatch occurs when two or more datasets contain different features. This challenge can notably be observed in the medical field, where hospitals often uphold different practices towards the same goal [52], or in the industrial field, where each company creates its features using different customized sensors [43, 56]. A distribution mismatch occurs when two or more sources of data contain differences between populations [53]. In the medical field, such a mismatch can be seen by comparing two hospitals at different geographical locations. In this scenario, there exist multiple distribution mismatches, where the medical equipment used is different, the ethnicity distribution across the populations is different, and the hospital standard of treatment and patient standard of living may be different.

In any field, to utilize data from several sources to construct subpopulation-specific models, possible distribution mismatches must be addressed and resolved [53]. When ignored, the trained model may be biased towards the most represented populations, resulting in poor model performance towards underrepresented populations. This paper investigates the following question: *How can related sources of time-series data be leveraged to improve the performance of a Long Short-Term Memory deep learning model (LSTM) towards a target population with underrepresentative data?*

The scope of this work is to tackle the challenge of leveraging multiple sources of time-series data in the same feature space to improve the predictive performance of an

LSTM deep learning model towards a target population. For simplicity, we consider each population of interest to be represented by a single dataset, with all populations differing by a distribution mismatch. We investigate the specific scenario where the *target dataset* (containing only the target population) is underrepresentative, and the *source datasets* (each containing a mismatched population to the target) are sufficiently representative to train a deep learning model.
An alternative approach using Generative Adversarial Networks (GANs) has shown good results [53], but GANs are notorious to train well and need large training sets. Although the source dataset may be sufficiently large, the target dataset certainly is not, therefore the use of GANs is not considered further.

2 Related Works

Transfer Learning

In machine learning, transfer learning methods focus on transferring knowledge across domains as a means of resolving the lack of abundant training instances [57]. The technique is inspired by the capability of the human mind to transfer knowledge across domains [57]. Specifically, transfer learning aims to leverage knowledge from related source domains to improve model performance on a target domain [26]. This technique is frequently employed in cases when insufficient data for a population is available to train a machine learning model, but sufficient data is available from similar populations to be used as an additional source of information [26]. Within the image-processing domain, it has become increasingly common to pre-train deep Convolutional Neural Networks (CNN) on the ImageNet dataset [42], to learn good general-purpose features [21]. The use of transfer learning on such a dataset has over time become a *de facto* standard for solving many computer vision problems [21], with impressive results having been presented in image classification [37], action recognition [54], and image segmentation [25]. Similarly, transfer learning was found to apply to time-series forecasting via the use of LSTM-CNNs or deep CNNs [11]. In such scenarios, transfer learning was found to improve regression and classification model performance across most deep learning architectures [11, 17, 45, 49].

However, while transfer learning has the potential to help machine learning models pre-learn robust features using larger related domain datasets, it can also occur that the target learner is negatively affected by the transferred knowledge, known as a *negative transfer* [57]. There exist many possible reasons why a negative transfer can occur, such as the relevance between the source and target domain or the relevance of the transferable knowledge [26, 57]. Several works in the field of image and time-series prediction show that such a phenomenon is common when the technique is poorly applied in unfavourable circumstances [11, 17, 45, 49].

Reweighting

Reweighting is a bias correction technique, which can be used to help leverage data from several source domains, by addressing different imbalances in the data, such as disparities between population sizes. When aggregating several related datasets, reweighting can be used to assign importance weights to each population in the training data, ensuring that the minority populations have a sufficient impact on the training of the model [5]. An example of this can be found in the medical domain, where reweighting was found to mitigate disparities introduced by data underrepresentation [2].
Furthermore, reweighting can also be applied to the combination of several model outcomes, as a form of Ensemble Learning [1]. This form of ensemble learning can be used for leveraging related datasets, by aggregating the outcome of models trained towards different populations, rather than aggregating the data of each dataset. Several works in this field have shown that weighting the predictions of multiple models can significantly decrease the forecasting error over time-series data, increasing the reliability and usability of models within critical fields, such as medicine [1, 10, 38].

Domain Adaptation

Domain adaptation is a sub-field within machine learning that aims to address the challenge of training a model on a source domain and generalizing it to a target domain differing by a distribution mismatch.
For instance, Jin et al. proposed an attention-sharing adversarial domain adaptation forecaster for both synthetic and real-world time-series data, which demonstrated superior performance over existing state-of-the-art baselines [24]. Wilson et al. also explored domain adaptation techniques for time-series data and proposed a novel convolutional deep domain adaptation model. Their method achieved significant improvements in accuracy and training times on real-world sensor data benchmarks [51].
In the healthcare industry, domain adaptation is particularly important for improving the generalization of machine learning models across different patient populations that exhibit distribution mismatches. McDermott et al. found that Multi-Task Learning (MTL) significantly improved single-task performance on time-series forecasting tasks [33]. Their study also demonstrated the benefits of MTL applied to both traditional and few-shot learning scenarios [33].

Bayesian Hierarchical Models and Hierarchical Markov Models

When combining multiple datasets, it is frequently the case that the datasets are not completely independent [29]. Hierarchical models are frequently applied when the generating mechanism behind the data is thought of as having a hierarchical structure [13]. For example, Bayesian Hierarchical Models (BHMs) have been shown to adjust for inter-study differences, incorporating multi-scaled spatial and temporal data, and provide less biased estimations than traditional methods [13, 32].
Hidden Markov Models (HMMs), are a tool for representing probability distributions over sequences of observations [18, 47]. Hierarchical HMMs (HHMMs), are an extended form of HMMs that include a hierarchy within the hidden states [7]. Both HMMs and HHMMs have shown promising results when applied in fields such as natural language processing and bio-molecular imaging [27, 44], notably when applied to temporal data [9, 16].

HMMs, though applicable to our data, are difficult to use with other leveraging techniques, such as Transfer Learning. Furthermore, incorporating HMMs would require architectural changes which prevent a direct comparison of methods. Hierarchical models show promise in leveraging multiple related datasets. Unfortunately, due to time constraints, they will not be further investigated.

Generative Adversarial Networks

Generative Adversarial Networks are a group of deep neural networks that have gained much popularity over the last decade with the affordability of computation and accessibility of data. Recent works in leveraging multiple datasets have made use of this unique deep neural architecture, as a tool to solve multiple distribution and feature mismatches simultaneously [35, 43, 53, 56]. While GANs are a powerful tool, they are not addressed in this paper, as they require large volumes of data and training time, and the use of a unique adversarial architecture.

3 Methodology

This section describes the scope of the problem, the data, the relevant definitions and assumptions made, as well as the methods employed for experimentation.

Scope, Data, Definitions, and Assumptions

In this paper, we consider the problem of leveraging a set of source time-series datasets $M = \{D_1, D_2, ..., D_N\}$ to improve the classification performance of a supervised model towards a target population, represented by the dataset D_{target}.

Specifically, we assume that both the source and target datasets have the same feature space, and are of the form $D = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_J, y_J)\}$, where j is the size of dataset D (differs per dataset), $\mathbf{X}_J$ is a sequence of T feature vectors,

$$\mathbf{X}_J = \underbrace{\left.\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,T} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,T} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{k,1} & x_{k,2} & x_{k,3} & \cdots & x_{k,T} \end{bmatrix}\right\} \text{features}}_{\text{time}}$$

and y_J a corresponding outcome label.

Additionally, we assume that the target dataset D_{target} is underrepresentative of the target population, and that the data in our set of sources M differ by a known distribution mismatch, but are related to the target population.
To leverage data from our set of sources M, we employ a variety of state-of-the-art techniques, detailed below.

Concatenation

Basic Approach: A simple method of merging multiple datasets consists of concatenating them. As our setting assumes that the source and target datasets have the same feature space, we can simply concatenate the different datasets into a single dataset $D^{\text{concat}} = \{D_1 \cup D_2 \cup \ldots D_N \cup D_{\text{target}}\}$, and use this mixture as the training data for the model, resulting in the following loss function:

$$L(y, \hat{y}) = \sum_{i=1}^{N+1} L(y_{D_i}, \hat{y}_{D_i})$$

where y is the true label, $\hat{y}$ is the predicted outcome, and the sum of i is over N source datasets and D_{target}.

Weighted Concatenation: In the basic approach, it is assumed that every sample holds equal importance during training. In reality, due to the size differences between source and target datasets, any trained model will be biased towards the larger sources. An extension to the basic approach, which aims to address this imbalance bias between datasets, is to use dataset weights $\{w_i; i = 1 \ldots N+1\}$. There exist many ways to set the weights per dataset, and in this paper, we chose to employ an algebraic solution, where the weight of each dataset is set inversely proportional to its size, artificially equating the importance of all datasets during training. The weighing term can be seen in the loss function below:

$$L(y, \hat{y}) = \sum_{i=1}^{N+1} w_i \, L(y_{D_i}, \hat{y}_{D_i})$$

$$w_i \mid D_i \mid = \frac{1}{N+1}; \forall i$$

where y is the true label, $\hat{y}$ is the predicted outcome, the sum of i is over N source datasets and D_{target}, and $\mid D_i \mid$ is the cardinality of D_i. As the source and target datasets differ by a distribution mismatch, we additionally experiment with biasing the model towards the target population. Though these weighing solutions offer a consistent and less time-consuming weight selection, it is also possible to employ other optimization strategies, such as trainable weights or grid-search methods to find a balance of weights.

Transfer Learning

Transfer Learning can also be used to leverage knowledge from source domains to improve model performance towards a target domain [57]. Typically, transfer learning is applied by first training a model on a source dataset(s) (known as pre-training) before fine-tuning it to the target dataset (e.g., training with a lower learning weight). The rationale is that using the source datasets initially allows the model to extract relevant patterns from the input data that may apply to the target dataset. Moreover, in pre-training the network with a set of sources similar to the target dataset, fewer samples from the target domain may be necessary [57].

Some factors must be considered when using transfer learning, such as the possibility of a negative transfer and the numerous hyper-parameters requiring optimization. We explore a simple, but representative use of transfer learning by fine-tuning the model with a lower learning rate than the pre-training phase. While this specific solution is not optimized towards the dataset-specific task, it serves as an example of how transfer learning can be used to leverage source datasets.

Ensemble Learning

Ensemble learning involves combining multiple models trained on different datasets to make a prediction [10]. This can be achieved using techniques such as boosting, bagging or stacking. This paper explores a form of boosting, where a model is trained per dataset, and the independent model outcomes are averaged into a final prediction. Given the set of source datasets M, the target dataset D_{target}, and a function $\hat{y}_j = f_i(\mathbf{X}_j)$ where a model trained on D_i predicts sample $\mathbf{X}_j$, the final ensemble prediction $\hat{y}_j$ is given by:

$$\hat{y}_j = \sum_{i=1}^{N+1} \beta_i f_i(\mathbf{X}_j)$$

$$\sum_{i=1}^{N+1} \beta_i = 1$$

where β_i is the per-model prediction weight and can be selected by hyper-parameter estimation, or manual tuning.
The use of a weighted combination of models ensures that the performance on D_{target} is maximized, and the models with a less robust prediction are given a lower weight. We investigate this methodology using both a grid search optimization over the training set and a heuristic-based solution derived from the baselines representing the distribution mismatch.

Domain Adaptation

In the case where the source and target datasets are collected from different domains, domain adaptation techniques can be used to merge the datasets.

Domain Separation: One method consists of adjusting the feature representation of the data in such a way as to learn a hidden representation where the source and target populations are distinguishable by the model. This can be achieved with a method such as Multi-Task Learning over the concatenated datasets, where there exist two target labels, the task label $Y = \{y_1, y_2, \ldots y_j\}$, and a dataset label $Z = \{z_1, z_2, \ldots z_j\}$, which represents the dataset each sample is drawn from. Using these two labels, a composite loss $L_{y,z,\hat{y},\hat{z}}$ is used to train the model:

$$L_{y,z,\hat{y},\hat{z}} = \alpha\, L_{\text{pred}}(y,\hat{y}) + (1-\alpha)\, L_{\text{data}}(z,\hat{z})$$

$$0 \leq \alpha \leq 1$$

where α is a weight factor that influences the importance of each respective task, L_{pred} is the loss function of the predictive task, and L_{data} is the loss function of the dataset classification. In the ideal situation, both tasks complement each other, and help the model better estimate the task-specific prediction by learning how to separate the source(s) from the target. In this paper, we use a grid-search optimization to find an ideal α parameter using the training set.

4 Experiments and Results

Using a real-world dataset as an example, we investigate how each method defined in Section 3 can be used to leverage source datasets to improve model performance on a target dataset. This section outlines the data used and the experiments conducted.

Dataset

The experiments conducted in the following sections are all performed on a scenario created using a real-world dataset. We used the 2019 PhysioNet/Computing in Cardiology Challenge dataset [19, 40, 41], which contains patient data from two distinct hospitals in the United States (the Beth Israel Deaconess Boston Medical Center and the Emory University Hospital Atlanta). The 40,336 total patient files each contain hourly time-series records of a patient's admission to the Intensive Care Unit (ICU) and consist of 41 features; 8 vital

signs (such as Heart Rate, Temperature, Pulse Oximetry, etc.), 26 laboratory values (such as Measure of Excess Bicarbonate, Fraction of Inspired Oxygen, etc) and 6 demographic variables (such as Age, Gender, Length-of-stay, etc). The final feature is a binary value, which indicates whether or not a patient has met the gold standard of sepsis, according to the modern Sepsis-3 guidelines [39], and is used as the target label described in Section 3[1].

Model

The focus of this paper is primarily to showcase the potential improvement in the predictive performance of an insufficiently sampled target dataset using a source dataset(s), rather than optimally solving the prediction of Sepsis. The methodology employed in this study is deliberately kept similar to the approach used by Congxing Zhu [55], to ensure reasonable performance from the source data. The specifics of the data processing can be found in Appendix A, with the final model described in Figure 1 below.

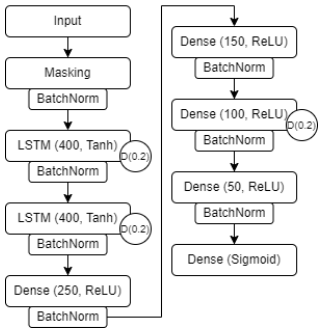


Figure 1: *Diagram displaying the architecture of the model. The boxes marked D have a dropout layer for regularization.*

Each experiment will be consistently trained using Binary Cross-Entropy for Sepsis prediction ($L_{\text{pred}}(y, \hat{y})$) and Categorical Cross-Entropy for domain adaptation ($L_{\text{data}}(z, \hat{z})$) over 10-fold cross-validation on patients. As the goal of the paper is not to optimally solve the PhysioNet Challenge, alternative loss functions will not be investigated.

Single Distribution Mismatch

In Section 3, several aggregation techniques were described to combine multiple source datasets towards a target population. This experiment investigates how each method leverages a single source dataset. To generate the source and target datasets from the PhysioNet database, we identified three distinct real-world distribution mismatches, described below:

- *Gender:* Eachempati et al. [14] found that gender is an independent predictor of Sepsis. As the PhysioNet datasets contain anonymized gender labels, they can be used to create two distinct populations. The gender labelled 0 (23931 patients) will be used as the source, and the gender labelled 1 as the target population (18675 patients)[2].
- *Hospital:* There exist several distribution mismatches between hospitals in different geographical locations, such as racial distribution, quality of healthcare and medical equipment [53]. As the two hospitals used in the PhysioNet challenge lie in different states of the USA (Massachusetts and Georgia), hospital location is a variable which can create two mismatched populations. Hospital *B* will be used as the source dataset (20815 patients), and Hospital *A* will be used as the target set (21790 patients)[2].
- *Age:* Martin, Greg S et al. [31] found a correlation between patient mortality to Sepsis and age. Although age is provided for each patient of the PhysioNet dataset, it is not a binary label. Consequently, to create two population, we will use patients younger than 30 and older than 45 (2050 vs 35354 patients). The age distribution across patients can be found in Appendix B. Patients between the age of 30 and 45 were removed from the data to create a gap between the young and older populations, to increase the effect of the distribution mismatch.

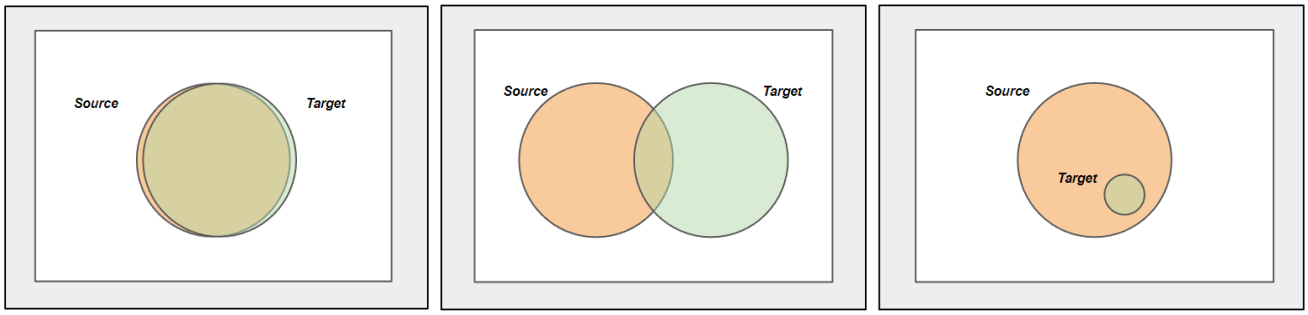
This paper aims to address the specific scenario where the target data is underrepresentative and insufficient. As the gender and hospital features split the data into two equally-sized datasets, the target training data was randomly undersampled to 5% of the fold data. In the case of age, the age thresholds were chosen such that the target population does not require further undersampling. We will test each of the methods described in Table 2 to leverage data from the larger source to the smaller target dataset. The performance of each method will be compared to a set of baselines, described in Table 1.

		Test	
		Source	**Target**
Train	**Source**	Indicates the representativeness of the source dataset	Provides a heuristic about the distribution mismatch
	Target	Secondary heuristic about the distribution mismatch, and a secondary indication of the under-representativeness of the target dataset	Provides an indication of the under-representativeness of the target dataset

Table 1: *Summary of the baselines, the training and testing combinations, along with the matching observations expected.*

[1]While there exists an additional dataset, which was used by the PhysioNet Challenge to evaluate participants' models, it was not made publicly available. Consequently, we have developed and evaluated our algorithms exclusively using the publicly available data (from Hospital A and B, according to the challenge labels).

[2]The source and target selection was made at random. Switching the source and target datasets is not expected to cause large changes in the observed outcome.



(a) Gender Distribution Mismatch **(b) Hospital Distribution Mismatch** **(c) Age Distribution Mismatch**

Figure 2: *Overview of the three selected distribution mismatches in the PhysioNet Sepsis dataset.* ***(a)*** *shows the gender distribution mismatch, where the source and target distributions largely overlap.* ***(b)*** *shows the hospital distribution mismatch, where a small overlap is present between source and target distributions.* ***(c)*** *shows the age distribution mismatch, where the target distribution is a subset of the source distribution. These distribution mismatches were identified using the baseline results of Table 3.*

S-Weighted Concatenation	Basic concatenation with no weights introduced. The model is biased towards the larger source dataset.
Equal-Weighted Concatenation	Weighted concatenation, with the source and target datasets equally contributing to model training, $w_{source} = \frac{1}{\lvert D_{source}\rvert}$ and $w_{target} = \frac{1}{\lvert D_{target}\rvert}$.
T-Weighted Concatenation	Weighted concatenation, with the source and target datasets being weighed inversely proportional to the square of their size, $w_{source} = \frac{1}{\lvert D_{source}\rvert^2}$ and $w_{target} = \frac{1}{\lvert D_{target}\rvert^2}$. This weighting biases the model towards the target population during training.
Transfer Learning	Pre-training on the source dataset (learning rate = 0.001), before fine-tuning on the target dataset (learning rate = 10^{-5}).
Boosting (Grid)	Ensemble Learning using a source-trained and target-trained model, with the weight β_i found using a grid-search over the target training set.
Boosting (Heuristic)	Ensemble Learning using a source-trained and target-trained model, with the weight β_i set using the baselines as a heuristic, $\beta_{\text{source}} = \frac{B_{S-T}}{B_{S-T}+B_{T-T}}$ and $\beta_{\text{target}} = \frac{B_{T-S}}{B_{S-T}+B_{T-T}}$
S-Weighted Domain Adaptation (DA)	Multi-task learning over the source-weighted concatenation of source and targets to predict both Sepsis label and dataset of origin per sample, with α set using grid search over the training set.
Equal-Weighted Domain Adaptation (DA)	Multi-task learning using the equal-weighted concatenation of source and target to predict both Sepsis label and dataset of origin per sample, with α set using grid search over the training set.

Table 2: *Summary of methods for leveraging single-source data. On the left-hand-side, is the name of the experiment conducted. On the right-hand-side, is the description of the experiment and all relevant parameter choices.*

	AUROC (in %)		
	Gender	**Hospital**	**Age**
Source to Source	79.3 ± 1.4	80.4 ± 2.4	80.0 ± 1.8
Source to Target	78.1 ± 3.3	68.3 ± 2.6	77.4 ± 6.1
Target to Target	68.0 ± 4.8	65.6 ± 2.4	72.8 ± 8.0
Target to Source	65.2 ± 3.9	54.1 ± 10.8	63.8 ± 4.9

Table 3: *Baseline mean AUROC results for Sepsis prediction with a single distribution mismatch across the 10-fold cross-validation over patients. The value following the $\pm$ sign is the standard deviation across the folds.*

Results

Data Representativeness

The source-to-source and target-to-target baselines assess the representativeness of the source and target datasets on the respective populations. The source-to-source performance across all distribution mismatches is reliable, and comparable to the AUROC performance obtained by Congxing Zhu [55]. The source-trained model performance across each validation fold is consistent for each distribution mismatch, indicated by the low standard deviations. The target-to-target results show that, while not critically insufficient, the target datasets are less representative than the respective sources, with each target-to-target baseline achieving a 7.2-14.8% lower mean AUROC performance. Moreover, the target-to-target baselines have a higher standard deviation than the source-to-source baselines, indicating large fluctuations across the validation folds. These fluctuations further highlight the underrepresentativeness of the target training data.

Distribution Mismatch

The source-to-target and target-to-source baselines both provide a quantitative assessment of the distribution mismatch between source and target populations. However, the target-to-source baseline is frequently less reliable as an indicator of mismatch, due to the underrepresentativeness of

the target training data.
The gender source-trained and target-trained models demonstrate insignificant changes in model performance when applied to the mismatched population ($p = 0.3038$ and $p = 0.1694$, respectively). This indicates, contrary to the findings of Eachempati et al. [14], that the gender mismatch is minimal. Similarly, the baseline results show a comparable mismatch pattern for age. The source-trained model performance insignificantly decreases by 2.6% when tested on the target data, indicating a small distribution mismatch ($p = 0.2124$). However, the target-trained model performance significantly decreases when tested on the source data ($p = 0.0077$). Taken together, these results indicate that the target distribution is likely a subset of the source distribution, and the smaller target dataset is insufficient to capture the whole population.
Lastly, the hospital baselines indicate that the patients from both hospitals differ by a larger distribution mismatch. Both the source-trained and target-trained models significantly decrease in performance when tested on the mismatched population. Notably, the target-to-source dataset indicates that the target-trained model is unable to generalize to the source population, with a mean AUROC of 54.1%.

The baseline results offer insights into the reliability of the source data, the representativeness of the target data, and provide a quantitative measure of the extent of the distribution mismatch. When examining gender and age as mismatch variables, the results show that the observed mismatches do not have a significant impact on the performance of the model. This suggests that there are no notable biases in the predictions from source to target data. In contrast, a distinct distribution mismatch can be observed between source and target populations when considering the hospital variable. Both the source-trained and target-trained models experience a decline in performance when tested on populations mismatched from their training data.
The remainder of this section delves into a detailed analysis of the outcomes obtained by employing state-of-the-art techniques for leveraging the source dataset towards the target, the result of which can be found in Table 4.

Experimental Results

Collectively, the experimental results show that all state-of-the-art techniques demonstrated comparable or superior performance to the target-to-target baseline, with the best performing experiments providing an increase of 11.2%, 11.8% and 8.3% for Gender, Hospital and Age respectively. For the gender distribution mismatch, the source-weighted concatenation, boosting and domain adaptation techniques provided the highest increase in performance, with grid-search boosting doing so with a much lower standard deviation. For the hospital mismatch, transfer learning outperformed all other methods, with source-weighted concatenation providing a similar, albeit slightly lower improvement. For age, source-weighted concatenation outperformed all other methods, with equal-weighted concatenation, grid-search boosting and domain adaptation returning similar improvements. Similarly to the gender mismatch, boosting also returned the lowest standard deviation across the folds.

	AUROC (in %)		
	Gender	**Hospital**	**Age**
S-weighted Concat.	79.1 ± 2.6	76.0 ± 2.3	81.1 ± 5.6
Equal-weight Concat.	76.6 ± 2.4	73.5 ± 2.7	80.0 ± 6.3
T-weighted Concat.	67.4 ± 3.5	70.0 ± 3.1	75.2 ± 4.4
Transfer Learning	78.6 ± 2.9	77.4 ± 1.9	76.0 ± 7.1
Boosting (Grid)	79.0 ± 1.3	69.9 ± 2.5	80.3 ± 2.8
Boosting (Heuristic)	77.6 ± 2.3	72.7 ± 2.3	77.3 ± 2.5
S-weighted DA.	79.2 ± 2.4	74.9 ± 2.5	80.5 ± 5.5
Equal-weighted DA.	77.8 ± 2.1	74.7 ± 2.7	79.8 ± 6.6

Table 4: *Mean AUROC results of leveraging techniques for a single distribution mismatch across the 10-fold cross-validation over patients. The value following the $\pm$ sign is the standard deviation across the folds.*

Concatenation

The three concatenation experiments, in conjunction with the source-to-target and target-to-source baselines, aim to investigate the influence of both source and target datasets on the model performance. Overall, the source-weighted concatenation approach demonstrated superior performance compared to other weighted concatenation methods and baselines across all distribution mismatches. While the combination of source and target datasets improved model performance on the target population, increasing the importance of the target dataset during training resulted in negative returns. This outcome can be attributed to the underrepresentativeness of the target population, which introduces bias and noise during training. In the case of gender and hospital mismatches, as the importance of the target dataset increased during training, the standard deviation also increased. A similar trend was observed for the age distribution mismatch, although the target-weighted concatenation yielded the lowest standard deviation. This behavior is unusual and no explanation for it was found, notably as the target-to-target baseline has a very high standard deviation.
For the three distribution mismatches, the optimal weights between the source and the target lies close to those of the source-weighted concatenation approach. The optimal balance is anticipated to vary from dataset to dataset, particularly depending on factors such as the sample size or noisiness of the source and target datasets.

Transfer Learning

The results of Table 4 demonstrate that transfer learning did not yield significant improvements over the source-to-target

baseline for gender and age ($p = 0.7231$ and $p = 0.6419$, respectively). However, in the case of the hospital mismatch, transfer learning outperformed other techniques, with an increase of 9.1% in model performance over the source-to-target baseline. The smaller mismatches observed with gender and age suggest that the initial pre-training of the model on the source dataset, followed by fine-tuning with the target data, may have led to a loss of representative information. Consequently, the performance gain achieved to transfer learning was comparatively lower than concatenation-based methods. However, in scenarios involving a larger distribution mismatch such as hospital, transfer learning proved to be effective in leveraging information from the source dataset and improving the performance towards the target population.

Boosting

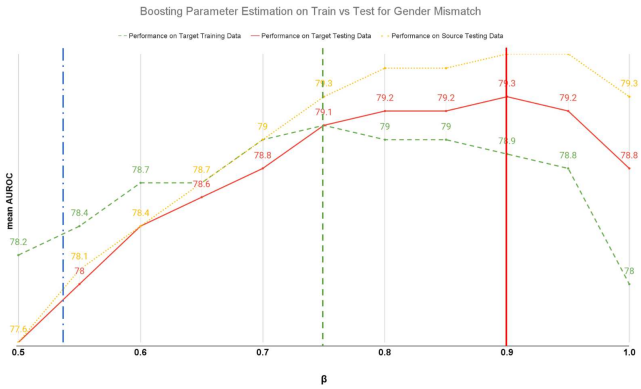


Figure 3: *Figure showing the boosting performance with different* β *parameter values for the gender distribution mismatch. The green (short dash) vertical line shows the optimal* β *found during training. The red (solid) vertical line shows the optimal* β *for the test set, and the blue (dash/dot) vertical line shows the baseline derived* β *value.* β = *1 indicates only source model usage.*

Figure 3 and 4 show the boosting mean AUROC for different β values over the target training, testing and source testing set. For all distribution mismatches, the predicted parameter values using the training set differ from the optimal value found on the test set. With the gender mismatch, the boosting performance minimally decreases, with a similar difference being observable for the age variable (see Appendix C). With the hospital mismatch, the performance difference between train and test-optimized models is 4.8%. Moreover, with all three mismatches, the absence of the target-trained model reduces overall performance across both source and target test data. This performance difference is likely due to the design behind the boosting method, where the source data is not adapted, but instead used as-is, thereby presenting a strong bias towards the incorrect population.

The baseline-derived weights aim to find an optimum without the need to re-train the model repeatedly. With the

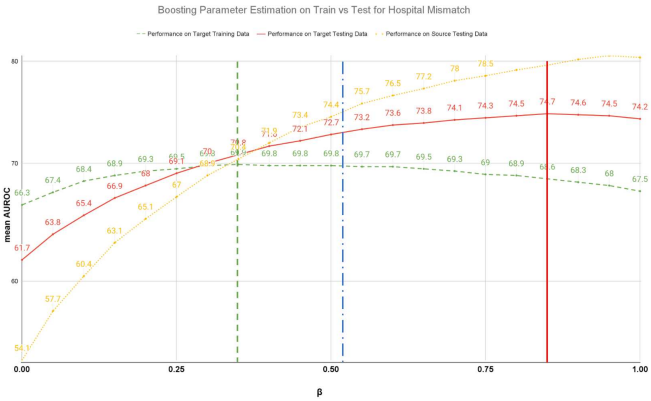


Figure 4: *Figure showing the boosting performance with different* β *parameter values for the hospital distribution mismatch. The green (short dash) vertical line shows the optimal* β *found during training. The red (solid) vertical line shows the optimal* β *for the test set, and the blue (dash/dot) vertical line shows the baseline derived* β *value.* β = *1 indicates only source model usage.*

gender and age distribution mismatches, as the differences between the source-to-target and target-to-target baselines are low, the optimal weight calculated hovers around 0.5 between source and target models. In this situation, the baseline-derived weight offers no benefit over a grid search optimization. However, with the hospital distribution mismatch, due to the underrepresentative and mismatched training data, a grid search is unable to find a set of performing weights. The heuristic, by weighting the source dataset more than the target, due to the baseline performance differences towards the target set, can better approximate the test-set optimum. It is hence expected that in cases where the target population is even less representative, the heuristic will perform better than most data-driven optimization strategies.

Domain Adaptation

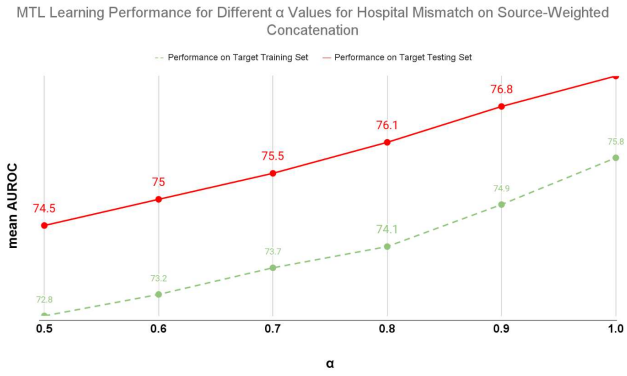


Figure 5: *Source-weighted Domain Adaptation performance on the target training and testing set for different* α *values. The domain adaptation technique hinders the training process, with* α = 1.0 *(S-weighted concatenation) resulting in the highest AUROC performance.*

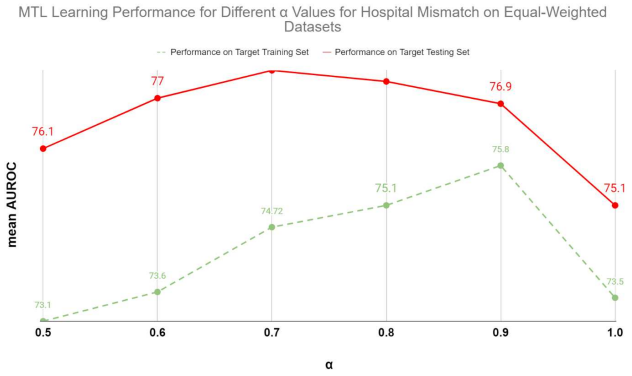


Figure 6: *Source-weighted Domain Adaptation performance on the target training and testing set for different* α *values. The domain adaptation technique hinders the training process, with* $\alpha = 1.0$ *(S-weighted concatenation) resulting in the highest AUROC performance.*

Figure 5 and 6 show the domain adaptation performance for different α parameter values for the hospital distribution mismatch. Figure 5, the source-weighted domain adaptation experiment, indicates that the use of multi-task learning for domain separation hinders the model performance towards both the training and testing target data. Conversely, Figure 6, the equal-weighted domain adaptation, indicates that multi-task learning benefits Sepsis prediction over both the training and testing target data.

The concatenation experiments indicated that increasing the importance of the target dataset resulted in negative returns. With source-weighted domain adaptation, the introduction of the dataset loss $L_{\text{data}}(z, \hat{z})$ either increases the importance of the target or decreases the importance of the source during training, resulting in lower performance. With equal-weighted concatenation, the source and the target populations both contribute equally to the model training. The introduction of $L_{\text{data}}(z, \hat{z})$ results in a 1.2% mean AUROC increase for both the gender and hospital distribution mismatch. For all three distribution mismatches, equal-weighted domain adaptation outperformed equal-weighted concatenation, but performed on par or worse than source-weighted domain adaptation. Hence, while the addition of a dataset loss may improve model performance towards the target population when the source and target hold equal importance during training, this technique is not suitable for the designed scenario due to the underrepresentative target training data.

Multiple Distribution Mismatches

The previous experiment investigated how the state-of-the-art techniques described in Section 3 leveraged a single source dataset towards a target population. This experiment investigates the performance of each method when used in a multi-source scenario. To create such a scenario, the single distribution mismatches from the previous experiment were combined, as shown in Figure 7.

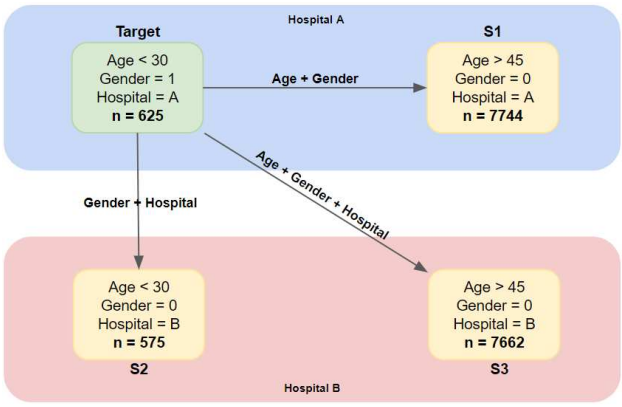


Figure 7: *Design of the multiple distribution mismatch datasets. The green box represents the target dataset, with the yellow boxes representing the source populations. With gender resulting in a small distribution mismatch, it was combined with age and hospital variables to create a larger mismatch. Though gender alone did not cause a strong mismatch, it is possible that combined with other mismatch variables, a bigger effect is observed. The bold values show the number of patients per dataset.*

In this scenario, the three source datasets contain a combination of distribution mismatches with respect to the target population. Specifically, gender was combined with age and/or hospital variables to further undersample each dataset and increase the increase the magnitude of the distribution gap between sources and target. Through the baselines of Table 3, we expect that the distribution mismatch between S_1 and the target is the smallest among all three sources, with the hospital variable providing the largest mismatch among the three variables. We additionally expect the performance of S_2 to be less consistent across the validation folds, due to the low cardinality of the dataset. Similarly to the single distribution mismatch, the baselines will be used to evaluate the performance of the methods in Table 5.

Results

Across all baseline results of Table 6, the standard deviation has increased in comparison to the single distribution mismatch baselines of Table 3. This increase in the standard deviation is likely due to the decrease in dataset size from the filtering per mismatch variable. However, despite the higher standard deviations, the baseline results present similar observations to the single distribution mismatch experiment.

The target-to-target baseline indicates that the target-trained model still has sufficient data to learn a predictive pattern towards the target population. The S_2 dataset has a comparable cardinality to the target population, and the S_2-to-S_2 baseline performs similarly to the target-to-target baseline, with a comparably higher standard deviation. In contrast, the S_1 and S_3 datasets have more training data,

S-Weighted Concatenation	Basic concatenation with no weights introduced. The model is biased towards the larger source datasets.
Equal-Weighted Concatenation	Weighted concatenation, with the sources and target datasets equally contributing to model training, $w_{source} = \frac{1}{\lvert D_{source}\rvert}$ and $w_{target} = \frac{1}{\lvert D_{target}\rvert}$.
Transfer Learning (Concatenation)	Pre-training on the concatenation of all source datasets (learning rate = 0.001), before fine-tuning on the target dataset (learning rate = $1 \cdot 10^{-5}$).
Transfer Learning (Sequential)	Sequential pre-training on S_3 (learning rate = 0.001), before fine-tuning on S_2, S_1 and the target dataset (learning rate = $1 \cdot 10^{-5}$).
Boosting (Grid)	Ensemble Learning using a source-trained model per source and target-trained model, with the weight β_i found using a grid-search over the target training set.
Boosting (Heuristic)	Ensemble Learning using a source-trained and target-trained model, with the weight β_i set using the baselines as a heuristic, $\beta_{\text{source}} = \frac{B_{S-T}}{\sum_{i=1}^{N+1} B_{i-T}}$ and $\beta_{\text{target}} = \frac{B_{T-S}}{\sum_{i=1}^{N+1} B_{i-T}}$, where the sum over $N+1$ sums the performance of each baseline from i to target (both sources and target).
S-Weighted Domain Adaptation (DA)	Multi-task learning over the source-weighted concatenation of source and targets to predict both Sepsis label and dataset of origin per sample, with α set using grid search over the training set.
Equal-Weighted Domain Adaptation (DA)	Multi-task learning using the equal-weighted concatenation of source and target to predict both Sepsis label and dataset of origin per sample, with α set using grid search over the training set.

Table 5: *Summary of methods for leveraging multi-source data.*

	Test - AUROC (in %)			
	S_1	**S_2**	**S_3**	**Target**
S_1	76.1 ± 3.1	62.6 ± 22.1	58.1 ± 12.0	74.5 ± 9.3
S_2	51.0 ± 6.2	73.1 ± 8.8	49.6 ± 10.8	57.9 ± 6.9
S_3	64.5 ± 4.8	75.4 ± 14.5	78.8 ± 4.0	65.2 ± 7.6
Target	57.5 ± 6.0	47.8 ± 14.7	53.01 ± 10.2	70.3 ± 6.8

Table 6: *Baseline mean AUROC results for Sepsis prediction with multiple distribution mismatches across the 10-fold cross-validation over patients. The value following the $\pm$ sign is the standard deviation across the folds. S_n represents a model trained on dataset D_n. The grey squares show model performance across only different sources and do not look at the target population.*

resulting in a higher more consistent baseline performance across the validation folds.

Furthermore, the performance of the three source datasets to the target population confirms the expectations held prior. A model trained on S_1 generalizes well to the target population, as they both contain data from the same hospital. S_2 and S_3 have a poor performance when tested on the target population, with the lower cardinality of S_2 resulting in performance closer to guessing.

The individual distribution mismatches results can also be found in the baseline results. For the age distribution mismatch, a model trained on S_2 is unable to generalize to S_3, while a model trained on S_3 generalizes well to S_2. A similar pattern can be seen between S_1 and the target population. For the hospital distribution mismatch, the mean AUROC of a model trained on the S_1 dataset decreases when tested on both S_2 and S_3. Similarly, the mean AUROC of a model trained on S_3 decreases when tested on datasets from Hospital A.
Overall, as the baselines offer consistent results within expectations, the remainder of this section will focus on the experimental results described in Table 7.

	Target AUROC (%)
S-weighted Concatenation	**83.2 ± 6.3**
Equal-weight Concatenation	80.0 ± 7.3
Transfer Learning (Concat.)	79.0 ± 7.5
Transfer Learning (Sequential)	81.0 ± 6.5
Boosting (Grid)	76.0 ± 8.3
Boosting (Heuristic)	75.1 ± 9.2
Source-weighted DA	**84.0 ± 6.7**
Equal-weighted DA	80.6 ± 6.6

Table 7: *Mean AUROC results of leveraging techniques for multiple distribution mismatches across the 10-fold cross-validation over patients. The value following the $\pm$ sign is the standard deviation across the folds.*

Generally, all methods outperformed the target-to-target and source-to-target baselines. However, with the exception of boosting, all methods performed very similarly, with S-weighted concatenation and domain adaptation obtaining the highest AUROC results. While both boosting methods outperformed the source-to-target baselines, they performed significantly worse than the S-weighted concatenation ($p_{\text{grid}} = 0.0338$ and $p_{\text{heuristic}} = 0.0424$), with a decrease of up to 8.1% mean AUROC. In comparison to concatenation-based transfer learning, though not significantly ($p_{\text{heuristic}} = 0.3126$), the boosting methods under-performed by 3-4%.

The concatenation results show a similar pattern found in the single distribution mismatch experiment. S-weighted concatenation outperforms all other concatenation methods, with the performance decreasing as the target importance

increases. More interestingly, for boosting, the performance difference between the grid search and the heuristic-derived weights is insignificant ($p = 0.5320$). The heuristic weighting approach gives higher importance to S_1 and the target dataset while lowering the importance of the datasets from Hospital B. Moreover, due to the low cardinality of S_2, its performance on the target set is very low, with the heuristic giving it a lower weight. This indicates that the baseline-derived weights for boosting can work as a reliable substitute for a full-search optimization.

Though statistically insignificant ($p = 0.5320$), sequential transfer learning outperformed concatenation-based transfer learning with a mean AUROC increase of 2%. Figure 8 shows the sequential transfer learning performance improvement across the source datasets through the fine-tuning process. Across all validation folds, the model performance increases as it continues to fine-tune to new source datasets. In two of the ten folds, S_3 provides an optimum AUROC across all datasets (including the target), indicating that the transfer learning technique can directly leverage the pattern across the hospital and age distribution mismatches. Appendix E provides an in-depth analysis of the sequential transfer learning process, notably regarding the fine-tuning dataset order.

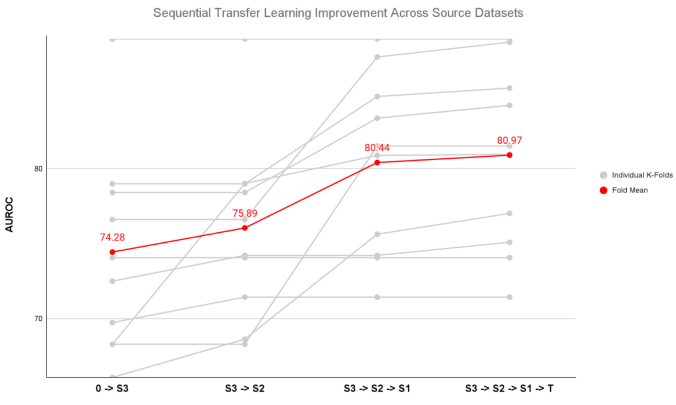


Figure 8: *Sequential Transfer Learning performance overview per fold. In red, is the mean performance across all folds. Across the folds the performance increases as the LSTM model is fine-tuned using each dataset, indicating no signs of a negative transfer.*

5 Discussion and Conclusion

In this study, we explored the applicability of several methods, namely Transfer Learning, Concatenation, Ensemble Learning and Domain Adaptation, for leveraging source datasets towards a target dataset in a time-series context. We specifically explored both the single-source and multi-source situations using a real-world medical dataset.

Applicability to other datasets

All methods explored in this study show potential applicability to other datasets beyond the scope of this study. However, it is important to understand that each method has a different set of data-specific requirements.

Transfer learning builds on the assumption that there exists shared features or patterns across all related populations, which allows for the pre-learning from a source domain to a target domain to be effectively utilized. The results demonstrate that transfer learning is most applicable when the source and target populations differ by a distribution mismatch (identifiable via the baselines). When employed in a situation where the source and target populations are drawn from very similar distributions, transfer learning was found less effective than concatenation or boosting. In such cases, we theorize that transfer learning risks un-learning predictors during the fine-tuning stage, resulting in a decrease in performance.

The concatenation techniques employed in this study combined the source and target datasets by concatenating their features and investigated how the presence of both source and target during training affected model performance towards the target population. The scope of this paper investigated the specific situation where the target dataset was underrepresentative. Within this scope, the concatenation experiments found that the presence of the target dataset was essential in improving model performance, but increasing its importance during training resulted in negative returns. However, across all distribution mismatches, concatenation consistently performed as one of the best-performing methods, making it a reliable technique to try on new uncertain distribution mismatch scenarios.

Ensemble learning aimed to utilize a similar theory to weighted concatenation, by weighting the combination of source and target models trained separately on each respective dataset. In doing so, ensemble learning aimed to combine the reliability of the source models and the noisy, but non-mismatched target-trained model. In situations with a small distribution mismatch, the weighted combination of model predictions yielded top-performing results with the additional benefit of a very stable performance across each cross-validation fold. However, as the magnitude of the distribution mismatch increased, ensemble learning failed to perform on par with other tested methods. Contrary to other investigated techniques, ensemble learning only combines the prediction outputs of models trained on a single dataset. By only combining model outputs, each model does not learn to adapt the source to the target well. Consequently, in situations with a noisy target, ensemble learning prioritizes the source-trained models, resulting in poorer target performance.

Multi-task learning, as a domain adaptation technique, assumes that learning the difference between source and target datasets aids the model in better capturing target-specific predictions. This approach allows for a more implicit consideration of the relevance and characteristics of the source dataset(s). Our results show that the use of an additional dataset separation loss improved model performance in cases where the source and the target equally contributed to model training. However, the benefits of the additional dataset loss fail to out-way the performance cost of increasing the target data importance during training. Consequently, it is difficult to evaluate if this technique is beneficial in cases where the training data is heavily underrepresentative.
The work of Yoon et al. [53] successfully made use of domain separation to learn the differences between several datasets using adversarial learning. As such, further research is needed to explore the specific applications of this technique and variations of it, such as domain confusion, for its applicability to leveraging source datasets.

The techniques investigated in this paper were unfortunately limitedly applied to a single real-world dataset in the medical field. Consequently, it is essential to exercise caution when considering the observed results to draw definitive conclusions. Though the exact performance differences observed are dataset specific, we expect very similar performances for each method in similar contexts (underrepresented target set, large source datasets) on different datasets. The disadvantages of transfer learning on two identical distributions have previously been observed [57], as have different aspects of both concatenation and ensemble learning, allowing us to more comfortably draw conclusions based on the observed results.

Baselines

The baselines used in this paper served as a measure of the distribution mismatch. These measurements were then applied as a heuristic to weigh the combination of source and target-trained models for both single and multiple distribution mismatches. In both mismatch scenarios, the baselines consistently served as a clear indication of the presence and magnitude of the distribution mismatch. Specifically, the baselines observed the three scenarios present in the data; no distribution mismatch, large distribution mismatch and partial distribution mismatch. It is important to note that the distribution mismatch observed by the baselines does not necessarily correspond to the real-world distribution mismatch. Though related and likely similar, the baselines specifically observe the difference between a model's performance on the train and the target set. It is likely the case that some data-specific model performance differences contribute to the observed mismatch, though not to the real-world expectation. This is illustrated by the gender distribution mismatch which is known to affect Sepsis prediction [14] but was not considered a mismatch based on the baseline observations.

Taken as a whole, the results show a promising direction for quantifying distribution mismatches and refining the hyper-parameter search space using the baselines as a heuristic. More research is needed to investigate the reliability of the baselines in different scenarios and datasets, notably cases of underrepresentative source datasets or feature mismatches. In the latter case, it is expected that the baselines may perform less reliably, as the feature mismatch correction may influence model performance along with existing real-world distribution mismatches. Unfortunately, addressing the feature mismatch is a complex task, frequently requiring domain expert knowledge or using complex feature-extracting methods (e.g. GANs), and as such, our work did not investigate this effect.

Distribution Mismatch and Time Complexity

Addressing multiple distribution mismatches is a more complex and time-consuming task than addressing a single mismatch. For all methods investigated, apart from domain adaptation, the minimal number of parameters required for optimization corresponds to the number of total datasets (source(s) and target). Hence, as the number of source datasets increases, the number of parameters requiring joint optimization increases. With ensemble learning, the process is less time-consuming, as each model is trained on a single respective dataset. However, for methods such as sequential transfer learning or concatenation, optimization of the hyper-parameters requires the repeated re-training of the models, a time-consuming process. The exception to this time-consuming process is domain adaptation, which relies on implicitly learning the separability between the source(s) and target during the training process. The downside of this implicit assumption is the lack of understanding of the models decision, making it difficult to interpret unexpected results.

In many cases, it may be too difficult or time-consuming to fully optimize each parameter. Furthermore, with methods such as transfer learning, the hyper-parameters heavily affect the final model performance. With sequential transfer learning, for instance, a too high learning rate during the fine-tuning stages can cause much previously learnt information to be lost, resulting in an incomplete leveraging of source datasets. In the case of ensemble learning in a multi-mismatch context, the grid search optimization is too time-consuming. Our work showcases the potential of using baselines as a heuristic to set the weighting parameters between the models. Though not tested on all methods, we expect a similar application of the baseline on concatenation weights to also perform well, notably in cases where the

target dataset is increasingly severely underrepresentative.

Time-series vs Cross-sectional Data

Leveraging or transferring knowledge with time-series data poses a unique challenge compared to cross-sectional data, due to the temporal dependencies in the data, and the frequent predictive nature of the task [17]. Though there exist network architectures which can facilitate the processing of time-series data (eg. LSTMs or CNNs), aligning sequential patterns across different datasets is a complex task. Within our investigated methods, transfer learning and concatenation are prone to suffering from source and target sequential pattern differences, with a risk of a negative transfer. Ensemble learning or MTL domain adaptation, due to the design of the approaches, is likely more robust against such differences. With ensemble learning, due to the reliance on model outputs, the target model can be prioritized, preventing negative influence. Similarly with MTL domain adaptation, by learning to identify each dataset, the model could implicitly learn not to consider sources of negative influence.

While our study focused on time-series data, previous works have explored the applicability of some of these techniques to cross-sectional data [43] and found them to be efficient for leveraging cross-sectional source datasets. As described in Section 2, for example, transfer learning is frequently employed in the image processing domain, where the similarity in the images is a lesser constraint in comparison to temporal sequences. Though this work did not investigate the application of these techniques in a similar context of target underrepresentativeness on cross-sectional data, it stands as an interesting topic for future investigation. We do however expect that the baselines can still identify existing distribution mismatches between source(s) and target, and allow for a refined optimization search space as a heuristic.

Future Work

The scope of this paper was limited to using an LSTM deep learning model, in the situation where the source dataset(s) contain sufficient training samples, and the target dataset is underrepresentative. Within this scope, many methods relying on unique architectures could not be investigated. Consequently, an interesting and required future work would be to investigate these additional techniques within a similar context and determine how these techniques can be compared despite the architectural differences (advantages and disadvantages). Works in this field using GANs, such as RadialGAN [53], have been explored in both a cross-sectional and time-series data context, but little work has been done on situations with small datasets, due to the notorious GANs data requirements.

Hierarchical Modelling is an approach which was not considered for this paper due to external circumstances. Nevertheless, it is a powerful technique for leveraging multiple datasets, notably where the mismatches are known. Similarly, recent works with Hidden Markov Models make use of LSTMs to provide "memoryful" state transitions to focus on past state realizations that best predict future states [3]. Investigating these techniques would be an interesting avenue for future research. With both HMM+LSTM and Hierarchical models, most investigated leveraging techniques remain applicable, though some model architecture changes may be required.

Investigating leveraging techniques towards an unseen target population also falls within the future work stemming from this paper. In such cases, active or semi-supervised learning can be interesting methods to begin the research, to restrict the leveraging process to work within very stringent real-world practical conditions. Similarly, meta learning is also an interesting avenue of research in finding model optimizations that allow for a swift adaptation to new domains, both observed and new.

A final future work of interest would be the investigation of interpretable machine learning. While all of the techniques tested successfully leveraged the source dataset towards the target population, it is unclear how most techniques achieved this. Notably for domain adaptation, it is unclear under what conditions the addition of a dataset-specific loss aids the model in separating the source from the target, or if the opposite loss (aiming for domain confusion) would benefit the task-specific goal. While some ideas can be theorized, such as for transfer learning, it is still difficult to understand the underlying decisions made by a deep learning model. Providing insights into the underlying distribution differences through the leveraging process is a very exciting avenue for future research.

Conclusion

In this study, we successfully employed state-of-the-art techniques to leverage related sources of time-series data to improve the performance of an LSTM model towards a target population with underrepresentative data. To do so, we defined baselines to quantify the magnitude of distribution mismatches present between the source and target populations within the same feature space. In addition, we conducted a comprehensive evaluation of several techniques, assessing their suitability towards leveraging related datasets across different distribution mismatch scenarios.
When concatenating the source(s) and target data, we found that the presence of the target data was crucial in improving

model performance. However, contrary to expectations, elevating the target importance during training resulted in negative returns. Additionally, we found transfer learning to be one of the most effective approaches to use in situations where a larger distribution mismatch is present. Conversely, ensemble learning demonstrated advantages when dealing with minor distribution mismatches, but failed to leverage sources differing from the target population by a larger mismatch. Finally, our results highlighted the use of baseline-derived model parameters as a superior alternative to search-based parameter optimization in the instance of underrepresentative target training data.

Our findings contribute to understanding how the transfer of knowledge in the time-series domain can be achieved, and help further research specialized approaches for successful knowledge transfer in real-world applications.

References

[1] ADHIKARI, R., AND AGRAWAL, R. K. Combining multiple time series models through a robust weighted mechanism. In *2012 1st International Conference on Recent Advances in Information Technology (RAIT)* (2012), pp. 455–460.

[2] AFROSE, S., SONG, W., NEMEROFF, C. B., LU, C., AND YAO, D. D. Subpopulation-specific machine learning prognosis for underrepresented patients with double prioritized bias correction. *Communications Medicine 2*, 1 (Sep 2022), 111.

[3] ALAA, A. M., AND VAN DER SCHAAR, M. Forecasting individualized disease trajectories using interpretable deep learning, 2018.

[4] ALAA, A. M., YOON, J., HU, S., AND VAN DER SCHAAR, M. Personalized risk scoring for critical care prognosis using mixtures of gaussian processes, 2016.

[5] AN, J., YING, L., AND ZHU, Y. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients, 2020.

[6] ARNX, A. First neural network for beginners explained (with code), Aug 2019.

[7] BUI, H., PHUNG, D., AND VENKATESH, S. Hierarchical hidden markov models with general state hierarchy.

[8] CALZONE, O. An intuitive explanation of lstm, Apr 2022.

[9] CHANG, Q., AND HU, J. Application of hidden markov model in financial time series data. *Security and Communication Networks 2022* (Apr 2022), 1465216.

[10] CHEN, W., HAN, H., HUANG, B., HUANG, Q., AND FU, X. Variable-weighted linear combination model for landslide susceptibility mapping: Case study in the shennongjia forestry district, china. *ISPRS International Journal of Geo-Information 6*, 11 (Nov 2017), 347.

[11] DRIDI, A., AFIFI, H., MOUNGLA, H., AND BOUCETTA, C. Transfer learning for classification and prediction of time series for next generation networks. In *ICC 2021 - IEEE International Conference on Communications* (2021), pp. 1–6.

[12] DUBEY, S. R., SINGH, S. K., AND CHAUDHURI, B. B. Activation functions in deep learning: A comprehensive survey and benchmark, 2022.

[13] DUTTA, R., BLOMSTEDT, P., AND KASKI, S. Bayesian inference in hierarchical models by combining independent posteriors, 2016.

[14] EACHEMPATI, S. R., HYDO, L., AND BARIE, P. S. Gender-Based Differences in Outcome in Patients With Sepsis. *Archives of Surgery 134*, 12 (12 1999), 1342–1347.

[15] ELSWORTH, S., AND GÜTTEL, S. Time series forecasting using lstm networks: A symbolic approach, 2020.

[16] ESMAEL, B., ARNAOUT, A., FRUHWIRTH, R. K., AND THONHAUSER, G. Improving time series classification using hidden markov models. In *2012 12th International Conference on Hybrid Intelligent Systems (HIS)* (2012), pp. 502–507.

[17] FAWAZ, H. I., FORESTIER, G., WEBER, J., IDOUMGHAR, L., AND MULLER, P.-A. Transfer learning for time series classification. In *2018 IEEE International Conference on Big Data (Big Data)* (dec 2018), IEEE.

[18] GHAHRAMANI, Z. An introduction to hidden markov models and bayesian networks. *IJPRAI 15* (02 2001), 9–42.

[19] GOLDBERGER, A. L., AMARAL, L. A., GLASS, L., HAUSDORFF, J. M., IVANOV, P. C., MARK, R. G., MIETUS, J. E., MOODY, G. B., PENG, C. K., AND STANLEY, H. E. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation 101*, 23 (June 2000), E215–20.

[20] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[21] HUH, M., AGRAWAL, P., AND EFROS, A. A. What makes imagenet good for transfer learning?, 2016.

[22] IOFFE, S., AND SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[23] JIN, R., AND NIU, Q. Automatic fabric defect detection based on an improved yolov5. *Mathematical Problems in Engineering 2021* (09 2021), 1–13.

[24] JIN, X., PARK, Y., MADDIX, D. C., WANG, H., AND WANG, Y. Domain adaptation for time series forecasting via attention sharing, 2022.

[25] KARIMI, D., WARFIELD, S. K., AND GHOLIPOUR, A. Critical assessment of transfer learning for medical image segmentation with fully convolutional neural networks, 2020.

[26] KOUW, W. M., AND LOOG, M. An introduction to domain adaptation and transfer learning, 2018.

[27] LAKIN, S. M., KUHNLE, A., ALIPANAHI, B., NOYES, N. R., DEAN, C., MUGGLI, M., RAYMOND, R., ABDO, Z., PROSPERI, M., BELK, K. E., MORLEY, P. S., AND BOUCHER, C. Hierarchical hidden markov models enable accurate and diverse detection of antimicrobial resistance sequences. *Commun. Biol. 2*, 1 (Aug. 2019), 294.

[28] LEE, G., RUBINFELD, I., AND SYED, Z. Adapting surgical models to individual hospitals using transfer learning. *2012 IEEE 12th International Conference on Data Mining Workshops* (2012), 57–63.

[29] LEE, S. Y., LEI, B., AND MALLICK, B. Estimation of COVID-19 spread curves integrating global data and borrowing information. *PLoS One 15*, 7 (July 2020), e0236860.

[30] LELLI, F. Neural networks: The basics and a collection of youtube videos, Feb 2020.

[31] MARTIN, G. S., MANNINO, D. M., AND MOSS, M. The effect of age on the development and outcome of adult sepsis*. *Critical Care Medicine 34*, 1 (2006).

[32] MCCARRON, C. E., PULLENAYEGUM, E. M., THABANE, L., GOEREE, R., AND TARRIDE, J.-E. Bayesian hierarchical models combining different study types and adjusting for covariate imbalances: a simulation study to assess model performance. *PLoS One 6*, 10 (Oct. 2011), e25635.

[33] MCDERMOTT, M. B. A., NESTOR, B., KIM, E., ZHANG, W., GOLDENBERG, A., SZOLOVITS, P., AND GHASSEMI, M. A comprehensive evaluation of multi-task learning and multi-task pre-training on ehr time-series data, 2020.

[34] MEHLIG, B. *Machine Learning with Neural Networks*. Cambridge University Press, oct 2021.

[35] PARIHAR, R., DHIMAN, A., KARMALI, T., AND R, V. Everything is there in latent space: Attribute editing and attribute style manipulation by StyleGAN latent space exploration. In *Proceedings of the 30th ACM International Conference on Multimedia* (oct 2022), ACM.

[36] PASCANU, R., MIKOLOV, T., AND BENGIO, Y. On the difficulty of training recurrent neural networks, 2013.

[37] PLESTED, J., AND GEDEON, T. Deep transfer learning for image classification: a survey, 2022.

[38] PRUDÊNCIO, R., AND LUDERMIR, T. A machine learning approach to define weights for linear combination of forecasts. pp. 274–283.

[39] RAFIEI, A., REZAEE, A., HAJATI, F., GHEISARI, S., AND GOLZAN, M. Ssp: Early prediction of sepsis using fully connected lstm-cnn model. *Computers in Biology and Medicine 128* (2021), 104110.

[40] REYNA, M., JOSEF, C., JETER, R., SHASHIKUMAR, S., MOODY, B., WESTOVER, M. B., SHARMA, A., NEMATI, S., AND CLIFFORD, G. D. Early prediction of sepsis from clinical data: The PhysioNet/Computing in cardiology challenge 2019, 2022.

[41] REYNA, M. A., JOSEF, C. S., JETER, R., SHASHIKUMAR, S. P., WESTOVER, M. B., NEMATI, S., CLIFFORD, G. D., AND SHARMA, A. Early prediction of sepsis from clinical data: The physionet/computing in cardiology challenge 2019. *Critical Care Medicine 48*, 2 (2020).

[42] RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., BERG, A. C., AND FEI-FEI, L. Imagenet large scale visual recognition challenge, 2014.

[43] SHAO, J., HUANG, Z., ZHU, Y., ZHU, J., AND FANG, D. Rotating machinery fault diagnosis by deep adversarial transfer learning based on subdomain adaptation. *Advances in Mechanical Engineering 13*, 8 (2021), 16878140211040226.

[44] SKOUNAKIS, M., CRAVEN, M., AND RAY, S. Hierarchical hidden markov models for information extraction. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (08 2003).

[45] SOLÍS, M., AND CALVO-VALVERDE, L.-A. Performance of deep learning models with transfer learning for multiple-step-ahead forecasts in monthly time series, 2022.

[46] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 56 (2014), 1929–1958.

[47] TAMPOSIS, I. A., THEODOROPOULOU, M. C., TSIRIGOS, K. D., AND BAGOS, P. G. Extending hidden markov models to allow conditioning on previous observations. *Journal of bioinformatics and computational biology 16 5* (2017), 1850019.

[48] WANG, S., AND JIANG, J. Learning natural language inference with lstm, 2016.

[49] WEBER, M., AUCH, M., DOBLANDER, C., MANDL, P., AND JACOBSEN, H.-A. Transfer learning with time series data: A systematic mapping study. *IEEE Access 9* (12 2021), 165409–165432.

[50] WIENS, J., GUTTAG, J., AND HORVITZ, E. A study in transfer learning: leveraging data from multiple hospitals to enhance hospital-specific predictions. *J. Am. Med. Inform. Assoc. 21*, 4 (July 2014), 699–706.

[51] WILSON, G., DOPPA, J. R., AND COOK, D. J. Multi-source deep domain adaptation with weak supervision for time-series sensor data, 2020.

[52] YOON, J., DAVTYAN, C., AND VAN DER SCHAAR, M. Discovery and clinical decision support for personalized healthcare. *IEEE J. Biomed. Health Inform. 21*, 4 (July 2017), 1133–1145.

[53] YOON, J., JORDON, J., AND VAN DER SCHAAR, M. Radialgan: Leveraging multiple datasets to improve target-specific predictive models using generative adversarial networks, 2018.

[54] ZHOU, Y., HE, Z., LU, K., WANG, G., AND WANG, G. Preserve pre-trained knowledge: Transfer learning with self-distillation for action recognition, 2022.

[55] ZHU, C. Early prediction of sepsis using LSTM networks, 2021.

[56] ZHU, J., CHEN, N., AND SHEN, C. A new multiple source domain adaptation fault diagnosis method between different rotating machines. *IEEE Transactions on Industrial Informatics 17*, 7 (2021), 4788–4797.

[57] ZHUANG, F., QI, Z., DUAN, K., XI, D., ZHU, Y., ZHU, H., XIONG, H., AND HE, Q. A comprehensive survey on transfer learning, 2019.

Appendix A - Data Preprocessing

Sliding Window vs Expanding Window

The goal of the PhysioNet Challenge is to predict the occurrence of Sepsis in new patients. Within this setting, a difficulty to address is the difference between patients' lengths of stay at the ICU. Due to the large differences ranging from 10-330h, it is not feasible to pass the entire EHR of a patient into a machine learning model, requiring the data to be sequenced.
On state-of-the-art solution is to use a sliding window over the data. This approach extracts local segments of fixed length out of the time-series hourly EHR data, allowing for a consistent input length. A diagram of the approach is shown in Figure 9.

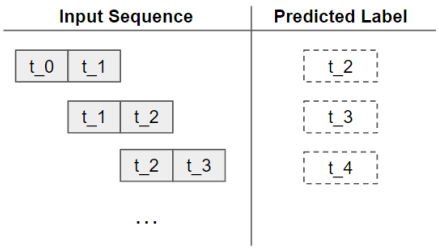


Figure 9: *Diagram demonstrating the windows created using a sliding window. Each window has an identical length and dimensions, with the number of dimensions corresponding to the number of selected EHR features. The predicted label consists of the binary sepsis label. The blocks in the input sequence represent patient EHR samples taken after pre-processing.*

An alternative approach for processing the data into usable sequences is to use varying-length sequences. While sliding windows are easier to create and interpret, varying-length sequences allow for longer-term dependencies to be captured by a machine learning model, and for deep learning models to learn the appropriate window sizes to observe for a prediction. There exists several methods by which one can construct varying-length sequences, such as random window lengths, or several fixed window sizes. However, Congxing Zhu et al. [55] showed that using expanding windows starting from the point of admission to the ICU allowed an LSTM model to extract sepsis-predictive patterns in the data. A diagram of the approach is shown in Figure 10.

A surprising performance difference was observed when comparing both sequencing approaches. As not the focus of this paper, no thorough experiments were conducted, though it is important to note that our experiments were conducted using varying-length sequences. Switching to a sliding-window approach may result in alternative conclusions, and require further investigation.

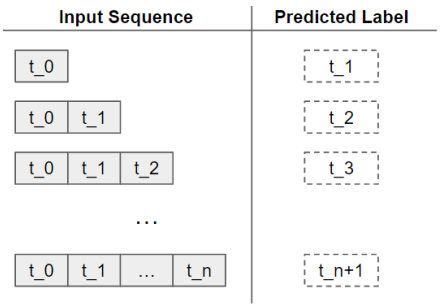


Figure 10: *Diagram demonstrating the varying length windows created from the point of admission to the ICU. The predicted label consists of the binary sepsis label. The blocks in the input sequence represent patient EHR samples taken after pre-processing.*

Data Cleaning

The cleaning of the data was kept minimal, loosely based on the pre-processing steps of both [55] and [39]. As the goal of this work was to investigate the leveraging of related datasets, less of a focus was placed on the specific processing of the data to maximize model performance.
All EHR components, outside of demographic information, were kept to create a high-dimensional input into the model. No normalization or standardization was performed on the data, with all missing values being first forward, then backward filled. This specific choice was influenced by Zhu et al.'s experiments [55].
To analyze the performance of the model and the improvements brought by the different leveraging techniques, the dataset was split into training and validation sets. To this end, all experiments were validated using a 10-fold Cross-Validation across the patients, with each fold containing data from a unique set of patients. Due to the large imbalance between positive and negative class samples, the cross-validation was stratified, allowing for sufficiently stable performance estimates across each fold.

Appendix B - Age Histogram

Figure 11 contains the age distribution of patients across hospitals A and B. As can be observed, the age distribution is skewed towards the older population, making it a better fit for a source dataset.

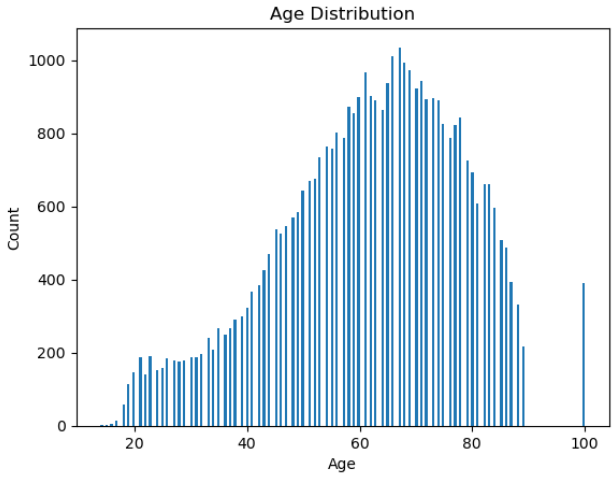


Figure 11: *Age distribution across both hospitals A and B. The final chosen thresholds for the scenario were to take patients younger than 30 years old as the smaller dataset, and patients older than 45 years old as the source dataset.*

Appendix C - Age Boosting Grid Search

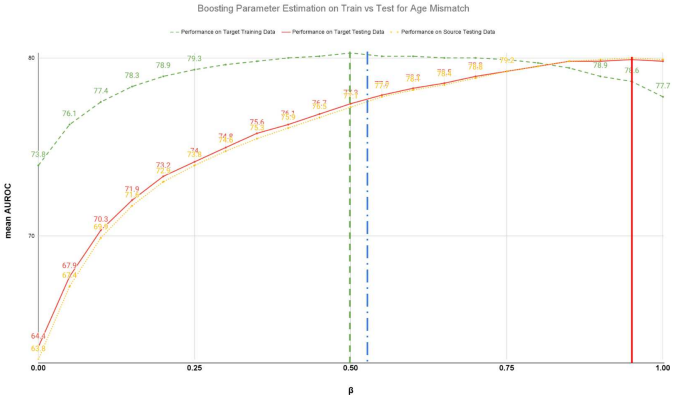


Figure 12: *Figure showing the boosting performance with different* β *parameter values for the hospital distribution mismatch. The green (short dash) vertical line shows the optimal* β *found during training. The red (solid) vertical line shows the optimal* β *for the test set, and the blue (dash/dot) vertical line shows the baseline derived* β *value.* β *= 1 indicates only source model usage.*

Figure 12 shows the ensemble learning results for different β parameter values. The green curve, indicating the performance of the final model on the target training data, finds an optimal balance of model around 0.5, where both the source and target-trained models have equal importance to the final predictions (indicated via the green/dashed vertical line).
The blue vertical line (dash/dotted) indicates the heuristically derived β value using the baseline performances. As the source-trained and target-trained models performed similarly on the target dataset in the baseline results, the weight of both models also balances around 0.5-0.55. The true test-set optimum (red/straight line) lies around 0.9, indicating that the target training set is too underrepresentative to allow for a proper adjustment of hyper-parameters. The heuristic weight, while similar, lies closer to the true optimum and does not rely on the undersampled training data, making it suitable in cases where the target dataset is more severely undersampled.

Appendix D - Metric Selection

We considered two commonly used performance metrics, Area Under the Receiver Operated Characteristic Curve (AUROC), and Area Under the Precision-Recall Curve (AUPRC) for our study. Our experiments showed that both metrics yielded similar results, allowing us to draw the same conclusions. Although AUPRC has advantages over the AUROC metric for imbalanced datasets or skewed class distributions, we chose to use the metric that provided better readability and connections to prior works. Nevertheless, we acknowledge the importance of the AUPRC metric in specific scenarios and encourage further research to explore its applicability to our problem. Table 8 and Table 9 show the baseline and experimental results for the single distribution mismatches of the AUPRC values.

	AUPRC (in %)		
	Gender	**Hospital**	**Age**
Source to Source	10.8 ± 2.2	8.8 ± 1.7	11.0 ± 2.0
Source to Target	9.0 ± 2.2	7.0 ± 1.4	11.0 ± 4.1
Target to Target	3.7 ± 0.9	4.9 ± 1.1	6.3 ± 3.3
Target to Source	3.7 ± 0.9	2.1 ± 0.8	3.6 ± 0.7

Table 8: *Baseline mean AUPRC results for Sepsis prediction with a single distribution mismatch across the 10-fold cross-validation over patients. The value following the ± sign is the standard deviation across the folds.*

The AUPRC baselines allow us to draw similar conclusions to the AUROC baselines of Table 3. For the gender distribution mismatch, the source-trained model decreases by 0.018 in mean AUPRC performance when tested on the target dataset, in comparison to the testing on the source dataset. However, the target-trained model performs identically on both the source and target dataset, indicating that both distributions are very similar.

For the hospital distribution mismatch, the AUPRC baselines show an identical pattern to the AUROC baselines. Both the source-trained and target-trained models decrease in performance when tested on the respective mismatched population. Finally, for the age distribution mismatch, the source-to-source and source-to-target baselines hold the same AUPRC, indicating no distribution mismatch. Moreover, with the poor target performance on the source set, the baselines indicate the same underrepresentativeness previously observed.

The experimental results for the single distribution mismatch allow us to draw comparable conclusions to the results of Table 4, with one notable difference. Though the relative difference between most techniques is similar between the AUPRC and AUROC results, the transfer learning experiments show better performance results with the AUPRC metric. For all three distribution mismatches, the transfer learning technique performed on par or better than all leveraging techniques.

The AUROC metric tends to give more weight to the majority class in cases where the datasets are highly imbalanced. In contrast, the AUPRC metric disregards the True Negative samples, giving more importance to the minority class. Though the AUROC results indicated a loss of information when transfer learning is used in cases with a small distribution mismatch, the AUPRC results indicate that transfer learning is still one of the highest-performing techniques.

	AUPRC (in %)		
	Gender	**Hospital**	**Age**
S-weighted Concat.	10.0 ± 2.3	9.6 ± 2.0	10.6 ± 4.4
Equal-weight Concat.	7.7 ± 11.0	8.7 ± 2.4	13.2 ± 8.6
T-weighted Concat.	3.7 ± 0.8	6.0 ± 1.3	7.1 ± 4.9
Transfer Learning	10.4 ± 2.4	11.1 ± 2.5	13.1 ± 5.2
Boosting (Grid)	9.2 ± 0.9	7.4 ± 1.4	11.0 ± 3.2
Boosting (Heuristic)	9.2 ± 1.0	7.1 ± 1.3	11.0 ± 3.2
Domain Adaptation	9.3 ± 1.1	9.4 ± 1.8	11.7 ± 3.9

Table 9: *Mean AUPRC results of leveraging techniques for a single distribution mismatch across the 10-fold cross-validation over patients. The value following the ± sign is the standard deviation across the folds.*

Appendix E - Sequential Transfer Learning

Figure 8 showed the sequential transfer learning process using an order from largest to smallest combined mismatches. Figure 13 and 14 present a similar experiment with different orders between the training datasets.

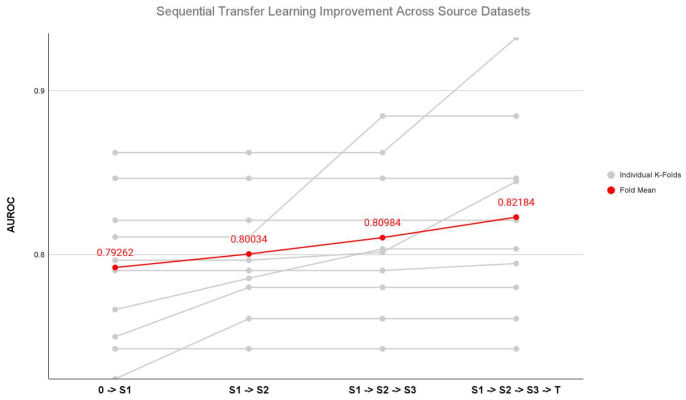


Figure 13: *Sequential Transfer Learning performance overview per fold. In red, the mean performance across all folds. Across the folds the model performance increases as each dataset is used for fine-tuning, indicating no signs of a negative transfer.*

Figure 13 inverts the order of the source datasets, such that the smallest mismatched dataset is trained on first, with the more distant sources used in the fine-tuning process. Similarly to Figure 8, the target is the final dataset used for fine-tuning, to ensure that the model is best fit to the target dataset.
In comparison to Figure 8, inverting the order of the source datasets resulted in different trends across the sequential fine-tuning stages. While eight of the ten validation folds of Figure 8 showed a continuous increase as the increasingly similar dataset to the target was used, Figure 13 shows that seven of the validation folds do not increase in when a model trained on S_1 is fine-tuned on S_2 and S_3. Moreover, in four of these seven examples, the optimum AUROC is found following the basic training on S_1, indicating that S_1 contains sufficiently related data to fully train a performing target-specific model.

Figure 14 first trains on the target dataset, before attempting to sequentially transfer information from the source datasets. This approach was expected to underperform, as the "furthest" source dataset is trained on during the last step. As a consequence, while information is leveraged across all datasets, the focus of the final model is the population of S_3. The results show such a scenario, where the mean AUROC (red) performance across the validation folds is lower than that of Figure 8 and 13.
Unexpectedly, across five of the ten folds, the fine-tuning on S_3 produces the largest increase in AUROC of all datasets.

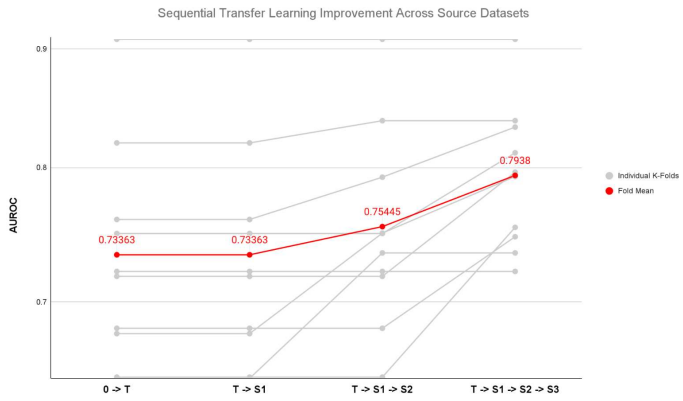


Figure 14: *Sequential Transfer Learning performance overview per fold. In red, the mean performance across all folds. Across the folds the performance remains increases as each dataset is fine-tuned upon, indicating no signs of a negative transfer.*

These results further indicate that the transfer learning technique can leverage the learnings across the hospital and age distribution mismatches.

Appendix F - Further Underrepresented Target Data

This experiment further halved the previously undersampled the target training data, to investigate the situation where the target training data provides insufficient information for the LSTM model to learn any usable pattern.

	AUROC (in %)		
	Gender	**Hospital**	**Age**
Source to Source	78.9 ± 1.2	80.2 ± 2.3	80.1 ± 1.3
Source to Target	78.6 ± 1.7	68.3 ± 3.1	79.7 ± 4.8
Target to Target	61.6 ± 6.8	60.2 ± 2.2	58.5 ± 8.6
Target to Source	60.0 ± 6.4	53.0 ± 1.1	55.6 ± 4.5

Table 10: *Baseline mean AUROC results for Sepsis prediction with a single distribution mismatch across the 10-fold cross-validation over patients using a heavily undersampled training set. The value following the $\pm$ sign is the standard deviation across the folds.*

Table 10 shows a similar set of baselines to Table 3, where the target training data is further undersampled to 2.5% of the original training fold. Across all target-trained models, the baseline performance decreased when predicting both the source and the target datasets. The source-trained baselines perform similarly to Table 3, as the models and training data were kept identical.

Though the baselines offer no new insights about dataset representativeness or distribution mismatches, this further undersampling of the target training data provides a clearer justification for the use of heuristic-derived parameters.

With the prior baselines of Table 3, the performance of the source-trained and target-trained models on the target testing set were comparable within a ± 10 mean AUROC difference. Consequently, baseline-derived parameters broadly weighed the source and target equally, with the source being slightly favored. With the following baselines, the source-trained models substantially outperform the target-trained model for all distribution mismatches. The baseline-derived weights hence weigh the source more than the target dataset, providing parameter estimation closer to the test set-derived optimum than a search strategy over the training data.

General Background on Deep Learning

This chapter provides a background of the theoretical information on deep learning for clear insight. We start with a basic overview of what neural networks are and how they work. We then provide a detailed explanation into the workings of Long Short-Term Memory neural networks (LSTMs) and Generative Adversarial Networks (GANs). We then look into the different techniques used in this paper, namely Concatenation, Weighted Concatenation, Transfer Learning, Ensemble Learning and Domain Adaptation. These background sections provide complementary information for the techniques utilized in this paper.

Neural Networks

Neural networks are mathematical computational models inspired by the structure of function of biological systems [34]. They are designed to learn from data, and are usually used to model complex relationships between inputs and outputs or to find patterns in the data based on that learning. A neural network consists of an interconnected group of computational units called neurons. Each neuron receives input data, performs a computation, and produces an output. The computation performed by a neuron involves a weighted combination of its inputs, followed by the application of an activation function [12]. Figure 15 shows the mathematical model of a neuron.

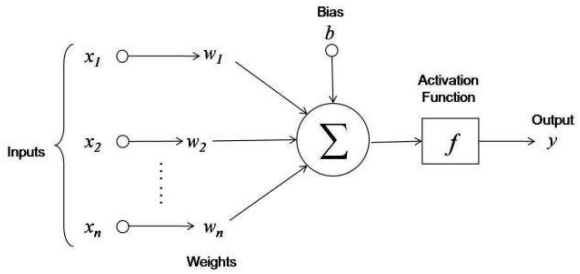


Figure 15: *Annotated diagram of a Neuron by Arnx [6]. The input data is combined with a set of weights and biases, before being passed into an activation function. The resulting output is then used in the later stages of a complex network made of many neurons.*

Given a neuron with n inputs $\{x_1, x_2 \dots x_n\}$, a set of weights $\{w_1, w_2 \dots w_n\}$ and a bias term b, the output of the neuron to the activation function consists of a weighted sum, described below:

$$z = \sum_{i=1}^{n} w_i x_i + b$$

While every input is multiplied by a unique learnt weight, the neuron has a separate bias term that is shared across all inputs of the neuron. This means that the bias term is not specific to any particular input-to-output connection, but influences the overall output of the neuron. The activation function $f(z)$, introduces a non-linearity to the neuron's output y, allowing it to represent intricate and complex patterns in the data.

A neural network consists of many layers of interconnected neurons, shown in Figure 16. The input layer receives the data, and subsequent hidden layers perform computations based on the output $f(z)$ of the previous layer. The output layer produces a network's *decision* using a loss function.

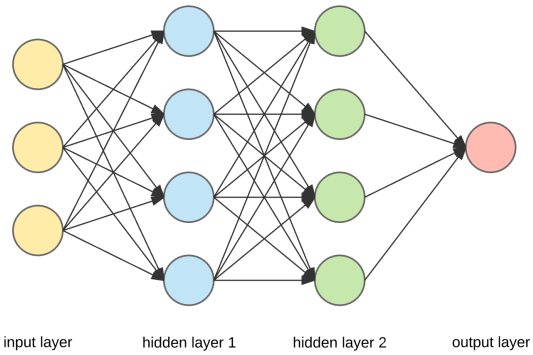


Figure 16: *Diagram of a neural network by Lelli et al. [30]. This diagram represents a three-layer fully connected network with 2 hidden layers. Fully connected is defined by every input having a mapping (arrow) to every output.*

To make accurate predictions or learn patterns of interest, the neural network is trained on labelled data, a process called supervised learning. This process involves adjusting the weights and biases of each neuron to minimize the difference between a predicted output and the ground truth. The most common method for training a neural network is called backpropagation [34], which uses the chain rule of calculus to calculate the gradient of the network's weight with respect to the output loss function. The calculated gradients are then used to update the weights and biases of the network iteratively using an optimization algorithm such as stochastic gradient descent. The following sections delve briefly into specific details of activation and loss functions, as well as normalization and regularization techniques.

Activation functions

Activation functions transform the weighted sum of inputs into the final output of a neuron, allowing the neural network to model complex non-linear relationships in the data [12]. Different activation functions have unique properties which make them suited for different types of problems. The choice of activation function is very task and network-specific. Below we describe the four most common activation functions, with each example shown in Figure 17.

- **ReLU:** Rectified Linear Units (ReLU), defined as $f(z) = \max(0, z)$ sets all negative values of the weighted sum z

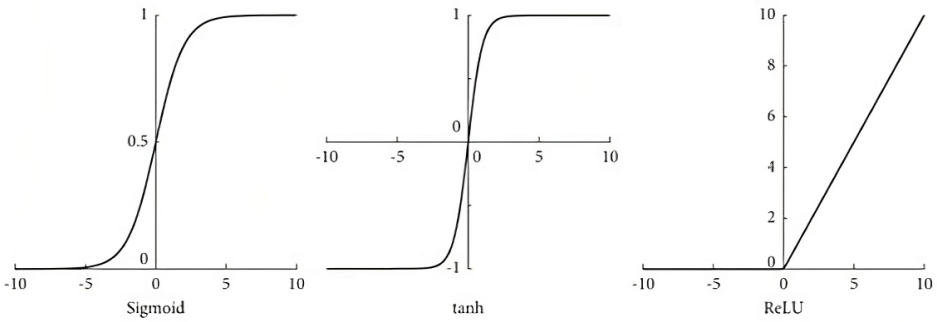


Figure 17: *Diagram of the Sigmoid, tanh and ReLU activation functions [23]. The sigmoid activation function collapses real numbers into a range [0,1], whereas tanh maps the real numbers to a range between [-1,1]. ReLU sets all negative numbers to 0, and has a linear slope for positive values.*

to zero, while leaving positive values unchanged. ReLU is computationally efficient and addresses the vanishing gradient problem, making it one of the most commonly used activation functions.

- **Sigmoid:** The sigmoid activation function $f(z) = \frac{1}{1+e^{-z}}$ squashes the weighted sum z to a range between 0 and 1, providing a probability-like value. This property of the sigmoid activation function makes it very suitable for binary classification or probability estimation problems. However, a downside of the sigmoid activation function, is that it suffers from the vanishing gradient problem, which limits its effectiveness and applicability [12].

- **Tanh:** The hyperbolic tangent function (tanh), $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, similarly to the sigmoid activation function, maps the weighted sum z of the neuron to a range between -1 and 1. The tanh activation function creates a centered output, allowing for better backpropagation. Similarly to the sigmoid function, it suffers from the vanishing gradient problem [12].

- **Softmax:** The softmax activation function $f(z) = \frac{e^{z_i}}{\sum_j^K e^{z_j}}$, where K is the number of classes, ensures that the output probabilities all sum to 1. Effectively, the softmax activation function normalizes the input values and transforms them into probabilities. In a classification scenario, the probabilities indicate the network's belief for each class, making it suitable for many classification tasks. In cases of binary classification, the softmax activation simplifies to the sigmoid.

Loss Functions

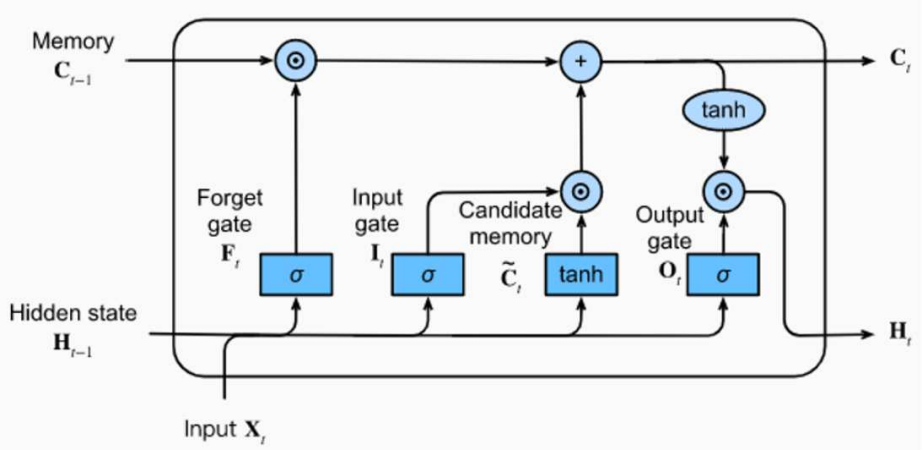
Loss functions play a pivotal role in the training process of deep learning networks, as a tool to quantitatively evaluate the disparity between predicted output labels and their corresponding ground truth labels. Two commonly employed loss functions in classification tasks are Cross-Entropy (CE) and Binary Cross-Entropy (BCE). CE is most frequently used in multi-class classification scenarios, where each sample is assigned to a single class from a set of classes. IT specifically quantifies the dissimilarity between the predicted class probabilities and the true class labels. BCE, on the other hand, is a particular use-case of CE, restricted to binary classification tasks where the sample is assigned to one of two classes. By optimizing these loss functions, deep learning models can leverage large datasets to improve their performance across a wide range of tasks, such as time-series prediction, computer vision or natural language processing.

Batch Normalization

Batch Normalization is a technique in deep learning that aims to address the internal covariate shift problem during training, a problem occurs when the distribution of the input to each layer changes as the network is training [36]. Batch Normalization normalizes the intermediate activations of a neural network layer across a mini-batch of samples. This normalization standardizes the distribution of layer inputs, stabilizing the optimization process. This process, as a result, helps alleviate the vanishing gradient problem, allowing for more complex deep learning architectures to be trained effectively [22].

Dropout Layers

Dropout is a regularization technique used in deep learning to mitigate overfitting [46]. Dropout layers are inserted into the network architecture, randomly dropping out a fraction of the neuron activations during training. By disabling random fractions of neurons, dropout layers introduce noise and force the network to learn more generalizable features. This prevents individual neurons from overfitting to a specific input.



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * tanh(C_t)$$

Figure 18: *On the left, a diagram showing the LSTM network architecture with the computational flow [8]. On the right, the formulas used for calculating each set in the computation flow [55]. The σ symbol represents the sigmoid activation function.*

Seen differently, dropout layers create an ensemble of multiple sub-networks with shared weights, each contributing to the final output prediction. During the testing/inference process, dropout layers are not used, with all neurons participating in the final inference. Dropout layers have shown success in a variety of deep learning contexts, such as image classification or natural language processing, and continue to be frequently employed today [46].

Training a Neural Network

Training a neural network involves the optimization of its training parameters (weights and biases), which are initialized with random values. This optimization process aims to minimize the output predictions towards the ground truth using a loss function. Traditional optimization methods, such as gradient descent, are commonly used to minimize the loss function, by adjusting the parameters based on the gradients of the loss function with respect to the parameters of each layer. To render the training process efficient, backpropagation is utilized, which makes of the chain rule to calculate the gradients by propagating the errors from the output layer back to the input layer. Techniques such as Stochastic Gradient Descent (SGD) or Adam are commonly used optimization algorithms in deep learning which leverage the backpropagation calculations to modify the trainable network parameters.

Long Short-Term Memory Neural Networks

Long Short-Term Memory neural networks (LSTMs) are a type of Recurrent Neural Networks (RNNs) capable of learning long-term dependencies in sequential data [20]. LSTMs have shown to be particularly effective in process and modelling sequences, such as natural language processing [48] and time-series forecasting [15].

The key innovation of the LSTM architecture lies in the memory cell [20], which allows the network to selectively retain or forget information over time. Figure 18 shows the architecture of an LSTM network. The LSTM network consists of three gates, the input gate I_t, the forget gate F_t and the output gate O_t. The network additionally uses a memory cell input C_t to store previous information for sequential data. The forget gate determines how much information is kept from the previous output. The input gate controls how much new information should be added to the memory cell. It specifically determines the extent to which the candidate memory cell $\tilde{C}_t$ should be integrated with the previous memory cell. Finally, the output gate determines how much information from the memory cell should be passed as the current hidden layer output of the model.
Generally, the memory cell serves as a memory bank which retains information across an entire sequence, while the output gate derives a final output using the memory cell, allowing the network to preserve and use long-term dependency information.