

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Preservation of Patient Level Privacy: Federated Classification and Calibration Models

Permalink

<https://escholarship.org/uc/item/9dr8s83t>

Author

Huang, Yingxiang

Publication Date

2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Preservation of Patient Level Privacy: Federated Classification and Calibration Models

A Dissertation submitted in partial satisfaction of the requirements for the degree of
Doctor of Philosophy

in

Bioinformatics and Systems Biology
with a Specialization in
Biomedical Informatics

by

Yingxiang Huang

Committee in charge:

Professor Lucila Ohno-Machado, Chair
Professor Rob El-Kareh
Professor Terry Gaasterland
Professor Mike Hogarth
Professor Xiaoqian Jiang
Professor Lawrence Saul

2020

Copyright

Yingxiang Huang, 2020

All rights reserved.

The Dissertation of Yingxiang Huang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2020

Table of Contents

Signature Page.....	iii
Table of Contents	iv
List of Figures.....	vii
List of Tables.....	ix
Acknowledgements	x
Vitae	xii
Abstract of Dissertation.....	xiii
1. Introduction.....	1
1.1 Challenges for Data-Driven Decision Support	2
1.2 Federated Learning.....	4
1.3 Model Assessment	5
1.4 Dissertation Organization.....	6
2. Federated Model through Contextual Representation	7
2.1 Overview.....	7
2.2 Challenges in Contextual Embeddings.....	7
2.3 Limitations of Federated Models Built on Partitioned Data.....	10
2.4 Word Embedding.....	11
2.5 Medical Events Embedding	13
2.6 Procrustes Harmonization	15
2.7 Patient Diagnosis Projection Similarity.....	20
2.8 Structured Data	23

2.8.1 Incomplete Information	24
2.8.2 Split Patient History	26
2.9 Unstructured Data	28
2.9.1 Patient Similarity Prediction Model.....	29
2.10 Limitations and Future Directions	32
2.11 Conclusion	34
3. Calibration Measurements and Calibration Models.....	36
3.1 Overview.....	36
3.2 Introduction.....	36
3.3 Ranking Patients vs. Estimating Individual Risk	39
3.4 Simulated Data	43
3.5 Measuring Calibration	44
3.5.1 Brier Score and Spiegelhalter's Z-Test	45
3.5.2 Average Absolute Error	47
3.5.3 The Hosmer-Lemeshow Test	47
3.5.4 Reliability Diagram.....	52
3.5.5 Expected Calibration Error and Maximum Calibration Error.....	54
3.5.6 Cox Intercept and Slope.....	55
3.5.7 Integrated Calibration Index.....	56
3.6 Calibration Models	57
3.6.1 Platt Scaling	57
3.6.2 Isotonic Regression.....	58
3.6.3 Bayesian Binning Quantiles	59

3.7 Real Clinical Data	65
3.8 Discussion	71
3.9 Conclusion	74
4. Calibration Measurement and Calibration Model in a Federated Manner	76
4.1 Overview.....	76
4.2 Calibration Models	77
4.3 Methodology	78
4.4 Secure Multiparty Computation (SMC) Sorting.....	86
4.5 Secure Multiparty Computation Isotonic Regression.....	92
4.6 Smooth Isotonic Regression	96
4.7 Communication complexity.....	100
4.8 Measuring Calibration	100
4.9 Federated Calibration Model Experiments.....	104
4.10 Discussion	108
5. Final Remarks	110
6. Bibliography.....	114

List of Figures

Figure 2.1 Horizontally-partitioned vs. Vertically-partitioned Data	9
Figure 2.2 Patient History	14
Figure 2.3 Embedding Representation	15
Figure 2.4 Contextual Embedding Vectors	17
Figure 2.5 Procrustes Alignment Example	20
Figure 2.6 Diagnosis Prediction	22
Figure 2.7 Harmonization of Embeddings	25
Figure 2.8 Split Patient Record Results	28
Figure 2.9 Split Patient – Unstructured Data	31
Figure 3.1. Illustration of discordant pairs, ties, and ROC Curve	40
Figure 3.2 Grouping Methods for H-L C and H-L H Statistics	49
Figure 3.3 Reliability Diagrams of Test Set Estimates Produced by Logistic Regression (LR) and Support Vector Machine (SVM) Models	53
Figure 3.4 Calibration Models Functions	59
Figure 3.5 Reliability Diagrams of Test Set Estimates Produced by Support Vector Machine (SVM) Models After application with Platt scaling, Isotonic Regression, or Bayesian Binning Quantiles for Simulated Data	61
Figure 3.6 Reliability Diagrams of Test Set Estimates Produced by Logistic Regression (LR) Models After Application of Platt Scaling, Isotonic Regression, or Bayesian Binning Quantiles for Simulated Data	63
Figure 3.7. Reliability diagrams of test set estimates produced by Logistic Regression (LR) and Support Vector Machine (SVM) models for NIS dataset	68
Figure 3.8 Reliability diagrams of test set estimates produced by Logistic Regression (LR) models recalibrated with Platt scaling, Isotonic regression, or BBQ for NIS dataset	69

Figure 3.9. Reliability diagrams of test set estimates produced by Support Vector Machine (SVM) model recalibrated with Platt scaling, Isotonic regression, or BBQ, grouped for the H-L C statistics and the H-L H statistics for NIS dataset	70
Figure 4.1 Example of Isotonic and Smooth Isotonic Functions.....	79
Figure 4.2 Process for Deriving the Isotonic Function	81
Figure 4.3 Lagrange Multipliers Derivation	85
Figure 4.4 The First Iteration of Federated Radix Sorting.....	88
Figure 4.5 The Second Iteration of Federated Radix Sorting.....	91
Figure 4.6 The Third Iteration of Federated Radix Sorting	92
Figure 4.7 Initial Step of Federated Isotonic Regression	94
Figure 4.8 Finding the least Lagrange Multipliers	95
Figure 4.9 Smooth Isotonic Regression	97
Figure 4.10 Distributed Calibration Measurements	104

List of Tables

Table 3.1 Average Estimates and Observed Outcomes.....	43
Table 3.2. Areas Under the ROC Curve (AUROC), Brier scores, and Spiegelhalter’s Z-test statistics for Logistic Regression (LR) and Support Vector Machine (SVM) models.	46
Table 3.3. Average Absolute Error for Logistic Regression (LR) and Support Vector Machine (SVM) models.	47
Table 3.4. Calibration results measured with the Hosmer-Lemeshow (H-L) test statistics and p-values for Logistic Regression (LR) and Support Vector Machine (SVM) models.	52
Table 3.5. Calibration results measured with the Expected Calibration Error and Maximum Calibration Error for Logistic Regression (LR) and Support Vector Machine (SVM) models.	55
Table 3.6. Calibration results measured with the Cox’s Slope and Cox’s Intercept for Logistic Regression (LR) and Support Vector Machine (SVM) models.	56
Table 3.7. Calibration results measured with the Integrated Calibration Index for Logistic Regression (LR) and Support Vector Machine (SVM) models.	56
Table 3.8 Discrimination and Calibration Results of the Logistic Regression (LR) and Support Vector Machine (SVM) Models Applied to the Test Set.	64
Table 3.9 Summary of Packages in R Used to Evaluate Calibration and to apply calibration models on Estimates.	65
Table 3.10 Discrimination and Calibration Results of the Logistic Regression (LR) and Support Vector Machine (SVM) Models Applied to the NIS Test Dataset.	67
Table 3.11 Summary of Advantages and Disadvantages of Calibration Measurement Methods Presented in this Chapter	73
Table 4.1 Results for the Simulated Dataset.	107
Table 4.2 Results for the NIS Mortality Dataset.	107
Table 4.3 Results for the ACS NSQIP colectomy Readmission Data set.....	108

Acknowledgements

I would like to thank a great number of people for helping me get to this point in my academic career, as it would not have been possible without them.

First, I would like to thank my two advisors, Professor Lucila Ohno-Machado and Professor Xiaoqian Jiang, for giving me the opportunities to work with them over the past years. It was a continuous learning experience under their wings, preparing my analytical mind for the upcoming challenges in my research career. Dr. Jiang allowed me to flourish and explore under his guidance, even though I had little experience in computer science and in the field of biomedical informatics in general when I first joined the program. He would let me determine the direction of research I would like to take, but still, gently steer me down the path that would maximize the chance of success. It was a pleasure discussing what experiments to do and very educational to understand his thought process. Dr. Ohno-Machado was instrumental in continuing my education in machine learning, especially in areas specific to biomedical informatics. Her patience with all my shortcomings in research, communication, and soft skills is incredible. She is constantly pushing me to raise my standard to the level of a true professional scientist.

Next, I would like to thank my committee members, Dr. Robert El-Kareh, Dr. Mike Hogarth, Dr. Lawrence Saul, and Dr. Terry Gaasterland, for their support and encouragement. They have broadened my view and encouraged me to approach my research from different perspectives. Their input has been invaluable in shaping my projects.

In addition, I would like to thank my colleagues and friends who have listened to my problems and encouraged my research throughout my Ph.D.: Wael Farhan, Dr. Junghye Lee, Dr. Tim Tsung-Ting Kuo, Jihoon Kim, Brian Tsui, Arjun Chandrasekhar, and Michelle Lowe. They have

been invaluable in my research discussion and mental wellbeing. Without them, I would have no one to share defeats and triumphs. Without them, I would have no one to share the fun of graduate school life.

Finally, I would like to thank my family. My parents have always been there to support me every step of the way, making life as easy as possible for me, and helping me in any way they can. Most of all, thanks to Fanny Du for being there every day to laugh with me, annoy me, be silly with me, and share life with me.

Chapter 2, in part, is a reprint of the material as it appears in *Privacy-Preserving Predictive Modeling: Harmonization of Contextual Embeddings From Different Sources* by Huang, Yingxiang; Lee Junghye; Wang, Shuang; Sun, Jimeng; Liu, Hongfang; Jiang, Xiaoqian, JMIR Medical Informatics, 2018. I was the primary investigator and author of this paper.

Chapter 3, in part, has been accepted for publication of the material as it may appear in *A Tutorial on Calibration Measurements and Calibration Models for Clinical Prediction Models* by Huang, Yingxiang; Li, Wentao; Macheret, Fima; Gabriel, Rodney; Jiang, Xiaoqian; Ohno-Machado, Lucila, Journal of the American Medical Informatics Association, 2020. I was the primary investigator and author of this paper.

Chapter 4, in part, is in preparation for submission as *Isotonic Regression Calibration through a Federated Network of Private Localized Patient Data* by Huang, Yingxiang; Gabriel, Rodney; Jiang, Xiaoqian; Ohno-Machado, Lucila, 2020. I was the primary investigator and author of this paper.

Vitae

2012 University of California, Berkeley

Bachelor of Art, Molecular Cellular Biology

2020 University of California, San Diego

Doctor of Philosophy, Bioinformatics and Systems Biology with a Specialization in

Biomedical Informatics

Huang, Y., Jiang, X., Gabriel, R., Ohno-Machado, L. (2020) Isotonic Regression Calibration through Federated Network of Private Localized Patient Data, in preparation for submission.

Huang, Y., Li, W., Macheret, J., Gabriel, R., Ohno-Machado, L. (2020) A Tutorial on Calibration Measurements and Calibration Models for Clinical Prediction Models. Journal of the American Medical Informatics Association, accepted for publication.

Huang, Y., Lee, J., Wang, S., Sun, J., Liu, H., and Jiang, X. (2018). Privacy-Preserving Predictive Modeling: Harmonization of Contextual Embeddings From Different Sources. JMIR Med. Inform. 6, e33.

Farhan, W., Wang, Z., **Huang, Y.**, Wang, S., Wang, F., and Jiang, X. (2016). A Predictive Model for Medical Events Based on Contextual Embedding of temporal sequences. JMIR Med. Inform. 4, e39.

Hefzi, H., Ang, K.S., Hanscho, M., Bordbar, A., Ruckerbauer, D., Lakshmanan, M., Orellana, C.A., Baycin-Hizal, D., **Huang, Y.**, Ley, D., et al. (2016). A Consensus Genome-scale Reconstruction of Chinese Hamster Ovary Cell Metabolism. Cell Syst. 3, 434-443.e8.

Fields of Study

Major field: Bioinformatics and Systems Biology with a Specialization in Biomedical Informatics

Abstract of Dissertation

Preservation of Patient Level Privacy: Federated Classification and Calibration Models

By

Yingxiang Huang

Doctor of Philosophy in Bioinformatics and Systems Biology
with a Specialization in
Biomedical Informatics

University of California San Diego, 2020

Professor Lucila Ohno-Machado, Chair

With the launching of the Precision Medicine Initiative in the United States, by the National Institute of Health, and the emergence of a large volume of electronic health records,

there are many opportunities to improve clinical decision support systems. A large number of samples are needed to build predictive models that have adequate discrimination and calibration. However, protecting patient privacy is also an important issue. Patient data are typically protected in localized silos, and consolidation of datasets from different healthcare systems is difficult.

Federated learning allows the training of a global model by amassing intermediate calculations from localized medical systems. The knowledge learned from the data can be transferred and aggregated to achieve better performance than the one achieved by individual local models. Federated learning may help build better models, providing more accurate predictions. There are two types of measures to assess how well a model performs: discrimination and calibration. While most papers report discrimination measures, calibration has often been neglected but it is a critical metric for evaluation. In this dissertation, I show a novel way to build classifiers and calibration models in a federated manner. I also show how I can evaluate and improve model calibration in this manner. Federated modeling enables the accumulation of knowledge and information that are otherwise locked behind local medical systems.

1. Introduction

Data-driven high-quality care is the ultimate goal for healthcare providers. However, just 10 years ago, the lack of efficient medical record systems and archaic government policies seriously hampered the realization of such an ideal. To remedy the situation, as part of the American Reinvestment & Recovery Act in 2009, the federal agency Centers of Medicare & Medicaid Services (CMS) established the Medicare and Medicaid Electronic Health Record Incentive Programs, commonly known as “Meaningful Use”, to encourage healthcare professionals to adopt, implement, upgrade, and demonstrate meaningful use of certified Electronic Health Record (EHR) technology.¹ By 2015, more than eight out of ten doctors across the country had adopted electronic healthcare records,² establishing a more efficient recording system and laying down the foundation for the widespread development of clinical decision support systems (CDSS). Such prevalence of EHR systems also prompted a redefinition of “meaningful use”: from the mere adoption of technology to ensure that the widely adopted technology improved the quality and cost of healthcare. In accordance to the new direction, Meaningful Use became one of the subsidiaries of Medicare Access and CHIP Reauthorization Act (MACRA) passed by Congress in 2015.³ Under MACRA, effective 2019, physicians are no longer reimbursed for medical services rendered, but for the outcomes, incentivizing high quality of care over high service volume. As part of this more quality-oriented policy, Meaningful Use implemented new guidelines that no longer reward just the adoption of EHR, but are directly linked to the merit of the physicians and the standard of care provided.

With the electronic infrastructure and policies structured to support a quality-focused healthcare experience, more and more CDSS are implemented in the EHR to elevate the quality

of care. As stated by the Office of the National Coordinator for Health IT, CDSS “provides clinicians, staff, patients or other individuals with knowledge and person-specific information, intelligently filtered or presented at appropriate times, to enhance health and health care.” One way for the information that is “person-specific” and “intelligently filtered” to manifest itself is through the use of predictive analytics. By learning historical records and how events occurred, predictive analytics provides physicians with the foresight and opportunity, for example, to take precautions in certain patients who may exhibit patterns that have led to complications in other patients.

1.1 Challenges for Data-Driven Decision Support

Predictive analytics works optimally with large and complete data sets. Studies such as those that try to find relevant biomarkers, distinguish patterns for rare diseases, discover the combined effects of multiple genetic variants or epistasis, or research unique phenotypes of diseases that only appear in certain demographics, have shown that patient characterization on a multidimensional level may lead to far more intricate diagnostic and prognostic groupings of populations than generally used today.^{4,5,6} They all illustrate an intuitive principle: predictive models, like the human brain, learn more the more samples they see. In order for predictive analytics to be meaningful to physicians (instead of being just another pop up on their EHR window, to be ignored immediately), predictive analytics needs to be accurate, and having large and complete data to train a generalizable model is the key.

To gather a large number of patient samples, the best way is for medical systems to share data. However, some data cannot be transferred outside of their designated healthcare systems, there can be enormous logistical complexity and cost before data can be transferred. For

example, in order for an individual's data to leave the European Union (EU), the person whose data it belongs to has to be informed and opt-in this data sharing activity.⁷ Procedures also need to be in place to ensure that the data are tracked, secured and protected. Some healthcare systems have very strict guidelines for transferring data outside their enclaves (e.g., the United States Department of Veterans Affairs (VA)). Research studies such as the AllofUs Research Program, the Strongheart Study, and many others also need to ensure data stays where participants have consented to. The data in these programs are not easily movable. The VA, which provides near-comprehensive healthcare services to eligible military veterans at VA medical centers and outpatient clinics, has to follow the Privacy Act of 1974 that provides a number of protections for veteran's personal information.⁸ The policy includes how information is collected, used, disclosed, stored, and disposed of, some of which may prevent the transfer of data for model building. Similarly, the AllofUs Research Program and Strongheart study both have limits and regulations as to who can handle the data and how they can access the data,^{9,10} making aggregation of data difficult.

Patient-level and institutional level data leakages cause concern. Transferring, creating copies, and storing data outside of a medical system expose data to situations of potential leakage. This is detrimental to both the patients and the medical systems. At the patient level, leakage of sensitive patient data could lead to discrimination or stigmatization. To curtail problems over leakage of sensitive information, there have been some efforts to share de-identified data in compliance with the Health Insurance Portability and Accountability Act (HIPAA).¹¹ However, such efforts are either costly, error-prone, or ineffective.¹² For example, human-based de-identification efforts on the Medical Information Mart for Intensive Care III

(MIMIC-III) dataset costed over 5,000 hours and US\$500,000,¹³ for about 50,000 patient visits and 100 million words, with error recall ranging from 0.63 to 0.94.^{14,15} Additionally, machine-assisted de-identification shows varying results, from time savings of 13.85–21.5% to no improvement in either quality or time saved.¹⁶ Machine learning algorithm-based automated de-identification can be useful, but state-of-the-art deep learning-based de-identification models for unstructured data are still incapable of reaching an adequate level of privacy protection.^{15,17} Continued research is needed in this area. Furthermore, such de-identification efforts do not protect institutional privacy. At the institutional level, sharing sensitive information could put a medical system at a disadvantage with respect to its competitors. This sensitive information could include mortality rates, complication rates, healthcare-associated infection rates, unplanned readmissions, etc. While these data may be aggregated at the institution's population level (and sharing them does not violate patient-level privacy), medical systems may be reluctant to share such data due to concerns over their reputation. Therefore, it is very attractive to have the ability to somehow create a good predictive model while keeping data private.

1.2 Federated Learning

With concerns over privacy and regulation that can prevent data sharing, I look toward federated learning as a potential solution. Federated learning involves training a global model by amassing pertinent knowledge from localized medical systems while protecting data privacy. The knowledge learned from the data can be transferred and aggregated to achieve better performance for global models than the performance of any local models. Federated learning can involve sharing intermediate calculations of a model that can be combined to form a final

solution.¹⁸ Federated learning should decrease the bias of models that are built from local data, providing more accurate predictions.

1.3 Model Assessment

As important as building a federated model is to be able to evaluate it in a federated manner. Assessing the performance of a federated predictive model is typically done by measuring discrimination and calibration. Given a model's probability estimates of a medical outcome for a population of patients, discrimination measures whether patients who have developed the outcome receive higher probability estimates than patients who did not. With each patient receiving a probability estimate for having the outcome, various thresholds can be set to discriminate patients who have the outcome and those who do not. Given the actual outcomes of patients and their estimates, one can measure the true positive, false positive, true negative, and false-negative rates, which can be collectively summarized by the measure of the area under the receiver operating characteristic curve (AUROC). The higher the AUROC, the better the model can accurately discriminate whether a patient will or will not have an outcome. On the other hand, calibration measures whether there is an agreement between estimates and the actual probabilities of the outcome. Accurately calibrated models are important in prognostic settings because predicting the right probability of developing a medical outcome (e.g., a disease, a complication, readmission to the hospital) helps physicians weigh the risks and benefits of their actions. However, measuring calibration is not straightforward because a patient's true probability of an outcome is not readily available. I can say that the true probability is 10% if 10 out of 100 patients just like this patient have a certain outcome, but it is difficult to define who is "just like" who. Instead, some proxies are used to group patients according to their estimates.

This strategy is used in the Hosmer-Lemeshow test (HL-test). In the HL-test, patients who are “just like” each other are grouped by similar model estimates. For example, if a perfectly calibrated model predicted that there are 100 patients with an estimated probability = 0.1 of developing diabetes within the next 10 years, I should observe that 10 patients actually develop diabetes. A detailed analysis of calibration, how to measure it, and how to apply calibration models on predictive models are found in later Chapters. Ultimately, a predictive model that exhibits high performance in both discrimination and calibration can be of high value in a CDSS.

1.4 Dissertation Organization

I developed *a new federated learning method* and an approach to utilize a model trained in one population for another population, *even when the respective data sets do not have the exact same variables*. I will demonstrate that my new federated model can outperform local models. I also present a new method to evaluate and improve a federated model’s calibration in a privacy-preserving manner using a novel federated evaluation method.

This dissertation is organized as follows: Chapter 2 describes a new approach to combine information from different local medical systems and create a classifier in a federated manner, even when not all variables are the same across the systems; Chapter 3 explains current calibration measurements, common calibration models, and their limitations; Chapter 4 extends methods from Chapter 3 to create calibration models and measure calibration in a federated manner; Chapter 5 concludes the dissertation.

2. Federated Model through Contextual Representation

2.1 Overview

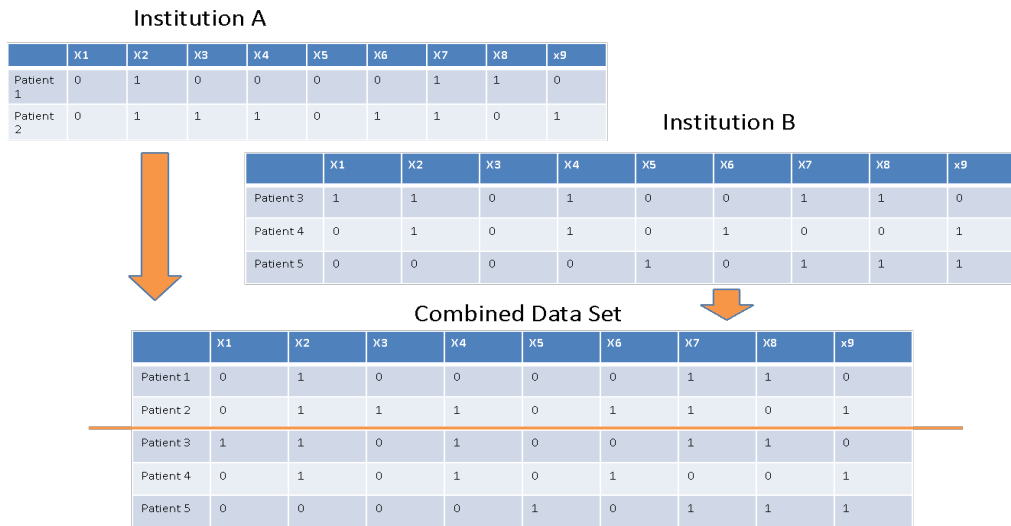
I propose a methodology to combine different local contextual embedding models into a global model. Contextual embedding models, as the name suggests, embed context into a representation of the data. In our case, this representation consists of word vectors. In short, I use Word2Vec³⁷ to generate contextual embeddings from each data source, which have different sets of words (or “variables”), and the Procrustes method to align different vector models into one common space. Previously unseen observations are then projected into this space, and their position allows us to determine the values for its nearest neighbors’ features of interest (e.g., diagnoses). Under different experimental scenarios, I confirmed that the global model built from combined local models achieves more accurate predictions than local models. Such an aggregation of local models using our unique harmonization technique serves as a proxy for a global model.

2.2 Challenges in Contextual Embeddings

As mentioned in the Introduction Chapter, creating global models that can outperform local models is attractive to medical systems that wish to maintain their patient and institutional level privacy. Sharing model parameters and contextual embeddings, instead of sharing individual observations, does not require a significant loss in data utility. Our vision to address this challenge is to create a federated clinical analysis framework through the aggregation of local representations. Related studies have been published focusing on not only simple analyses such as database queries with very specific inclusion/exclusion criteria, but also sophisticated

algorithms for prediction analysis including logistic regression,^{18–21} support vector machine (SVM),^{22,23} *k*-nearest neighborhood,²⁴ Bayesian inference,²⁵ Cox regression,^{26,27} and tensor factorization.²⁸ Most studies involve restrictive assumptions originating from the requirement that data should be integrated in a matrix format. There is a requirement for common features in so-called “horizontally-partitioned” data, or for common patients in so-called “vertically-partitioned data” (**Figure 2.1**). Most algorithms operate on either horizontally- or vertically-partitioned data, but not both.

a)



b)

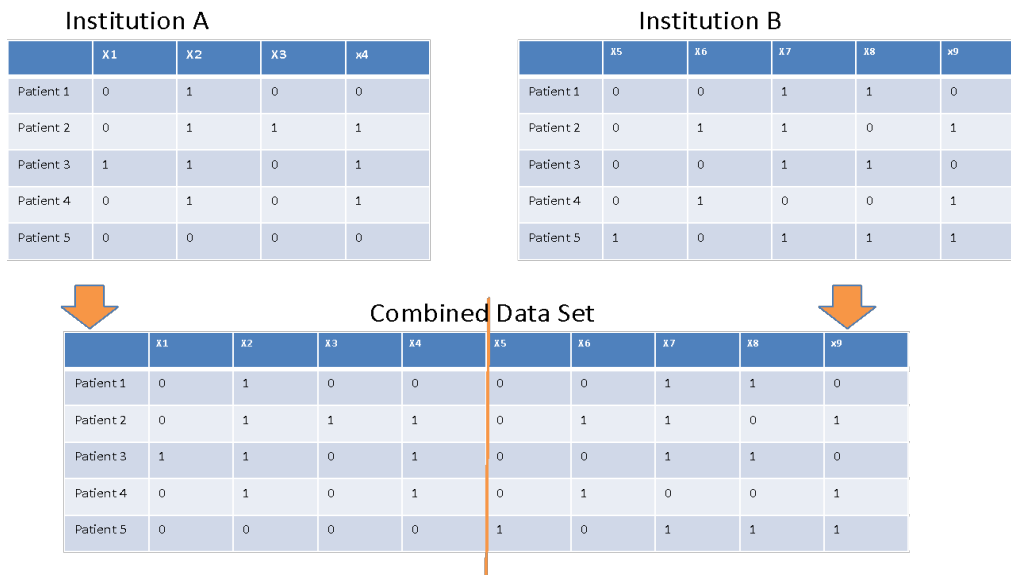


Figure 2.1 Horizontally-partitioned vs. Vertically-partitioned Data

a) Federated models assume horizontally-partitioned data (e.g., patient data have common features in different medical institutions). b) Federated models assume vertically-partitioned data (e.g., patients have portions of their data in different medical institutions).

2.3 Limitations of Federated Models Built on Partitioned Data

There have been ongoing efforts to develop model building methods assuming horizontally-partitioned data structures that do not transfer individual-level data to build models in a federated manner. There are methods that use summary scores that allow researchers to perform matched or stratified analysis without having to share highly sensitive patient-level data.^{29–31} On the other hand, federated models, which strive at being computationally equivalent to pooled data regression, are more useful.³² Continued research has shown that a federated model that combines summary scores may achieve analytical flexibility and privacy protection.²⁰ All the methods mentioned can be calculated sequentially or in batches. Training a model sequentially updates a model as new data is added without having to train the whole model,³³ while training a model in batches mean models are trained separately and the final result strives to achieve the exact same model as a ‘traditional’ regression model.¹⁸ For all the horizontally-partitioned methods mentioned, having all common features an unreasonable assumption, as different medical systems may record different attributes due to different specialties. Also, different medical systems might have their own annotation systems for the same medical events.

Besides horizontally partitioned data regression, vertically partitioned data regression is also an area of interest. Expectation-Maximization (EM) algorithms assume the data can be reasonably described by either a multivariate normal or multinomial distribution.³⁴ Under such an assumption, the EM algorithm requires sharing only relevant statistics, which for the multivariate normal are sums and inner-products of the observed data values. These quantities can then be shared without sharing data values. Other methods include secure matrix multiplications, which are used by pairs of owners to compute off-diagonal blocks of the full data

covariance matrix.³⁵ Most recently, VERTICAL Grid Logistic regression (VERTIGO) solves the vertically partitioned data regression problem by calculating the first and second derivatives of the log-likelihood function followed by an iterative maximization procedure.³⁶ Nevertheless, for vertically-partitioned data, having a large number of common patient records in different institutions may be unrealistic.

2.4 Word Embedding

I wanted to create federated models that do not need to assume horizontally-partitioned or vertically-partitioned data format. This way, the features did not need to be exactly the same across institutions, since projecting them into a lower dimension places the same types of features near each other, as shown in Section 2.5. I utilized a popular word embedding technique to learn representations of medical events in each local medical system and combine the local embeddings. Unique information learned in each local medical system is represented in each embedding (or vector) and the accumulation of embeddings can increase the number of features utilized.

Deriving accurate word representations with contextual information is an endeavor extensively explored in the field of natural language processing. A popular program developed by Google called Word2Vec has shown great promise in contextual word embedding.³⁷ I used Word2Vec to contextually represent medical events. Training a text corpus and mapping each word into a vector space can represent words in such a way that the more related the words are, the closer their vectors are in terms of cosine distance. Unlike representations in previous models, where words had no connection to other words, this model trains each word using the context of its surrounding words with a two-layer neural network on a Skipgram architecture.³⁷

A Skipgram architecture takes every word in large corpora and also takes, one by one, the words that surround it within a defined 'window', and feed them into a neural network that, after training, will predict the probability for each word to actually appear in the window around the word of interest. The neural network consists of essentially a three-layer network with input, projection, and output layers. Providing the input as a sequence of 1-of- M coding, where M is the vocabulary size, Word2Vec is capable of projecting words into a lower dimensional space while extracting their context. Under the assumption that words that appear in a similar context (e.g. synonyms or prepositions) would have similar meanings and functions to a sentence, Word2Vec provides a representation that is context-aware.

Unlike one-hot representation, which may not be able to make distinctions between related concepts such as 'congestive heart failure' and 'myocardial infarction', contextual embeddings produce a closer distance for these two concepts than other unrelated concepts (i.e. kidney failure). Furthermore, in terms of prescriptions, previous one-hot representation models would perceive prescriptions for Warfarin labeled as "WARF10" and "WARF75" as two different medications, but human experts could tell they are just different dosages. Therefore, if one patient was taking "WARF10" while the other "WARF75", previous models unnecessarily represented the patients more dissimilarly than in reality. Consider another example. If there are three patients where one has diabetes and kidney failure, the second one has diabetes and neuropathy, and the third has kidney failure and neuropathy, previous systems might find all three patients to be similar since they each share an event with the other two. However, a context-aware model should find the first and second patients to be more similar since they have diabetes and each has a complication related to diabetes.

Models that utilize word embeddings have shown higher prediction performance than previous models.^{38,39} Furthermore, research into named entity recognition,⁴⁰ abbreviation expansion,⁴¹ unplanned medical system readmission prediction,⁴² and disease risk estimation that incorporates long/short term dependencies in the EHR⁴³ are examples of areas with improved results given the application of word embedding.⁴⁴ As more and more deep learning models dive into the realm of clinical text (instead of just using structured data to make predictions), word embeddings are becoming more used. My strategy was to create word embeddings separately at each institution, then “fuse” these embeddings (i.e., aligning the embeddings by translating and/or rotating the resulting vectors based on the common features across the institutions).

2.5 Medical Events Embedding

To train the model, I extracted data from a publicly available medical database called MIMIC (Medical Information Mart for Intensive Care),¹³ which contains data from ~45,000 patients who were admitted to the Intensive Care Unit (ICU) at a large tertiary care hospital between 2001 and 2012. The data include demographics, admissions, lab results, prescriptions, procedures, and ICD-9 discharge diagnoses. For each patient, I ordered medical events in chronological order, from oldest to newest, as shown in **Figure 2.2**. To prevent conflicting identifiers between events, I added prefixes for each category: ‘p_’, ‘l_’, ‘s_’, ‘c_’, and ‘d_’ for prescriptions, lab tests, symptoms, conditions, and diagnoses, respectively. ICD-9 codes were preprocessed to level 3, which is moderately specific (diagnoses in **Table 2.1** are level 3 descriptions). Ninety percent of the data were used as the training set, and 10% were withheld as the test set. Each sequence of patient events acted as a sentence that Word2Vec could learn

from. The result was the embedding of all medical events onto a hyperdimensional space with 350 dimensions.

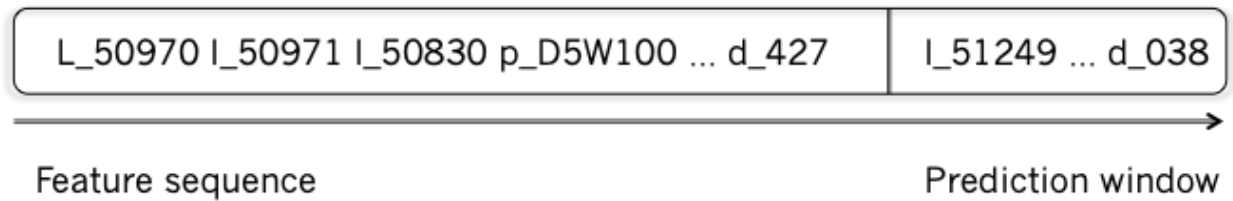


Figure 2.2 Patient History

A patient’s chronological history of events from the test set, with prefix corresponding to the event’s category. The model predicts diagnoses in the prediction window.

The learned medical event vectors were mapped onto the hyperdimensional space, and are represented in this two-dimensional plot in **Figure 2.3**. While a two-dimensional representation is very limited, I can still see that a large number of diagnoses (shown in black) were clustered together. Symptoms and conditions (green and yellow), which were also represented with ICD-9 codes and used in a similar context, are also clustered together. Looking for more evidence on contextual relationships, I calculated the cosine similarity between ‘diabetes’ and all the other medical events in the corpus. The most similar vector in terms of cosine similarity was c_V5867, or ‘long term use of insulin’ and ‘Acute infective polyneuritis’ (a complication of long-term diabetes), and the most similar prescriptions were ‘Insulin pump’ and ‘Insulin’.

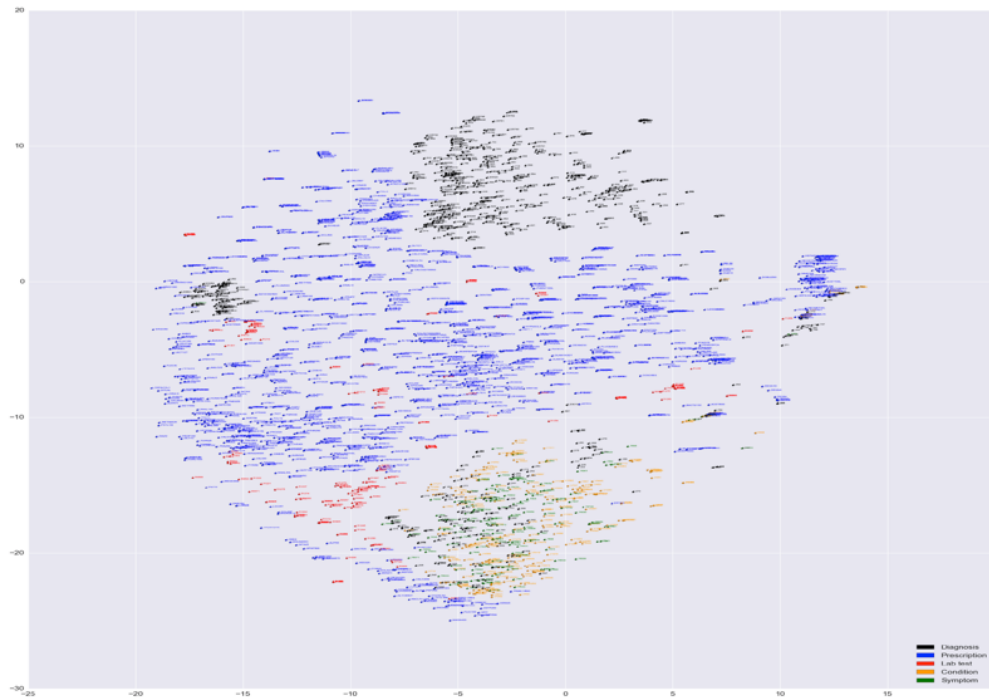


Figure 2.3 Embedding Representation

A two-dimensional representation of medical events projections. Black dots are diagnoses. Blue dots are medications. Red dots are lab tests. Yellow dots are conditions. Green dots are symptoms. I can see that all the diagnoses are clustered together. Conditions and symptoms are clustered together because they both use ICD9 codes.

2.6 Procrustes Harmonization

Existing contextual embedding models are often built from EHR data from a single institution. Each of these separate sites may contain information that other sites lack (i.e., different patients and different features). It would be ideal if a model could be built on raw data aggregated from different medical systems in order to compensate for the missing/sparse information each site may have. However, due to privacy and/or policy concerns or the inability

to move some datasets, such aggregation is often infeasible. To address these problems, I propose that each medical system builds its own contextual embedding model, after which no explicit patient-level information would remain in the acquired representations (i.e., in the embeddings). Then, medical systems can share their own embeddings or representations without violating patient privacy. I propose a methodology that harmonizes different contextual embeddings into a global embedding or representation model.

Given a cohort of patients and their history like the example given in **Figure 2.2**, I use Word2Vec to create continuous vectors for each medical event. An example is shown for structured data in **Figure 2.4** (The detailed information about the Figure will be described in the next section). For this dissertation, I choose the skip-gram model. The model requires two parameters, size and window, defining the dimensionality of the final vector representation and maximum distance for contextual consideration, respectively.^{38,45}

There is one limitation to contextual embedding techniques such as Word2Vec and GloVe.⁴⁶ Due to random sampling in the training process, the same dataset can result in embeddings of different orientations. This means that, even if embeddings trained from the same dataset are pooled together, the medical events or concepts in one embedding could have unreasonable relationships with events or concepts in the other embedding. Therefore, it is critical to find a way to 'align' these embeddings.

Clinical Pathways of Medical System 1

- p_ACET325 p_ACET650R p_AMBIS p_ASA325 p_ASAEC325 p_CALG11 p_CARA1 p_CLOP75 p_CLOP75 p_D5W250 p_D5W250 p_DOCU100 p_GLYC11 p_INHRIV p_KCL20PM p_KCLBASE2 p_KETO151 p_KETR301 p_MAGS11 p_MEPPE501 p_METO101 p_MIDA21 p_MORP41 p_NEOSI
- p_CITA20 p_COMP101 p_COMP101 p_DULO30 p_GABA300 p_HEPA51 p_INHRIV p_INHRIV p_KCL20/1000D5NS p_KCL20/1000D5NS p_KCL20/1000NS p_MAAL30L p_METO51 p_NACLFLUSH p_NS100
- I_50912 I_51221 I_51222 I_51248 I_51275 I_51279 p_ACET325 p_AMBIS p_AMLO25 p_ASAB1 p_ATOM40 p_COZAA50 p_DOCU100 p_HEPAPREMIX p_HEPBASE p_JSOS20 p_NACLFLUSH
- I_50893 I_50912 I_50931 I_50971 I_51003 I_51006 I_51221 I_51222 I_51237 I_51244 I_51248 I_51249 I_51256 I_51274 I_51277 I_51279 p_CALG11 p_CALG11 p_CALG11 p_D5W250 p_DDAV41 p_DDAV41

Clinical Pathways of Medical System 2

- p_VANCOBASE p_PAPA300 c_V4581 d_250 d_272 d_357 d_362 d_401 d_410 d_428 d_428 d_440 d_447 d_536 d_583 d_584 d_730 d_997 d_998
- p_ACET325 p_AMLO25 p_NS100 p_PANT401 p_SERT50 I_50862 I_51006 I_51221 I_51222 I_51244 I_51256 I_51279 p_AMLO25 p_HALD51 p_MAGS11 p_MICROK10 p_NS100 p_PANT40
- p_AMID200 p_ATOM10 p_CEFX1F p_COSY25I p_D5W250 p_D5W250 p_FRBD50 p_HEPA1005YR p_HEPA51 p_INHRIV p_LEV250 p_LEVO41 p_LEVO41 p_NS100 p_OXYCS p_PANT40 I_50882 I_50893 I_50902 I_50912 I_50931 I_50970 I_51003 I_51006 I_51144
- c_V1089 c_V4581 d_201 d_412 d_414 d_428 d_481 d_496 d_555
- I_50971 I_50983 I_51006 I_51221 I_51222 I_51244 I_51248 I_51249 I_51256 I_51265 I_51277 I_51279 I_51301 I_50806 I_50822 I_50910 p_ALPR25 p_ASAB1 p_D545NS1000 p_D545NS1000 p_DIVA500

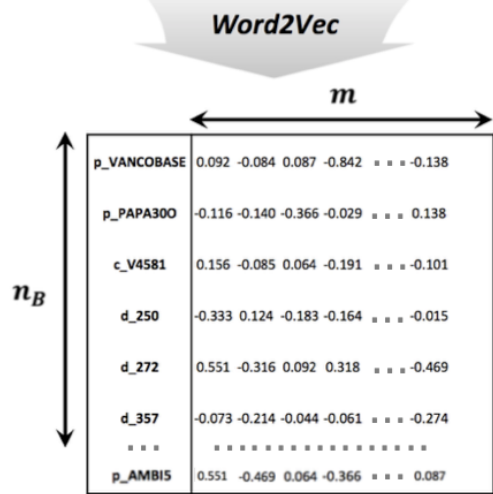
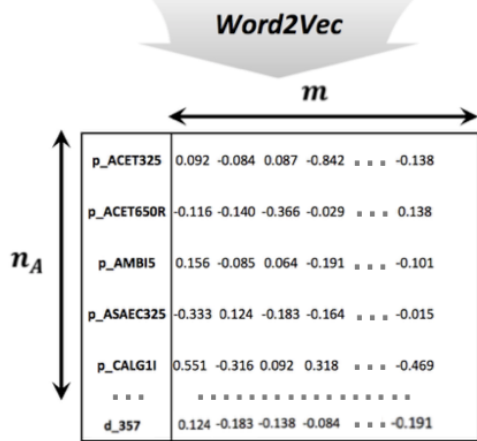


Figure 2.4 Contextual Embedding Vectors

Example of two contextual embeddings created from two medical systems.

In this section, I will go into more detail as to how the Procrustes method,⁴⁷ a method to align the embeddings, works. Using the Procrustes method to fuse different vector models into one common space requires a list of corresponding pairs. These are pairs of words, which are the same events but may or may not be labeled differently in different institutions. With most medical systems using standardized terminology system such as the International Classification of Diseases, Ninth Revision (ICD-9) and ICD-10 for billing purposes, I can reasonably identify a list of codes referring to the same events in different medical system sites to serve as our “anchor pairs” for alignment. Using these common events, I can derive a matrix that transforms one contextual embedding into the space of another.

Taking the two contextual embeddings in **Figure 2.4** as examples, the embeddings are $A \in R^{n_A \times m}$ and $B \in R^{n_B \times m}$, where n_A and n_B are the numbers of contextual embeddings from two medical systems respectively, and m is the dimensionality of the embeddings. With the corresponding anchor pairs X and Y , where $X \in A$ and $Y \in B$, I can solve for the orthogonal matrix Q and scalar k from the corresponding anchor pairs in the following Equation (2.1),

$$(2.1) \min_{Q,k} \|(X - 1_n \mu_x^T) - kQ(Y - 1_n \mu_Y^T)\|$$

where μ_x^T and μ_Y^T represent column-wise mean vectors of X and Y ; n is the number of corresponding anchor pairs. Q is solved as follows using singular value decomposition as shown in Equation (2.2),

$$(X - 1_n \mu_x^T) - kQ(Y - 1_n \mu_Y^T) = U \Sigma V^T$$

$$Q = UV^T$$

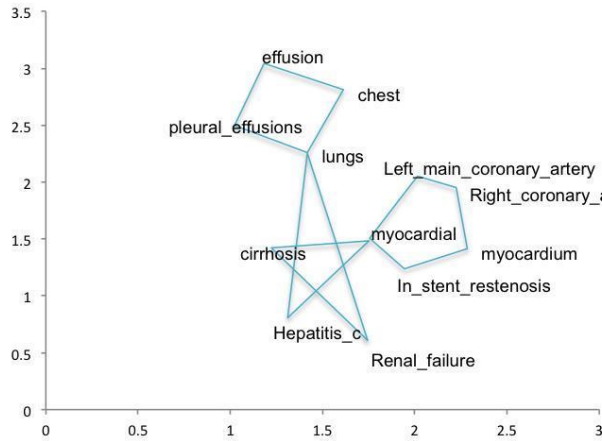
$$(2.2) k = \text{trace}(\Sigma) / \text{trace}(Y^T Y)$$

where U and V are the unitary matrices of the singular value decomposition, 1_n is an n -dimensional column vector whose elements are all 1's, k is a scalar, and Σ is the rectangular diagonal matrix with non-negative real numbers on the diagonal. Applying Q and the scalar k , I can solve for A^f and B^f , which are the harmonized vector representations of A and B , as shown in Equation (2.3):

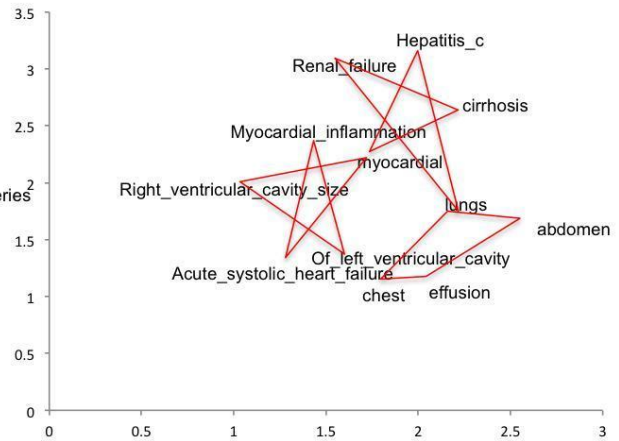
$$\begin{aligned}
A &\in R^{n_A \times m} \\
B &\in R^{n_B \times m} \\
A^f &= A - 1_{n_A} \mu_x^T \\
\text{(2.3)} \quad B^f &= kQ(B - 1_{n_B}) \mu_x^T
\end{aligned}$$

An example is illustrated in **Figure 2.5**. I can see that ‘Hepatitis_c’, ‘Cirrhosis’, ‘Lungs’, ‘Myocardial’, and ‘Renal_failure’ in (a) and (b) are common in both local models. Using them as anchor pairs to derive the orthogonal matrix, I can harmonize the two local representation models into a common one as shown in **Figure 2.5** (c).

a)



b)



c)

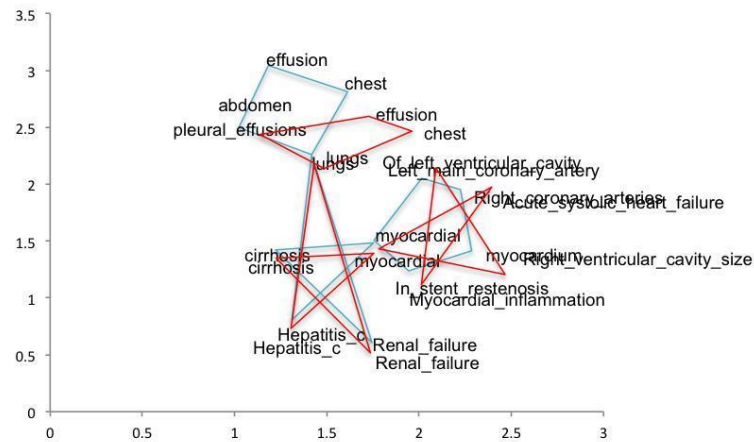


Figure 2.5 Procrustes Alignment Example

Example of Procrustes harmonization. (a) Local embeddings of Site 1. (b) Local embeddings of Site 2. (c) Combined embeddings of two sites after Procrustes harmonization.

2.7 Patient Diagnosis Projection Similarity

To predict the next likely diagnosis of a new patient for structured data experiments, I use the patient-diagnosis projection similarity (PDPS) method.³⁸ To calculate PDPS, I first create a

patient vector. In short, I normalize the summation of each vector representation of events in the clinical pathway of a patient, with each event vector multiplied by a time decay function (i.e., $e^{-\lambda t}$ with a time decay factor λ) (**Figure 2.6**). To calculate the probability of each diagnosis, I calculate the cosine similarity between the patient vector and a diagnosis vector in Equation (3.4).

$$(3.4) \hat{y}(S, d) = \text{cosine}(\vec{V}_d, \sum_{c \in S} \vec{V}_c e^{-\lambda t_c})$$

where λ is the time factor, t_c is the number of events before the last event of the clinical pathway, \vec{V}_d is the contextual vector representation of diagnosis d in the vector space, \vec{V}_c is the vector contextual representation of a medical event in the clinical pathway of a patient, S , and thus $\sum_{c \in S} \vec{V}_c e^{-\lambda t_c}$ is the patient vector.

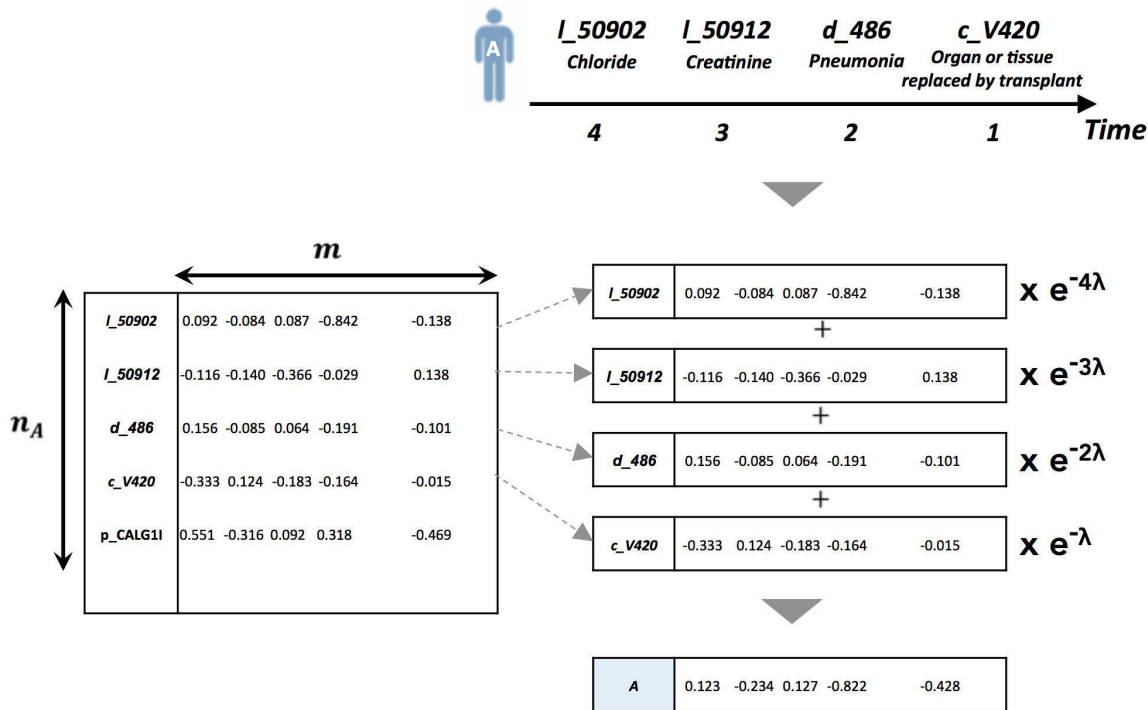


Figure 2.6 Diagnosis Prediction

Example of creating a patient vector from event-level vector representations. The patient vector is a linear combination of the event vectors, weighted by a time decay function ($e^{-\lambda t}$ with a time decay factor λ).

The data set I used is MIMIC-III.¹³ I had two sets of experiments. One was conducted on structured data, such as codes for diagnoses, prescriptions, lab tests, symptoms, and conditions. ICD-9 code was used for symptoms, diagnoses, and conditions. Custom local codes were used for lab tests and prescriptions. The other experiment was conducted on unstructured data (i.e., clinical notes), which will be discussed in later sections. For structured data, ICD-9 codes for diagnoses in MIMIC-III were generalized to level-3. For example, a patient with ‘*Diabetes with ketoacidosis, type I [juvenile type] uncontrolled*’ (250.13) was generalized to ‘*Diabetes mellitus*’ (250) by reducing all ICD-9 codes to 3 digits. Because our evaluation is based on the prediction accuracy given patients’ prior history, I excluded patients who only had one admission. I also

excluded rare medical events that had fewer occurrences than 50. In the end, I kept 5,639 patient records for this experiment. From these records, I constructed the temporal clinical pathway for both structured and unstructured data. Ten-fold cross-validation was implemented for all experiments, which randomly split the dataset into ten folds with equal sizes, using nine folds for training and one fold for testing.

In each replicate, to simulate two different medical system sites, I randomly divided the training patient records into two groups of patients; I call these '*local*' sites. Experiments done on all training patients were used as a benchmark for comparison; I call that model '*benchmark*.' From the training set, I created the '*benchmark*' contextual embedding model using all patient records and the two '*local*' contextual embedding models each using half of all patient records. The size of vector and window parameters used to learn word embeddings for structured data were 350 and 30, respectively. For the unstructured data, the parameters were 350 and 100, respectively. These two '*local*' embeddings were harmonized into a common model using the Procrustes method.

2.8 Structured Data

For structured data, the harmonization of the two '*local*' embeddings required common terms to serve as corresponding anchor pairs. There were a total of 2,713 total unique terms between the two sites, of which there were 2,538 common terms. To create more realistic simulations, I used different percentages of all possible common terms as corresponding pairs for different experimental scenarios. The rest of the pairs were artificially labeled differently, where the words not in a pair from one site were appended with the suffix '*_m1*', and the words not in a pair from the other site were appended with '*_m2*'. This was done to simulate that

different medical systems have their own terminology, and maybe only a fraction of all of their medical events codes are in common.

For all scenarios, I used PDPS to predict test patients' diagnoses in the final admission, given all their records before the final admission. As an evaluation measure, I used AUROC, for which 1 represents a perfect model and 0.5 represents a random model. The average AUROCs of a variety of diagnoses of different models were compared.

2.8.1 Incomplete Information

To evaluate the performance of the Procrustes harmonization, I first looked at how well missing diagnoses in one site could be represented by the diagnosis vectors from another site. Oftentimes, small clinics or medical systems do not encounter some medical diagnoses. In terms of PDPS prediction, missing diagnoses with no embeddings cannot be incorporated into the patient vector, making the prediction model less accurate. Moreover, prediction simply cannot be made with PDPS for certain diagnoses if there are no embeddings for those diagnoses.

I represented the missing information using diagnosis vectors from another medical system. In order to test whether diagnosis vectors from a medical system could be used accurately to predict diagnoses at another site, I first took the 40 most common diagnoses in MIMIC-III and randomly separated them into two sets of 20 diagnoses. The two '*local*' sites both originally had all 40 diagnoses, but I took one '*local*' site and deleted all instances of one set of 20 diagnoses (**Appendix 2.A** colored blue), and took the other '*local*' site and deleted the other set of 20 diagnoses (**Appendix 2.A** colored red). I trained and contextually embedded these two raw datasets separately, making two embedding models, where each lacked a different set of 20 diagnoses. I explored what vectors added to the '*local*' embeddings could compensate for the

missing diagnoses vectors. When I simply added random vectors for the 20 missing diagnoses to the respective sites and predict those 20 diagnoses, the average AUROC for those missing diagnoses for both sites was close to 0.5 (**Figure 2.7**). If I represented vectors for the missing diagnoses using embeddings from the other site without the Procrustes harmonization, the average AUROC only improved to ~ 0.55 . However, with the harmonization of one embedding to another, the AUROC improved to ~ 0.65 . When compared to benchmark models, where the 20 diagnoses of interest in each site were never deleted in the first place, I can see that vectors for the missing diagnoses with Procrustes harmonization perform just as well as the benchmark models. I also tested whether using different percentages of corresponding anchor pairs for the Procrustes harmonization would alter the AUROC. As expected, **Figure 2.7** shows that increasing the percentage of corresponding pairs increases the AUROC, but the effect is small.

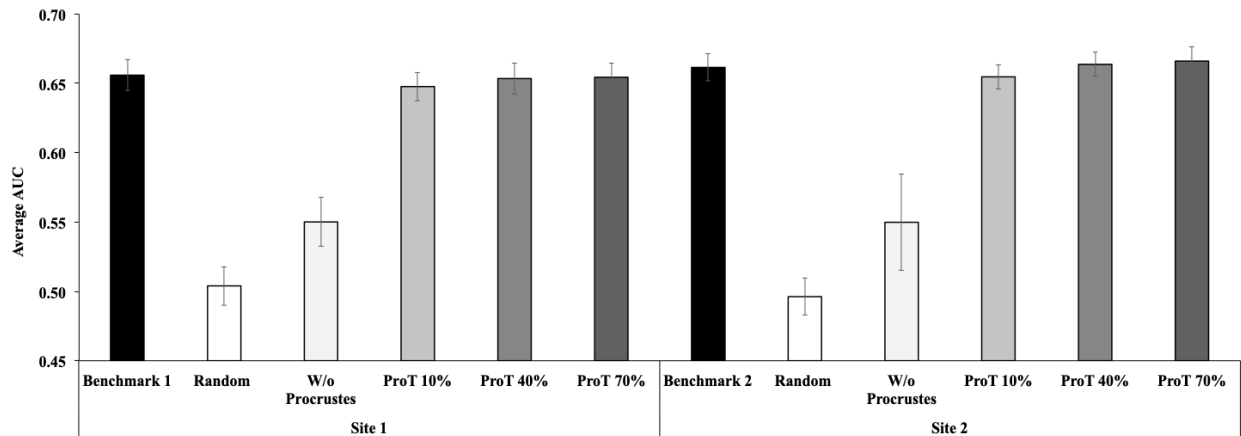


Figure 2.7 Harmonization of Embeddings

The average AUROC of several different scenarios. Site1 had no vectors for a set of 20 diagnoses, while Site 2 had no vectors for another set of 20 diagnoses. The Benchmark models did not have any missing vectors. Using the benchmark model, I predicted the 20 diagnoses missing from site 1 (*Benchmark 1*) and the 20 diagnoses missing from site 2 (*Benchmark 2*). In different scenarios, the missing vectors were compensated with either random vectors (*random*), untransformed vectors from the other site (*W/o Procrustes*), or Procrustes harmonized vectors from the other site that were harmonized using different percentage of corresponding pairs (*proT 10%, 40%, 70%*).

2.8.2 Split Patient History

In another scenario, it is conceivable that patients may go to different medical systems, leaving parts of their patient history in one medical system while other parts in other medical systems. One can simply predict future events for these patients using part of their clinical pathway at each site, but a more accurate prediction can be made from their entire clinical pathway. However, obtaining the entire clinical pathway is not easy. First, it might be time-consuming or even infeasible to release patient history across medical system sites for model building. Second, even if the entire patient history is in one site, the events in a patient's clinical pathway may be coded differently from site to site, leading to some events being unrecognizable by a model built solely on one site, and thus unusable for prediction. To solve these two problems, medical systems can first share their own contextual embeddings and combine them into a common space using Procrustes. Then, for all the patients who have histories in multiple medical systems, each local clinical pathway in each medical system can be made into a local patient vector. Finally, every local patient vector can be summed and normalized to obtain an approximation of the *'global'* patient vector. Then prediction can be conducted using the approximated patient vectors and diagnoses vectors in each *'local'* medical system. The following experiment shows that the initial harmonization step is required to obtain a high AUROC.

For this task, I divided the raw MIMIC-III training set into three *'local'* sites then trained three *'local'* embedding models. For medical events in each *'local'* site 1, 2, and 3, suffixes *'_m1'*, *'_m2'*, and *'_m3'* were added to the end, respectively, simulating that each *'local'* site used their own coding system. To simulate test patients who have records in three *'local'* sites, I divided the

clinical pathway of each test patient into three sections, where each section was appended with suffix ‘_m1’, ‘_m2’, or ‘_m3’ to designate which section belonged to which site. The average AUROCs of the 80 most common diagnoses from MIMIC-III (**Appendix 2.A** blue, red, and black) were evaluated based on PDPS for different models and are shown in **Figure 2.8**. A Benchmark model is trained with no splitting of training data or test data. The average AUROCs obtained with ‘local’ embedding models for sites 1, 2, and 3, using only the local patient vector, dropped significantly when compared to the global model (**Figure 2.8**). This was because each ‘local’ model could only use one-third of the information of each test patient for prediction. If I summed and normalized the three local test patient vectors without harmonizing the embeddings first, the AUROC did not improve (**Figure 2.8 Combined 1, 2, and 3, Original**). However, when the three contextual embeddings were harmonized with the Procrustes method (*ProT*) first, using the normalized summation of the local test patient vectors improved the AUROC, making it closer to the AUROC of predictions made by the benchmark model. **Figure 2.8** shows three ‘Combined’ results because each local site had its own diagnosis vectors that PDPS used; AUROCs were based on these vectors. I tested different percentages of corresponding pairs of 10%, 40% and 70% for the Procrustes harmonization. Similar to the previous experiment, **Figure 2.8** shows that increasing the percentage of corresponding pairs had a positive but small effect.

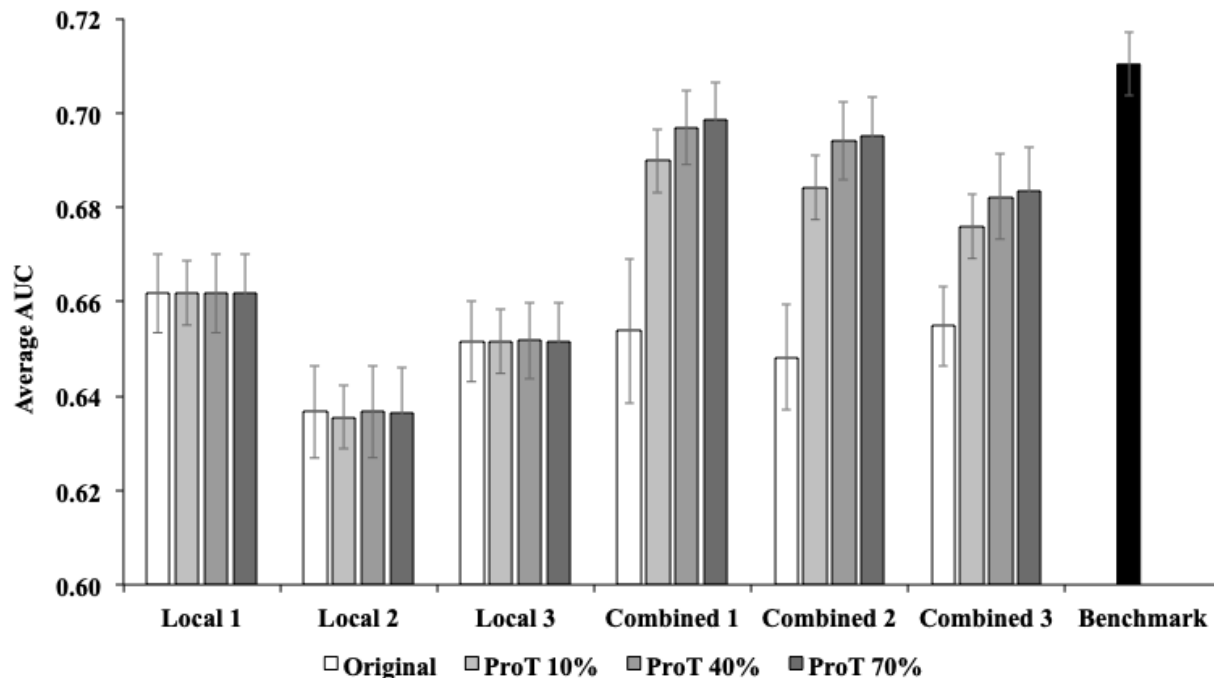


Figure 2.8 Split Patient Record Results

Average AUROC for the 80 most common diagnoses. *Local 1, 2, and 3* show results using local embedding models either unharmonized (*Original*) or harmonized (*ProT*) and using only the part of the clinical pathway of test patients in the respective medical system. *Combined 1, 2, and 3* show all three local test patient vectors combined where each local vector is made with locally learned event vectors either unharmonized (*Original*) or harmonized (*ProT*). *Benchmark* did not split the training set or test set into 3 parts. There are three *Combined* results because each local embedding model has its own diagnosis vector used by PDPS. Harmonization experiments were done with different percentages of corresponding pairs (10%, 40%, and 70%).

2.9 Unstructured Data

The harmonization method can be extended to unstructured clinical notes. Using the same method as I used for structured data, I built clinical pathways for unstructured data. However, I used medical concepts extracted from Metamap.⁴⁸ I built a global embedding model to serve as a benchmark and two ‘local’ embedding models. Then, I harmonized the two ‘local’ models with Procrustes. There were 150,034 unique medical concepts extracted with Metamap, of which 72,195 were common to both sites and could be used as anchor pairs. For anchor pairs,

I used the top 10% most commonly occurring corresponding pairs. The most commonly occurring anchor pairs were chosen because they were the most likely to have similar neighborhood and structure relative to other concepts in each 'local' hyper-dimensional space, giving us the most reliable transformation matrix Q in the Procrustes transformation. The following experiment was conducted to demonstrate the benefit of the harmonized local embedding models.

2.9.1 Patient Similarity Prediction Model

For the following experiment, which included unstructured data and limited structured data (only diagnosis labels), I explored patient similarity and whether the most similar patients from another medical system could be used to predict diagnoses. In order to predict for future diagnosis in this experiment, PDPS was not used because it required diagnoses vectors, which, for the structured data, were ICD-9 diagnosis codes learned during contextual embedding. However, contextual embedding for unstructured data was done on clinical notes, which did not contain any ICD-9 codes. Instead, I used the most similar training patients to each test patient to predict a test patient's future diagnosis (i.e., a nearest neighbor approach). To find the most similar patients for each test patient, I calculated the cosine similarity between a test patient and all training set patients. I calculated the average cosine similarity between that test patient and all the training set patients. Then, I used the training set patients whose cosine similarities were one standard deviation above the average to define the most similar patients, i.e., the neighborhood. Finally, prediction and subsequent AUROCs were calculated by weighted voting, where the weight was the cosine similarity of the two patients. Using the most similar training patients and their true ICD-9 diagnosis from the structured data as labels, the probability of a patient having a diagnosis was the total number of patients in the neighborhood with a diagnosis

divided by the total number of patients in the neighborhood.

After learning contextual embeddings for the two *'local'* sites, I again created training patient vectors for the two sites. In Site 2, I deleted all patients with a certain diagnosis of interest but retained patients with this diagnosis in Site 1. This experiment was done for the 80 most common diagnoses in MIMIC-III, with each diagnosis taking a turn in acting as the diagnosis of interest. The results are shown in **Figure 2.9**. Given a set of test patients, their patient vectors were created using either embedding from Site 1 or embeddings from Site 2, depending on which medical system they were admitted to. **Figure 2.9 (a)** shows the results where new test patients were admitted to Site 1, and their patient vectors were created with Site 1 embeddings. I predicted whether these new patients had the diagnosis of interest based on the most similar training patients in Site 1 and obtained a benchmark. However, when new test patients were admitted to Site 2 and I used the most similar training patients in Site 2 to predict the probability of the diagnosis of interest, the AUROC was 0.5, as shown in **Figure 2.9 (b)**. This is because Site 2 did not contain patients with the diagnosis of interest. Similarly, when new test patients were admitted to Site 2, and I used the most similar patients found from Site 1 to predict the diagnoses, the result was no better than guessing, as shown in **Figure 2.9 (c)**. This is because the embedding space was not harmonized and thus not enough similar patients were found. Finally, I harmonized the *'local'* embeddings between Site 1 and Site 2 first and created training patient vectors from them. When new test patients were admitted to Site 2 and their patient vectors were created with the harmonized *'local'* embedding of Site 2, I could use Site 1 to find the most similar patients and obtain an AUROC at the level of the benchmark, as shown in **Figure 2.9 (d)**. This shows that, if new patients were admitted to a medical system and the site lacked truly similar patients to

make accurate predictions for a diagnosis of interest, the patient records from another medical system could compensate and provide similar patients. However, such compensation could only be achieved if the contextual embeddings were harmonized.

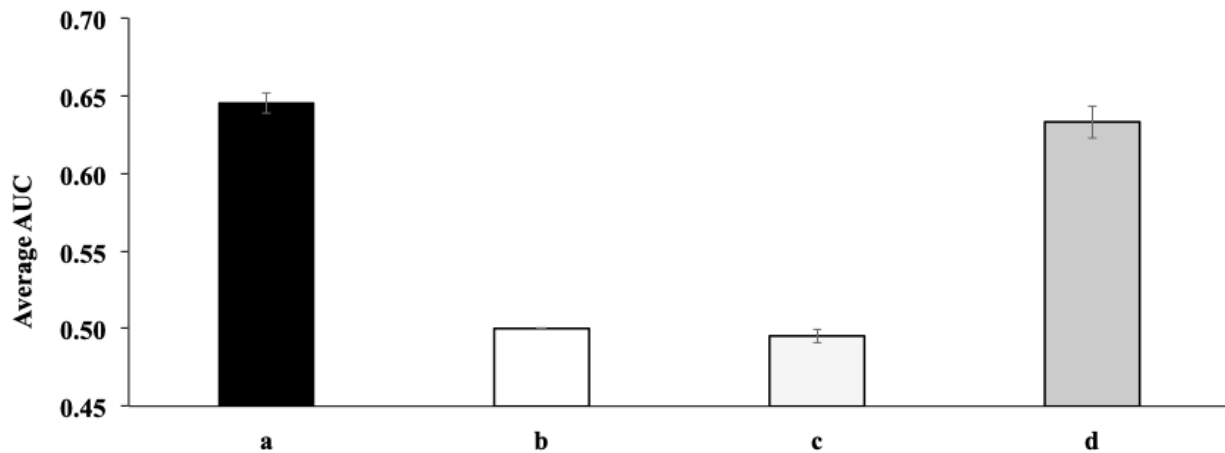


Figure 2.9 Split Patient – Unstructured Data

Average AUROC over 80 most common diagnoses, with each diagnosis taking a turn in acting as the diagnosis of interest. Site 1 contained patients with the diagnosis of interest, while Site 2 did not. (a) Patient vectors of new test patients were created with embeddings learned from Site 1, and training vectors from Site 1 were used to find the most similar patients for new test patients and to predict the diagnosis of interest. This is a benchmark. (b) Patient vectors of new test patients were created with embeddings learned from Site 2, and training vectors of Site 2 were used. Since Site 2 did not contain any patients with the diagnosis of interest, the AUROC was no better than guessing. (c) Patient vectors of new test patients were created with embeddings learned from Site 2, and training vectors of Site 1 were used to find the most similar patients. Since there was no Procrustes harmonization, the most similar patients from Site 1 to the test patients did not provide accurate predictions. The AUROC result was still no better than guessing. (d) Embeddings of Site 1 and Site 2 were harmonized prior to creating patient vectors. Then, patient vectors of new test patients were created with embeddings learned from 2, and training vectors of Site 1 were used to find the most similar patients. With Procrustes harmonization, the most similar patients in Site 1 to the test patients admitted to Site 2 were able to provide truly similar patients, allowing for the AUROC to reach the benchmark level.

2.10 Limitations and Future Directions

This Chapter described a proof-of-concept that contextual embedding models, which are becoming bedrocks to deep learning analysis in replacement of one-hot representations, can be harmonized, and subsequently synchronize information from different medical systems for better prediction capability, without sacrificing privacy. However, one limitation of our work is that all experiments were conducted on a single MIMIC-III database. The underlying structure of the simulated local models may be similar, making it easy to approximate the global model from combining Procrustes harmonized local models. However, it is possible that every medical system may have similar structures and relationships for medical events or concepts related to diagnoses that are common and widespread. Using events related to these common diagnoses, it should nevertheless be possible to derive a reasonable transformation matrix to apply to the rest of the data when the method is extended beyond the MIMIC-III database.

Ultimately, harmonization can bring knowledge contained in each medical system into the same space. This is a major benefit because, once embeddings are created for each medical event or concept, it becomes difficult to add new events or concepts in relation to the existing embeddings without training the model again. With harmonization, I can leverage embeddings learned from other sources and add new vectors. What this process fails to address at the moment is that, after harmonization, there are two vectors for the same event in some cases. Researchers have to make their own decision as to which of the vectors to use. It is an easy decision if the vectors have the same closest neighbors, but in situations where two medical

systems with conflicting embeddings regarding an event or concept, the current method does not alleviate the issue. A workaround would be to weigh the probabilities derived from using the two vectors and produce a consolidated estimate for a patient. However, to create a truly global model where two embeddings fully harmonize into one, further work is required. The current method shows the feasibility of incorporating new information that medical systems may be missing.

There are also situations in which harmonization of embeddings from different sites may be inappropriate. The knowledge that is specialized in one medical system may not benefit from the incorporation of embeddings from another. For example, an embedding learned from a medical system specialized in pediatrics may have unique knowledge, and the embeddings may have cosine distances that are different from embeddings learned from medical systems that specialize in adult care. The harmonization of these two medical systems may create conflicting embeddings, where two vectors for the same event learned from the two medical systems have different nearest neighbors. Care must be taken when applying downstream analysis. For example, if one was to make patient vectors of pediatric patients, one should not use embeddings from the medical system specialized in adult care without validation. Even if a medical event embedding is missing in the medical system specialized in pediatrics but exists in the medical system specialized in adult care, incorporation of embedding from the medical system specialized in adult care could add erroneous information and noise to the pediatric patient vectors.

Finally, while I showed in a limited way that the extension of the harmonization method onto three sites works, I also see that, even with harmonization, the prediction performance does

not reach the global level. This is expected given that the alignment of embeddings is typically imperfect.

2.11 Conclusion

The portability of medical events and patient vectors is the major strength of the method. With the medical events and patient vectors rendered to vectors of numbers, privacy is preserved yet information is still conveyed to medical systems involved in the harmonization. Instead of preserving privacy through de-identification and encryption, I take a machine-learning approach that relies on sharing the embeddings but not the data, hence I tackle privacy protection and the sharing of ‘aggregate data’ simultaneously. With patient privacy being a paramount concern, it is non-trivial to directly share medical records in both structured and unstructured form. The emergence of contextual embedding in healthcare allows for a new way to share models without sharing patient-level data. I proposed an innovative framework to combine locally trained embeddings into embeddings in a global sense. Utilizing this unique harmonization, more accurate analysis can be made with the accumulated knowledge acquired from local sources than would be possible using only local embeddings. Such a technique can allow for information unique to a certain medical system to become available to other sites, increasing the fluidity of information flow in health care. Our demonstration used Word2Vec but the approach is widely applicable to other contextual embedding models, including the most recent Med2Vec³⁹ and Graph2Vec.⁴⁹

Chapter 2, in part, is a reprint of the material as it appears in *Privacy-Preserving Predictive Modeling: Harmonization of Contextual Embeddings From Different Sources*. Huang, Yingxiang;

Lee Junghye; Wang, Shuang; Sun, Jimeng; Liu, Hongfang; Jiang, Xiaoqian. JMIR Medical Informatics, 2018. The dissertation author was the primary investigator and author of this paper.

Appendix 2.A Eighty Diagnoses used for Experiments

List of 80 most common diagnoses used for prediction.

Septicemia, Acquired hypothyroidism, Disorders of lipid metabolism, Purpura and other hemorrhagic conditions, Nondependent abuse of drugs, Essential hypertension, Old myocardial infarction, Other forms of chronic ischemic heart disease, Chronic pulmonary heart disease, Cardiac dysrhythmias, Heart failure, Hypotension, Asthma, Chronic airway obstruction not elsewhere classified, Pleurisy, Chronic kidney disease (CKD), Other disorders of urethra and urinary tract, Other disorders of bone and cartilage, Complications peculiar to certain specified procedures

Bacterial infection in conditions classified elsewhere and of unspecified site, Viral hepatitis, Diabetes mellitus, Disorders of fluid, electrolyte, and acid-base balance, Other and unspecified anemias, Anxiety dissociative and somatoform disorders, Depressive disorder not elsewhere classified, Organic Sleep Disorder, Hypertensive chronic kidney disease, Acute myocardial infarction, Other diseases of endocardium, Pneumonia organism unspecified, Pneumonitis due to solids and liquids, Other diseases of lung, Diseases of esophagus, Chronic liver disease and cirrhosis, Acute renal failure, Chronic ulcer of skin, Certain adverse effects not elsewhere classified.

Intestinal infections due to other organisms, Candidiasis, Secondary malignant neoplasm of respiratory and digestive systems, Secondary malignant neoplasm of other specified sites, Other and unspecified protein-calorie malnutrition, Gout, Disorders of mineral metabolism, Overweight obesity and other hyperalimentation, Iron deficiency anemias, Coagulation defects, Diseases of white blood cells, Transient mental disorders due to conditions classified elsewhere, Persistent mental disorders due to conditions classified elsewhere, Alcohol dependence syndrome, Pain, Epilepsy and recurrent seizures, Other conditions of brain, Inflammatory and toxic neuropathy, Other retinal disorders, Cardiomyopathy, Late effects of cerebrovascular disease, Atherosclerosis, Aortic aneurysm and dissection, Other peripheral vascular disease, Other venous embolism and thrombosis, Varicose veins of other sites, Other bacterial pneumonia, Chronic bronchitis, Other diseases of respiratory system, Intestinal obstruction without mention of hernia, Diverticula of intestine, Functional digestive disorders not elsewhere classified, Other disorders of intestine, Liver abscess and sequelae of chronic liver disease, Diseases of pancreas, Gastrointestinal hemorrhage, Hyperplasia of prostate, Other cellulitis and abscess, Osteoarthritis and allied disorders

3. Calibration Measurements and Calibration Models

3.1 Overview

So far, my model performance assessment has been based on discrimination. This is the general trend of current predictive model assessment: the main focus is on discrimination, while calibration is often not measured or delegated to a supplementary role. Calibration is the measure of how close the model's estimates are to the true probability. A calibrated predictive model would produce estimates that match the underlying true probability. I will introduce measurements of calibration and calibration models to improve improperly calibrated models.

3.2 Introduction

Most clinicians can recall seeing that inpatient who was listed as 50 years old but appeared decades older due to the effects of chronic illness, a physical exam finding commonly described as "Appearing older than stated age." And yet, the patient's stated age is used to dictate much of their care, including the calculation of glomerular filtration rate for medication dosing, risk of developing illnesses, and risk of death. Experienced clinicians can "calibrate" their mental models to account for the patients' appearance, but predictive models cannot do this without further instructions. When predictive models are built based on a population that differs from the population in which they will be used, blind application of these models could result in large "residuals" (i.e., a large difference between a model's estimate and the true outcome) because of factors that are difficult to account for. This deficiency could lead to catastrophic decisions for a single patient, even when the average residual for the overall population is very low. The analysis of such residuals can serve as a proxy for measuring the "calibration" of the

model. While *calibration-in-the-large* is concerned about gross measurements of calibration, such as whether the model's overall expected number of cases exceeds the observed number, or whether the proportion of expected over observed cases departs significantly from "1," other measurements of calibration are based on population stratifications, which can include anything from analyzing residuals on a few large subgroups all the way to analyzing residuals for each individual. Calibration is an essential component of the evaluation of computational models for medical decision making, diagnosis, and prognosis.^{50,51} In contrast to discrimination, which refers to the ability of a model to rank patients according to risk, calibration refers to the agreement between the estimated and the "true" risk of an outcome.⁵² A well-calibrated model is one that minimizes residuals, which is equivalent to saying that the model fits the test data well. Note that observing small residuals on the training set does not necessarily mean that it is a good model, since "overfit" models are known not to generalize well to previously unseen data (i.e., the residuals could be large in new test cases).

There are a number of cases that illustrate the omnipresence and importance of calibration and its critical role in model evaluation. If individualized predictions are used for clinical decision making, well-calibrated estimates are paramount. Take the case of dementia, a neurodegenerative disorder that affects at least 14% of Americans and recently costed the US healthcare system over \$150 billion/yr.⁵³ One recent study evaluated the calibration-in-the-large of several models that predict the risk of developing dementia in the general community and found that models drastically overestimated the expected incidence of dementia.⁵⁴ At a predicted risk of 40%, the observed incidence was still only 10%, so the test overestimated incidence by 30%. For an individual patient, and for the healthcare provider, an overestimation

of this magnitude could lead to different decisions. For example, it is recommended by the American College of Cardiology and the American Heart Association that patients with a cardiovascular risk over 7.5% be prescribed statins, and those between 5 and 7.5% be considered for this type of treatment.⁵⁵ Even risk calculators that are not based on percentages may benefit from calibration. An example is MELD,⁵⁶ which provides a risk score that is used to prioritize cases for liver transplantation. When score thresholds are used (for example, to determine which patients are eligible for transplants), calibration becomes critical. Accurate, well-calibrated estimates are necessary to allocate resources appropriately. Additionally, even when the models are well calibrated-in-the-large (e.g., the average predicted risk was 40% and the observed incidence was also 40%), there could be severe discrepancies to particular groups of individuals. It is thus critical to understand how a model will be employed in order to emphasize certain performance measures.

This Chapter covers some techniques to assess and correct model calibration in the context of employing clinical predictive models to estimate individualized risk. I present issues with measures of calibration that go beyond calibration-in-the-large, and I include examples of some calibration models that have been recently used in the biomedical literature. This Chapter does not include all available measurement methods and calibration models and is complementary to book chapters and articles that serve as references to this topic and will be of interest to those who want to deepen their understanding of model calibration.^{52,57-59} I provide here some simple and interpretable calibration measures and calibration models that can illustrate the concepts and have appeared in the recent biomedical predictive modeling literature so that clinical researchers and informaticians may familiarize themselves with this topic. Despite

its importance to understanding the utility of a model, calibration is vastly underreported: one systematic review noted that, while 63% of published models included a measure of discrimination, only 36% of models provided a measure of calibration.⁶⁰ Precision medicine involves individualized prevention, diagnosis, and treatment. It thus needs to rely on predictive models that are well-calibrated. I describe, in a didactic manner, key steps for measuring calibration and applying calibration models to a predictive model.

3.3 Ranking Patients vs. Estimating Individual Risk

Initial comparison and selection of appropriate models are often done through the evaluation of discrimination, which is measured with the Area Under the Receiver Operating Characteristic Curve (AUROC), but the AUROC says nothing about the calibration of the model.

Figure 1 shows how relying on AUROC overlooks calibration. **Figure 1 a, b, and c** contain three models' predicted estimates, sorted in ascending order, for two groups (*Alive* = "0" and *Deceased* = "1"). The three models' estimates are

a) original estimates,

b) original estimates divided by 10, and

c) original estimates after applying a calibration model to have the estimates be closer to the actual outcomes.

The AUROC, which is equivalent to the concordance index,^{61,62} can be easily calculated by counting the arrows in **Figure 3.1**. The blue arrows, showing discordant pairs, indicate pairs of estimates where the estimates for the *Alive* patients (coded as 0) are greater than the estimates for the *Deceased* patients (coded as 1), while the orange arrows indicate pairs of estimates where estimates for *Alive* and *Deceased* patients are equal (ties).

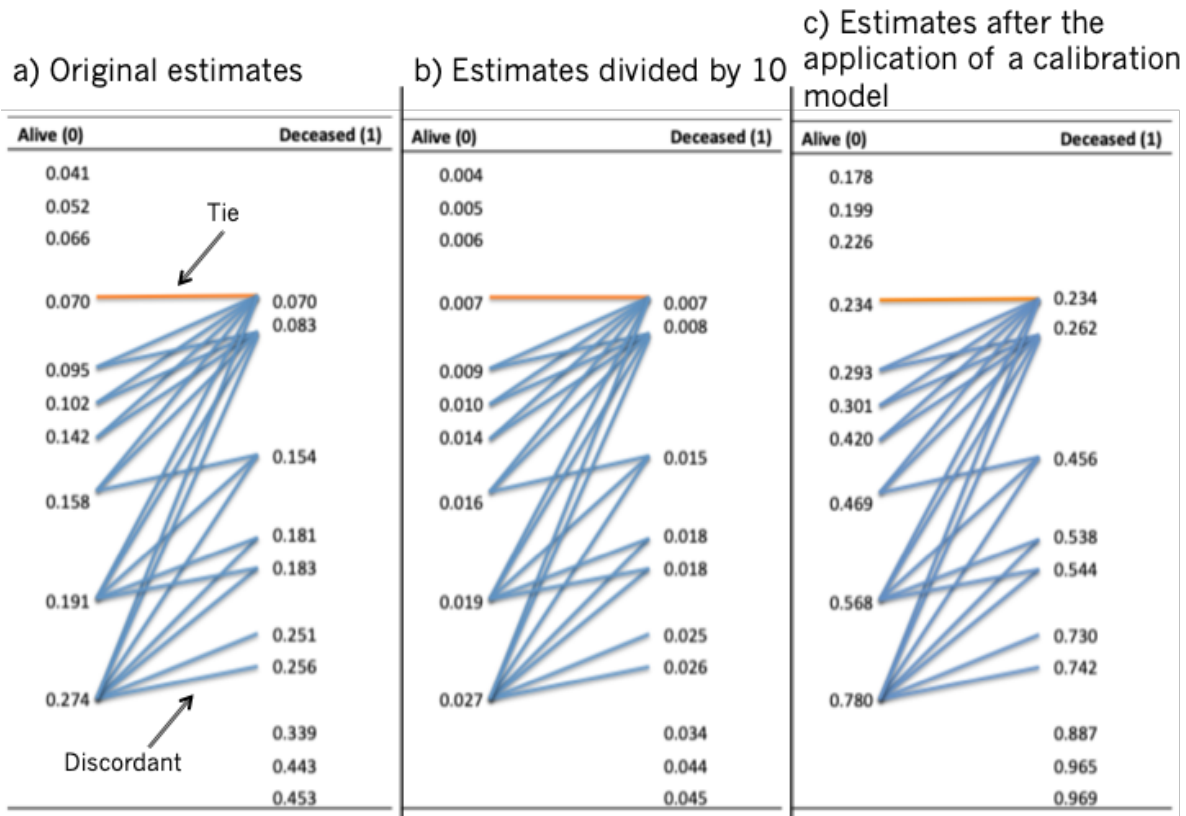


Figure 3.1. Illustration of discordant pairs, ties, and ROC Curve

a) Original estimates. **b)** Original estimates divided by 10. **c)** Original estimates after the application of a calibration model to have the estimates be closer to the actual outcomes. The blue arrows, showing discordant pairs, indicate pairs of estimates where the estimates for the *Alive* patients (coded as 0) are greater than the estimates for the *Deceased* patients (coded as 1), while the orange arrows indicate pairs of estimates where estimates for *Alive* and *Deceased* patients are equal (ties). The Area Under the ROC Curve (AUROC) is equivalent to the concordance index, which can be calculated here by the number of concordant pairs (i.e., total number of pairs minus the discordant and half of the tied pairs) over the total number of pairs, shown in the text as equation (3.1). **d)** Identical ROC curve and AUROC (0.785) for the three models.

d)

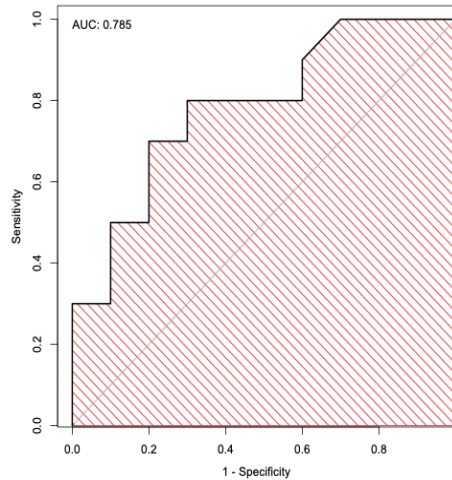


Figure 3.1. Illustration of discordant pairs, ties, and ROC Curve, continued

The non-parametric AUROC can be calculated by the concordance index as follows:

$$(3.1) \quad \text{concordance index} = \frac{\text{total \# pairs} - 1 \times (\# \text{ discordant pairs}) - 0.5 \times (\# \text{ ties})}{\text{total \# pairs}}$$

where the ‘total # pairs’ is the number of pairs of *Alive* and *Deceased* estimates, ‘# discordant pairs’ is the number of pairs composed of one *Alive* and one *Deceased* patient in which the estimate for the *Deceased* (coded as “1”) is lower than the estimate for the *Alive* (coded as “0”) patient, and the ‘# ties’ is the number of pairs in which the estimates are equal. Intuitively, the blue arrows are called “discordant pairs” because, in a model that has perfect discrimination, no estimates for observations in the *Alive* group should be greater than estimates for observations in the *Deceased* group. As illustrated in **Figure 3.1**, there is no need for the actual values of the estimates: Only the ranks (i.e., the order of the estimates) are necessary to calculate the concordance index or the AUROC. Therefore, when equation (3.1) is applied to estimates in **a, b,**

and **c**, the resulting ROC curves for all three models are the same, and their corresponding AUROCs are also identical, as shown in **Figure 3.1d**. However, the three models' estimates are vastly different. Such differences are reflected in the absolute errors between the estimates and actual outcomes, as well as the average estimates for *Alive* and *Deceased* (**Table 3.1**). These values help give an idea of the models' calibration, but still do not reveal whether there is gross under- or overestimation of the probability of *Deceased* for particular groups or individuals. Without a means for measuring calibration and for choosing appropriate models accordingly, erroneous decisions could be made in processes that rely on the values of the estimates. For example, if a clinical practice guideline recommends that all individuals at greater than 20% risk receive a certain intervention, a non-calibrated model such as the one in **Figure 3.1b** could result in no one receiving the intervention. In this Chapter, I illustrate methods that measure calibration in different ways. For models that have improper calibration, I show how calibration models can mitigate the problem of poorly calibrated models.

Table 3.1 Average Estimates and Observed Outcomes

AUROC: Area Under the Receiver Operating Characteristic Curve.

	Figure 3.1 a)	Figure 3.1 b)	Figure 3.1 c)
AUROC:	0.785	0.785	0.785
Average Outcome	0.500	0.500	0.500
Average Absolute Error:	0.439	0.499	0.367
Average Estimate:	0.180	0.002	0.500
Average Estimate (Alive):	0.115	0.001	0.349
Average Estimate (Deceased):	0.241	0.002	0.633

3.4 Simulated Data

I used simulated data to demonstrate the forthcoming calibration measures and calibration models. The code can be found in https://github.com/easonfg/cali_tutorial. To create artificial data, I utilized the method from *Zimmerman et. al.*⁶³ Twenty-three artificial features were constructed with 20 binary and 3 continuous independent variables. A uniform distribution was then used to determine the dependent variable, mortality, where 0 indicates *Alive* and 1 indicates *Deceased*. The observed mortality was 15%. To compare models, a logistic regression

(LR) model and a Support Vector Machine (SVM) model with a linear kernel were built. SVM uses the Hinge Loss function as its objective function, so it often produces improperly calibrated estimates.⁶⁴ I present different measures of calibration and the results of calibration models.

Five thousand samples were created. The entire simulated data were separated into three parts: (1) Training set part 1 (2500 samples), (2) Training set part 2 (1250 samples), and (3) Test set (1250 samples). I used hold-out validation: I trained the classifier on Training set part 1, I trained the calibration models on Training set part 2, and I used the Test set to “validate”, i.e., to assess performance when the model is used in previously unseen cases.

For this Chapter, two classification models (LR and SVM) were trained on the Training set part 1 and calibration models were built based on the resulting model applied to Training set part 2. Then the classifier and calibration models were applied to the Test set and evaluation of calibration and discrimination were calculated on the Test set’s predicted estimates. I compared discrimination and calibration on the original Test set estimates (i.e., pre-application of the calibration model) and on the calibrated estimates. By separating the entire data into three parts and training the classification model and calibration model on different datasets, the aim is to avoid overfitting.

3.5 Measuring Calibration

There is no best method to measure the calibration of predictive models. While some methods are frequently used and have specific strengths, all have limitations. Here I present these calibration assessment methods and the scenarios in which each can be used appropriately. Additionally, some methods combine implicit measures of calibration with other components such as discrimination, which may be difficult to separate.

3.5.1 Brier Score and Spiegelhalter's Z-Test

The Brier score is the mean square error between the actual outcome and the estimated probabilities, as shown in equation (3.2):

$$(3.2) \text{ Brier Score} = \frac{\sum_{i=1}^N (E_i - O_i)^2}{N}$$

where N is the number of patients, E_i is the predicted estimate for patient i , and O_i is the actual outcome for patient i . The Brier score should be interpreted carefully. Without understanding whether the error is caused by a relatively small number of estimates with high errors or a large number of estimates with smaller errors, it is difficult to say whether this model could be used in practice. Note that, by squaring errors that are in the $[0,1]$ range, large errors “count less” to the overall score, when compared numerically to smaller errors. The Brier score includes components of discrimination and calibration, so a lower Brier score does not necessarily imply higher calibration.⁶⁵ However, it can be shown that, from the decomposition of the Brier score, a formal measurement that can serve as a proxy for calibration can be calculated: the Spiegelhalter's Z-test.⁶⁶ The z statistic can be calculated with equation (3.3).

$$(3.3) \quad Z(E, O) = \frac{\sum_{i=1}^N (O_i - E_i)(1 - 2E_i)}{\sqrt{\sum_{i=1}^N (1 - 2E_i)^2 E_i (1 - E_i)}}$$

If $|Z(E, O)| > q_{1-\alpha/2}$, where q_α is the α -quantile of the standard normal distribution (0.05),

the result is significant, suggesting an improperly calibrated model.

The discrimination (AUROC), Brier scores, and Spiegelhalter’s Z-test results for the LR and SVM models are shown as AUROC in **Table 3.2**, as are other measures described in subsequent sections of this Chapter. The AUROCs of the two models are the same, but there is a difference in Brier scores. Also, p-values for the Spiegelhalter’s Z-test indicate that the SVM classifier is not well calibrated.

Table 3.2. Areas Under the ROC Curve (AUROC), Brier scores, and Spiegelhalter’s Z-test statistics for Logistic Regression (LR) and Support Vector Machine (SVM) models.

	LR	SVM
AUROC	0.870	0.870
Brier	0.087	0.111
Spiegelhalter Z Score	0.762	2.214
Spiegelhalter p-value	0.223	0.013

The calculation of the Brier score is relatively simple. A line of code is sufficient or use of packages that calculate the Brier score in R, such as the “rms” package with the function “val.prob” and the “DescTools” package with the function “BrierScore” (packages that implement the Brier Scores and all subsequent calibration methods conducted with the simulated data are listed in **Table 3.9**). The Spiegelhalter’s z-statistic can also be calculated with the function “val.prob” from the “rms” package.

3.5.2 Average Absolute Error

Another easily implemented measure is the average absolute error, which is calculated in Equation (3.4).

$$(3.4) \text{ Average Absolute Error} = \frac{\sum_{i=1}^N |E_i - O_i|}{N}$$

The average absolute error is very similar to the Brier score, but small and large errors contribute in the same way to the sum. The results for LR and SVM are shown in **Table 3.3**.

Table 3.3. Average Absolute Error for Logistic Regression (LR) and Support Vector Machine (SVM) models.

	LR	SVM
Average Absolute Error	0.177	0.236

3.5.3 The Hosmer-Lemeshow Test

The Hosmer-Lemeshow (H-L) test has been a popular measure of goodness-of-fit for predictive models of binary outcomes and is sometimes used as a proxy for calibration⁶⁷ despite its shortcomings, which I describe in the Discussion section. A PubMed search returns hundreds of articles per year mentioning the Hosmer-Lemeshow goodness-of-fit test. Despite its age and known shortcomings, this test has been frequently used in health sciences research. Because the way in which it groups observations is also used in Reliability diagrams and Calibration curves, I explain some details here.

There is no ideal method to assess the calibration of models. A calibration measure could ideally compare the predicted estimate with the “true” probability for each patient, but the measurement of actual probability for a single individual is challenging. While the Brier score provides an aggregate score obtained from a single group (the whole population), the H-L test first assembles the individuals using the predicted estimates to form groups, for which the proportion of events can be calculated, serving as the “gold standard” for the group. That is, in the data, I can only ascertain the binary outcome but not actual or “true” probabilities, therefore a proxy for this probability is used. Forming groups of individuals and calculating the proportion of positive outcomes is a way to achieve such a proxy. For example, I can say that the actual probability of death is 10% for a patient if 10 out of 100 patients “just like” this patient died. A critical step is to determine who is “just like” this patient. For the H-L test and some other calibration methods, patients who are “just like” each other are patients whose predictive model’s estimates belong to the same group (i.e., patients who received similar estimates once a model is applied), and the ratio of event and non-event within each group is the proxy for the “true” probability for the patients in that group. There are two ways by which the H-L test assigns individuals to the same group, resulting in H-L *C* or H-L *H* statistics. For H-L *C* statistics, patients are divided into g groups where each group contains approximately the same number of patients, typically grouped deciles of risk. For *H* statistics, groups are divided based on equal increment thresholds for the estimates (e.g. if there are 10 groups, estimates in the interval $[0, 0.1]$ belong to one group, estimates in the interval $(0.1, 0.2]$ belong to the second group, and so on). This is shown in **Figure 3.2**.

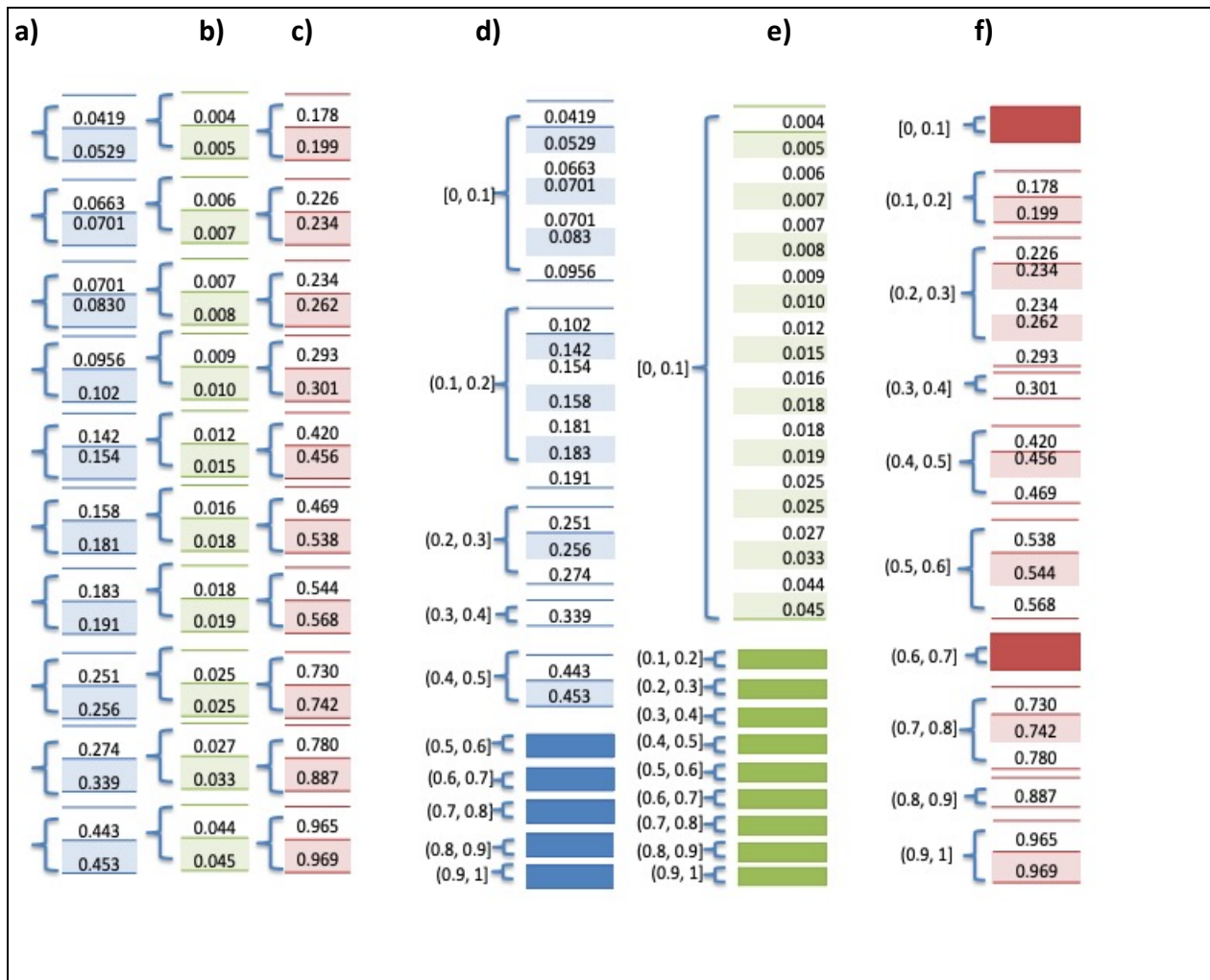


Figure 3.2 Grouping Methods for H-L C and H-L H Statistics

The small number of observations would not warrant a test but serves to illustrate the contrast between two different ways of forming groups for the HL test.

- For H-L C statistics, patients are divided into g groups where each group contains approximately the same number of patients, typically grouped by deciles of risk ($g=10$).
- For H-L H statistics, groups are divided based on equal increment thresholds for the estimates (e.g. if there are 10 groups, estimates in the interval $[0, 0.1]$ belong to one group, estimates in the interval $(0.1, 0.2]$ belong to the second group, and so on).

Numbers shown in a blue background correspond to **Figure 3.1a** estimates, green corresponds to **Figure 3.1b**, and red corresponds to **Figure 3.1c** estimates. **a)**, **b)**, and **c)** show groups of estimates using deciles of samples utilized for the H-L C Statistics. **d)**, **e)**, and **f)** show groups of estimates using equal interval groups utilized for the H-L H Statistics. Since the degree of freedom is equal to $(g - 2)$, the degrees of freedom for groups in **a)**, **b)**, **c)**, **d)**, **e)**, and **f)** are 8, 8, 8, 3, N/A, and 6, respectively.

H-L test statistic and p-value. The total estimates for ‘similar’ patients in a group are compared to total observed outcomes within each group. The H-L C or HL H statistics can be calculated using equation (3.5),

$$(3.5) \quad test\ statistic = \sum_{i=1}^g \left[\frac{(O_{s,i} - E_{s,i})^2}{E_{s,i}} + \frac{(O_{f,i} - E_{f,i})^2}{E_{f,i}} \right]$$

where $O_{s,i}$ is the number of patients with the outcome “1” within each group i and $E_{s,i}$ is the sum of the estimates of patients with the outcome “1” within each group i . $O_{f,i}$ is the number of patients with the outcome “0” within each group i , and $E_{f,i}$ is the sum of the estimates of patients with the outcome “0” within each group i . Finally, g is the number of groups. The distribution of the test statistics follows a chi-square distribution with $(g - 2)$ degrees of freedom. The p-value can be subsequently calculated. A p-value of 0.1 or higher is considered appropriate, a p-value < 0.1 and > 0.05 indicates that the model is neither well-calibrated nor grossly miscalibrated, and a p-value < 0.05 indicates miscalibrated estimates.⁶⁸

The ideal number of groups and the method to determine membership in a group are often points of contention when performing an H-L test. With H-L C statistics, where an approximately equal number of samples are in each group, the range of the estimates could differ wildly, as shown in **Figure 3.2**. Take our small toy example, partitioning of groups in **Figure 3.1a**, **3.1b**, and **3.1c** for the C statistics is shown in **Figure 3.2a**, **3.2b**, and **3.2c**, respectively. Patients in the same group can have very close or very distant estimates depending on how groups are formed. With the H-L H statistics, on the other hand, where estimates are partitioned into groups

according to equal intervals, there could be groups with no patients, or others with a large number of them, as seen in **Figure 3.2e**. The H-L C statistic is used much more frequently than the H-L H statistic in practice, but the groups formed in the latter are frequently used to plot calibration plots or reliability diagrams, which I will explore later in the Chapter.

H-L statistics and p-values are calculated and shown in **Table 3.4**. The p-values for the SVM model ($p=0$) are significant, indicating improper calibration, while LR is properly calibrated ($p>0.1$). The packages and functions that implement the H-L test are listed in **Table 3.9**. The package ‘ResourceSelection’ from R with function ‘hoslem.test’ and the package “generalhoslem” with function “logitgof” both provide the same calculation of the H-L test.⁶⁹ However, these packages are only capable of calculating the H-L C statistics. A modified method is presented in the Github file that allows calculations for both H-L C and H statistics.⁴⁵ H-L statistics and p-values results are calculated and shown in **Table 3.4**. for using these methods on the test set estimates that were produced by the LR and SVM models for our simulated data. The p-values for the SVM model ($p=0$) are significant, indicating improper calibration, while LR is properly calibrated.

Table 3.4. Calibration results measured with the Hosmer-Lemeshow (H-L) test statistics and p-values for Logistic Regression (LR) and Support Vector Machine (SVM) models.

	LR	SVM
H-L C Statistics	5.886	176.482
H-L C p-value	0.661	<1e-22
H-L H Statistics	9.185	160.274
H-L H p-value	0.327	<1e-22

3.5.4 Reliability Diagram

The reliability diagram is a visualization technique that uses observation groupings such as the ones formed for the H-L test.⁶⁴ However, instead of the sum, the mean of actual outcomes of each group is plotted against the mean of estimates of each group. While the points are typically connected to help with visualization, it is obviously not a true curve. No information can be derived between the points on the diagram. A perfectly calibrated model would result in a 45-degree line. **Figure 3.3** shows the reliability diagrams for the LR and SVM models using H-L C and the HL H statistics in **a)** and **b)**, respectively. The actual data are also plotted for reference. While the reliability diagram of LR follows the diagonal line, I can see that the reliability diagram for the SVM model deviates from the diagonal, trending upward. This indicates improper calibration and underestimation of the actual number of *Deceased* (outcome = 1) in some groups.

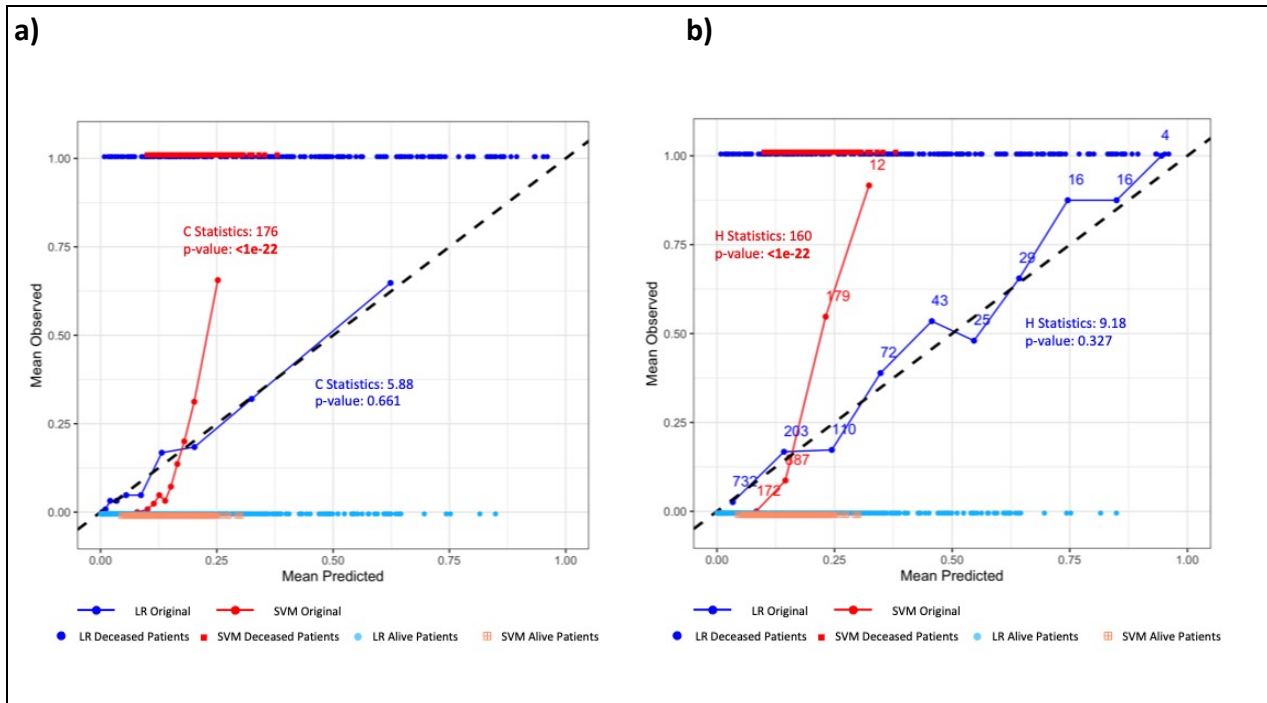


Figure 3.3 Reliability Diagrams of Test Set Estimates Produced by Logistic Regression (LR) and Support Vector Machine (SVM) Models

Data points of estimates produced by the models and their actual binary outcomes are plotted. The *Alive* outcome is indicated as 0 and the *Deceased* as 1. Corresponding H-L statistics and p-values are shown in the graphs. **a)** LR and SVM estimates grouped for the calculation of the H-L C statistic. The number of patients within each bin is the same (125). **b)** estimates grouped for the calculation of the H-L H statistic. The number of patients in each group is shown in the graph. Both graphs show that SVM underestimates the actual number of deaths, as shown by the red line deviating from the diagonal line. LR is relatively well-calibrated (blue line).

Table 3.9 shows R packages and commands for the calibration measures discussed in this chapter. Neither grouping method is perfect when drawing the reliability diagram. As I can see in **Figure 3.3a**, by using the grouping method of H-L C statistics, there are no data points for estimates above 0.7 because of the small number of patients in that region. With the H-L H statistics, since I am using equal increment intervals, the reliability diagram extends beyond 0.7, allowing a glimpse of the calibration at those points. However, there are few patients in those intervals.

A package from “PresenceAbsence” in R is able to draw a reliability diagram with the function “calibration.plot”.⁷⁰ This method groups the estimates according to the H-L H statistics and plots the average of the actual number of positive outcomes against the midpoint of each group’s interval. R packages for the H-L test use the H-L C statistics grouping method, while the reliability diagram typically uses the H-L H statistics grouping method.

3.5.5 Expected Calibration Error and Maximum Calibration Error

Aside from the H-L test and the Reliability diagram, grouping was also used in recent papers to calculate the Expected Calibration Error (ECE) and the Maximum Calibration Error (MCE).^{71–73} To compute the ECE and the MCE, predictions or estimates are sorted and divided into K groups with an approximately equal number of patients in each group. The ECE calculates the average calibration error over the bins, while the MCE calculates the maximum calibration error for the bins in equation (3.6),

$$ECE = \sum_{i=1}^k P(i) \cdot |O_i - E_i|$$

$$(3.6) \quad MCE = \max_{i=1, \dots, K} (|O_i - E_i|)$$

where $P_{(i)}$ is the fraction of all patients who fall into group i . The number of groups I used is 10. Results for simulated data are shown in **Table 3.5**: LR has smaller ECE and MCE than SVM does, confirming what I already knew via the H-L C and H-L H tests.

Table 3.5. Calibration results measured with the Expected Calibration Error and Maximum Calibration Error for Logistic Regression (LR) and Support Vector Machine (SVM) models.

	LR	SVM
ECE	0.038	0.403
MCE	0.014	0.109

3.5.6 Cox Intercept and Slope

Unlike the above methods, Cox’s intercept and slope do not group estimates into bins. The Cox method assesses calibration by regressing the observed binary outcome to the log odds of the estimates with a general linear model, as shown in Equation (3.7),²⁴

$$(3.7) \text{logit}\{P(O_i = 1)\} = a + b \text{logit}(E_i)$$

Here a is the regression slope and b is the intercept. The estimated regression slope dictates the direction of miscalibration, where 1 denotes perfect calibration, >1 denotes underestimation of high risk and overestimation of low risk, and <1 denotes underestimation of low risk and overestimation of high risk. The estimated regression intercept represents the overall miscalibration, where 0 indicates good calibration, >0 denotes an average underestimation, and <0 denotes an average overestimation. The results of Cox’s slope and intercept are shown in **Table 3.6**. The slope and intercept for LR are close to 1 and 0, respectively, indicating a proper calibration. SVM, on the other hand, exhibits an underestimation of high risk and overestimation of low risk given its slope >1 and exhibits overall underestimation given its intercept >0 .

Table 3.6. Calibration results measured with the Cox’s Slope and Cox’s Intercept for Logistic Regression (LR) and Support Vector Machine (SVM) models.

	LR	SVM
Cox’s Slope	1.070	5.014
Cox’s Intercept	0.080	6.193

3.5.7 Integrated Calibration Index

Similar to Cox’s method, the Integrated Calibration Index (ICI) assesses calibration by first regressing the binary response to the estimates. However, ICI uses a locally weighted least squares regression smoother (i.e., the loess algorithm).⁷⁴ Cox’s slope and intercept can equal 1 and 0, respectively, while deviations from perfect calibration can still occur (e.g., when these deviations “cancel” each other in terms of the linear regression). However, these deviations can be captured by the loess smoother, and a subsequent numerical summary is the ICI. ICI takes the average of the absolute difference between the estimates and the predicted estimates based on the loess calibration curve. The results for LR and SVM are shown in **Table 3.7**. The ICI of SVM is higher than the ICI of LR, and particularly for estimates before application of calibration models, as expected.

Table 3.7. Calibration results measured with the Integrated Calibration Index for Logistic Regression (LR) and Support Vector Machine (SVM) models.

	LR	SVM
ICI	0.010	0.104

3.6 Calibration Models

Calibration models can be applied to improve calibration performance, and there are two ways to attempt to obtain calibrated estimates. The first approach is to include measures and terms in the objective function that specifically cater to calibration during model development.⁷⁵ When re-training a model to emphasize calibration is not feasible, it is sensible to improve calibration by applying calibration models to the estimates produced by the classifiers. The advantage of applying calibration models to estimates is that the method can be used in addition to any existing classification method and adjusted to the local patient population.

In a relatively recent article from the biomedical informatics literature, *Walsh et al.*^{76,77} re-emphasize the calls from *Van Calster et al.*⁷⁸ and *Riley et al.*⁷⁹ on the importance of calibration in addition to discrimination when evaluating predictive models. *Walsh et al.* select the following calibration models for their experiments: logistic calibration, Platt scaling, and prevalence adjustment. I utilize Platt scaling,⁸⁰ isotonic regression,⁸¹ and the Bayesian Binning Quantiles (BBQ)⁷³ calibration models to illustrate differences in calibration.

3.6.1 Platt Scaling

Platt scaling transforms model estimates by passing the estimates through a trained sigmoid function.⁸⁰ The sigmoid function is shown in (3.8) exactly as it appears in the original paper:

$$(3.8) \quad P(y = 1|f) = \frac{1}{1 + \exp(-(Af + B))}$$

where f is the predicted estimate and parameters A and B are derived using gradient descent.

Figure 3.4a shows the fitted sigmoid function derived using Training set part 2. Platt scaling trains this sigmoid function with the codomain constrained to the interval $[0,1]$, using the built-in function 'glm' in R, with link function 'logit'. It is a univariate logistic regression model that uses the model estimates as independent variables and the binary outcomes as dependent variables.

3.6.2 Isotonic Regression

Isotonic regression uses a step function with monotonically increasing values on the estimates.⁸¹ There are two algorithms to find the stepwise function. One is the pair-adjacent violator algorithm and the second is the active set algorithm.⁸¹ Both minimize residuals (i.e., the difference between the true outcome and the estimate) under the assumption that there is no change in the ranking of estimates (3.9):

$$(3.9) \quad \hat{y}^{iso} = \underset{\hat{y} \in \mathbb{R}^N}{\operatorname{argmin}} \sum_{i=1}^N (O_i - \hat{y}_i)^2 \text{ subject to } \hat{y}_1 \leq \dots \leq \hat{y}_N$$

where O_i is the actual outcome and \hat{y}_i is the fitted value. **Figure 4b** shows an example of a fitted isotonic curve derived using Training set part 2. R has a built-in function, 'isoreg', that fits the best monotonically increasing step function using model estimates as the independent variable and the binary outcomes as the dependent variable.

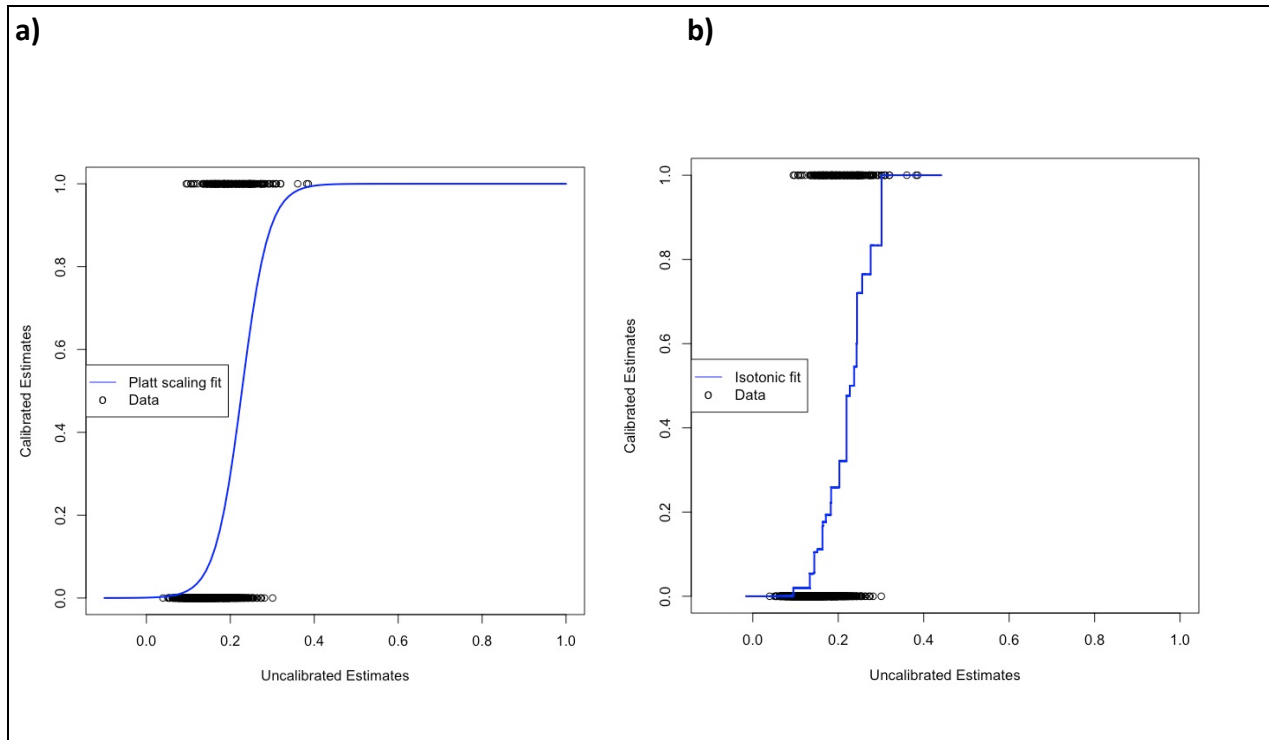


Figure 3.4 Calibration Models Functions

a) Example of fitted sigmoid function on SVM Training set part 2 estimates (Platt scaling). **b)** Example of a fitted isotonic regression on the Training set part 2 estimates.

3.6.3 Bayesian Binning Quantiles

In addition to Platt scaling and isotonic regression, an ensemble method called Bayesian Binning Quantiles (BBQ) has been recently proposed to improve calibration.⁷³ BBQ is based on quantile binning developed by Zadrozny and Elkan.⁸² In quantile binning, estimates are partitioned into K bins of equal number of patients. For every estimate within each bin, an estimate is calibrated to be equal to the fraction of positive samples in that bin. One drawback of quantile binning is the arbitrariness of K . BBQ takes an ensemble approach, calculating multiple binning size models and combining them. Individual calibration functions calculated with different size bins are combined with a weighted sum.⁸³ The Matlab implementation of Bayesian Binning Quantiles can be accessed as indicated in the original paper. Note that this approach,

unlike monotonic (i.e., order-preserving) transformations such as Platt scaling and isotonic regression, does not require or guarantee that the order of estimates to remain the same after the application of the calibration model, so a decrease or increase in discrimination after the application of such calibration models can occur.

By applying Platt scaling, isotonic regression, and BBQ to test set estimates produced by the SVM model, the estimates became better calibrated, as shown by the Spiegelhalter's Z-test and H-L test results in **Table 3.8** Application of the calibration models also lowered ECE, MCE, and ICI, which indicates estimates exhibit better calibration than those from the original SVM. Finally, Cox's slope and intercept became closer to 1 and 0, respectively. That is, there was consistency among most calibration measures in that estimates obtained by calibration models resulted in smaller errors than the ones calculated for the original SVM estimates.

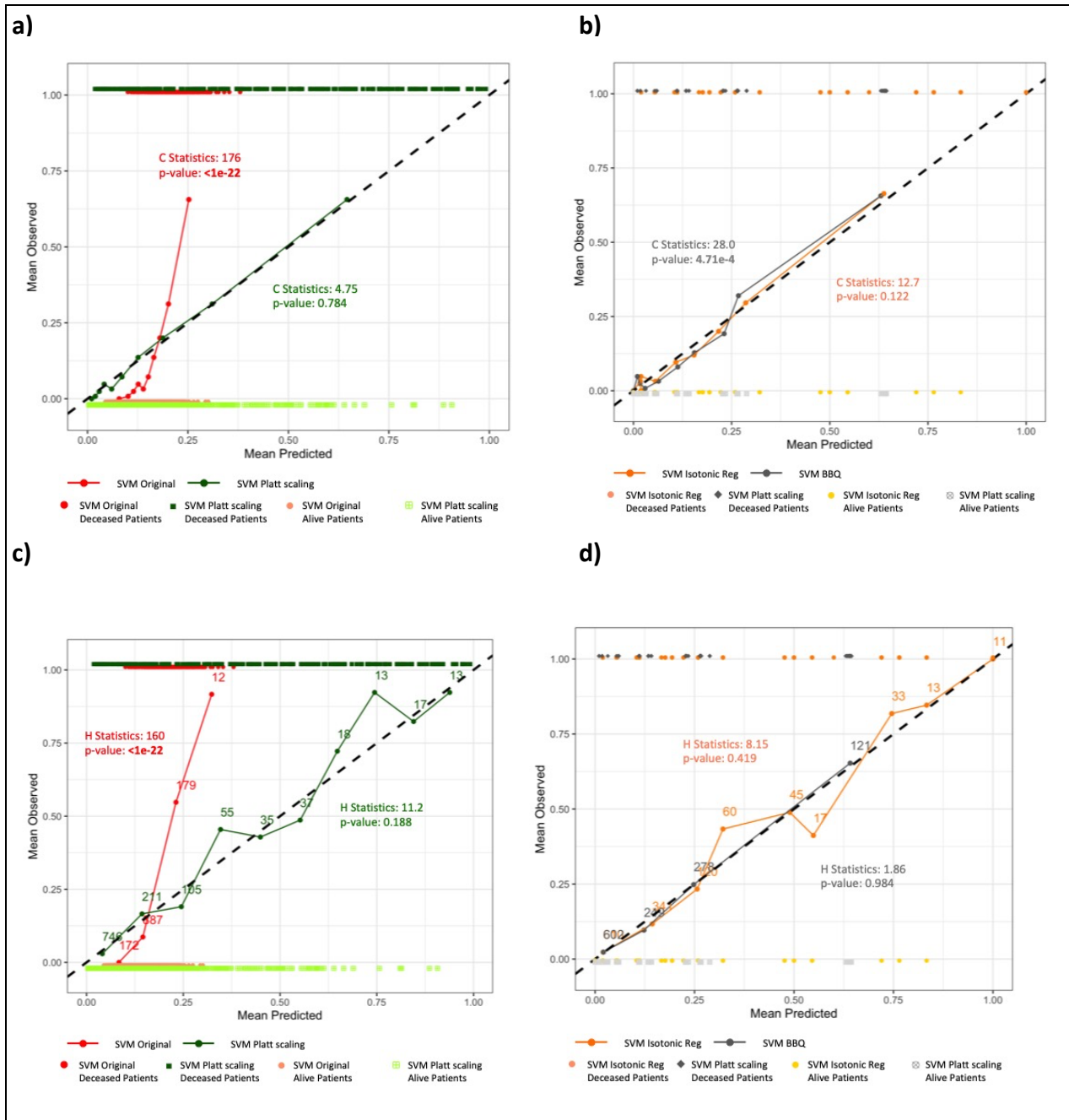


Figure 3.5 Reliability Diagrams of Test Set Estimates Produced by Support Vector Machine (SVM) Models After application with Platt scaling, Isotonic Regression, or Bayesian Binning Quantiles for Simulated Data

Estimates produced by the models and their actual binary outcomes are plotted to show the distribution of the actual data. The *Alive* outcome is indicated as 0 and *Deceased* as 1. Corresponding H-L statistics and p -values are shown in the graphs. **a) Platt scaling** and **b) isotonic regression and BBQ** are grouped for the calculation of the H-L C statistic. The number of patients within each bin is the same (125). **c) and d)** are grouped for the calculation of the H-L H statistic. After applying calibration models, the SVM estimates show proper calibration.

An unintended consequence of applying calibration models can be the worsening of calibration for models that are already well-calibrated. **Table 3.8** also shows results after the use of calibration models to the Test set estimates produced by the LR model, which were already well-calibrated, as shown previously in **Figures 3.3a** and **3.3b**. After using Platt scaling, H-L C or H-L H statistics returned significant p-values, while the Spiegelhalter Z-test did not. Looking at the corresponding reliability diagrams in **Figure 3.6**, the lines with applications of Platt scaling show more deviation from the diagonal line than the original LR line. Such a phenomenon sometimes happens with Platt scaling, since its underlying assumption is that the estimates' distribution of 'true' probabilities is sigmoidal in shape.⁸⁴ When the logistic parametric assumptions are not met, properly calibrated estimates could become improperly calibrated. Application of isotonic regression and BBQ on the LR model also raised the H-L C and H statistics, indicating worsened calibration. This result is consistent with the increase in ECE, MCE, and ICI for LR calibrated with BBQ, and consistent with the increase in ICI for LR calibrated with isotonic regression.

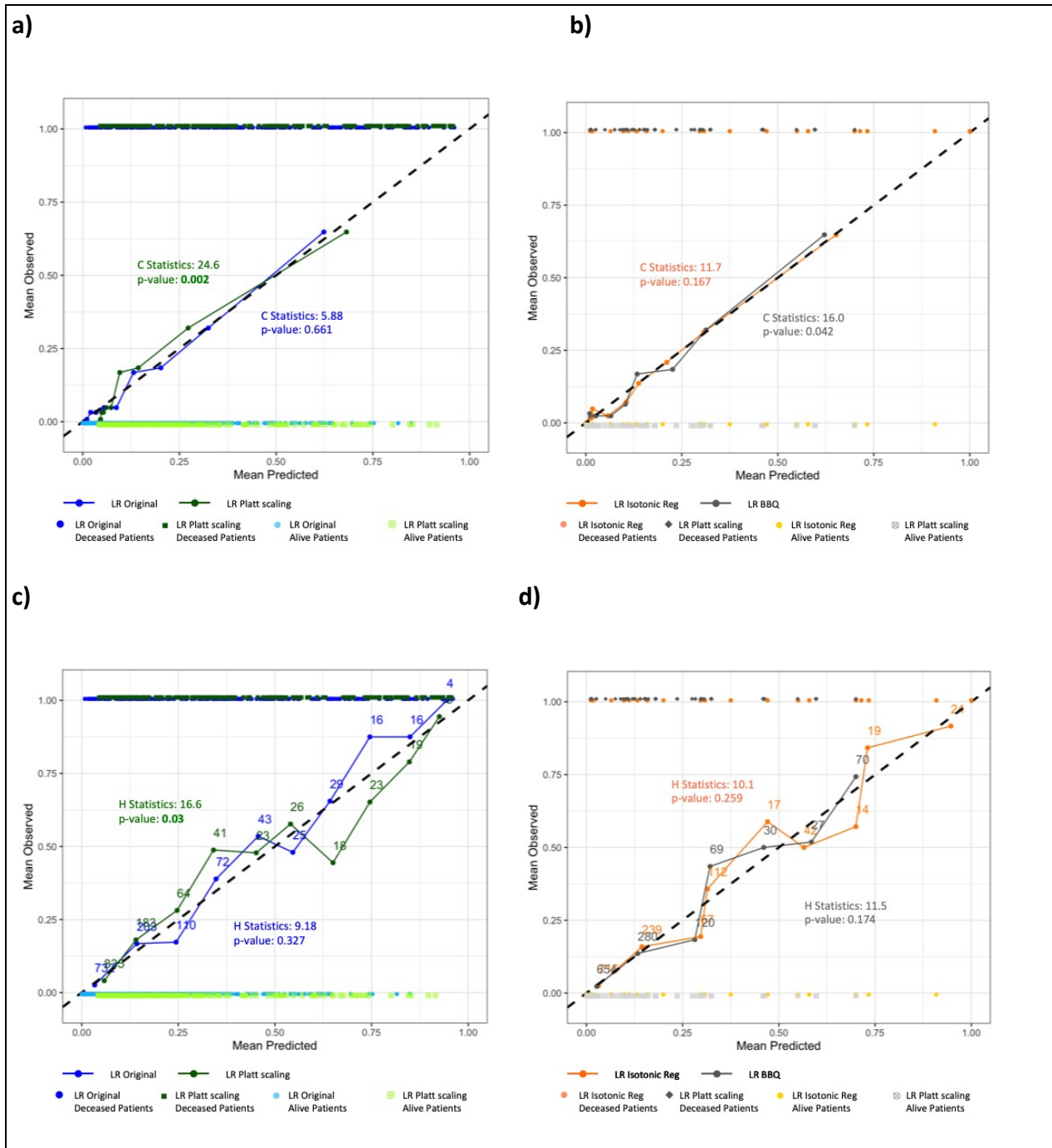


Figure 3.6 Reliability Diagrams of Test Set Estimates Produced by Logistic Regression (LR) Models After Application of Platt Scaling, Isotonic Regression, or Bayesian Binning Quantiles for Simulated Data

Data points of estimates produced by the models and their actual binary outcomes are plotted to show the distribution of the actual data. The *Alive* outcome is indicated as 0 and *Deceased* as 1. Corresponding H-L statistics and p-values are shown in the graphs. **a)** and **b)** are grouped for the calculation of the H-L C statistic. The number of patients within each bin is the same (125). **c)** and **d)** are grouped for the calculation of the H-L H statistic.

Table 3.8 Discrimination and Calibration Results of the Logistic Regression (LR) and Support Vector Machine (SVM) Models Applied to the Test Set.

Discrimination is measured by the Area Under the Operating Characteristic Curve (AUROC). The Brier score is a combined measure of discrimination and calibration. Calibration is measured by the Spiegelhalter's Z-test, Average Absolute Error, H-L test, the maximum calibration error (MCE), the expected calibration error (ECE), Cox's slope and intercept, and the Integrated Calibration Index (ICI). SVM estimates for the Test set produced were improperly calibrated. Application of Platt scaling, isotonic regression, or BBQ was performed. Bolded values show significant results.

	LR	LR Platt scaling	LR Isotonic Regression	LR BBQ	SVM	SVM Platt scaling	SVM Isotonic Regression	SVM BBQ
AUROC	0.870	0.870	0.870	0.867	0.870	0.870	0.870	0.862
Brier	0.087	0.088	0.088	0.089	0.111	0.086	0.088	0.090
Spiegelhalter Z Score (p-value)	0.762 (0.223)	0.417 (0.338)	0.087 (0.465)	0.748 (0.227)	2.214 (0.013)	0.826 (0.204)	0.693 (0.244)	0.731 (0.232)
Average Absolute Error	0.177	0.177	0.177	0.182	0.236	0.177	0.177	0.185
H-L C Statistics (p-value)	5.886 (0.661)	24.6 (0.002)	11.7 (0.167)	16.194 (0.042)	176.482 (<1e-22)	4.751 (0.784)	12.736 (0.122)	28.034 (4.71e-4)
H-L H Statistics (p-value)	9.185 (0.327)	16.632 (0.03)	10.126 (0.259)	11.532 (0.174)	160.274 (<1e-22)	11.264 (0.188)	8.154 (0.419)	1.868 (0.984)
MCE	0.038	0.072	0.033	0.042	0.403	0.028	0.034	0.052
ECE	0.014	0.035	0.012	0.022	0.109	0.011	0.018	0.027
Cox's Slope	1.070	1.074	0.953	1.020	5.014	1.087	1.023	1.008
Cox's Intercept	0.080	0.072	-0.092	-0.007	6.193	0.081	-0.001	-0.02
ICI	0.010	0.034	0.012	0.012	0.104	0.008	0.013	0.020

Table 3.9 Summary of Packages in R Used to Evaluate Calibration and to apply calibration models on Estimates.

Calibration Measure	R package	R commands
H-L Test	ResourceSelection ⁶⁹	hoslem.test
Reliability Diagram	PresenceAbsence ⁷⁰	calibration.plot
Spiegelhalter’s Z-test	stats	val.prob
Brier Score	Rms mDescTools	val.prob BrierScore
Recalibration Method	R package	R commands
Platt scaling	stats	glm*
Isotonic Regression	stats	isoreg

*link function ‘logit’ has to be used for glm

3.7 Real Clinical Data

The simulated experiments were repeated with real data set from the National Inpatient Sample (NIS). I picked 10,000 random patients from the NIS 2014 dataset and predicted whether patients would need a major therapeutic procedure during their stay, (20% did). The predictors were pre-admission features (age, sex, race, admission month, elective or non-elective admission, expected primary payer, median household income quartile range, and presence or absence of 30 chronic conditions). Experiments were done with LR and SVM with a radial kernel.

The results are shown in **Table 3.10**. H-L C and H-L H statistics, and Spiegelhalter’s Z-test showed significance (i.e., inadequate calibration) for both LR and SVM. For LR, Platt scaling was not able to produce calibrated estimates, as shown by the H-L test, ECE, MCE, and ICI. However,

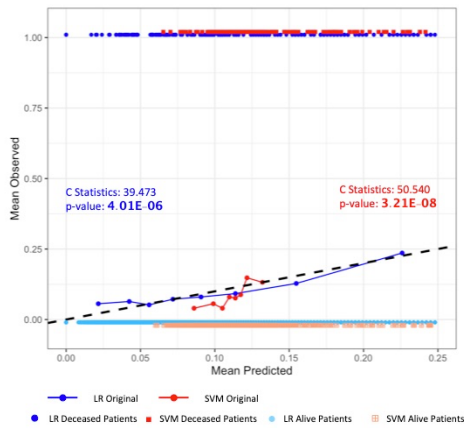
Cox's slope and intercept suggest that calibration improved. For SVM, Platt scaling produced statistically significant H-L C statistics and H-L H statistics, and elevated ECE, MCE, and ICI. Cox's slope and intercept also suggest worse calibration than that of LR. Isotonic Regression and BBQ were able to improve calibration in both LR and SVM, resulting in non-significant p-values for the Spiegelhalter's Z and H-L tests. They also lowered MCE, ECE, and ICI. Cox's slope and intercept became closer to 1 and 0, respectively, for both LR and SVM. All related reliability diagrams are shown in **Figures 3.7-3.9**.

Table 3.10 Discrimination and Calibration Results of the Logistic Regression (LR) and Support Vector Machine (SVM) Models Applied to the NIS Test Dataset.

Discrimination is measured by the Area under the Operating Characteristic Curve (AUROC), while calibration is measured by the Spiegelhalter’s Z-test, H-L test, maximum calibration error (MCE), expected calibration error (ECE), Cox’s slope and intercept, and the Integrated Calibration Index (ICI). Estimates for the Test set produced by both LR and SVM were improperly calibrated. Application of Platt scaling, isotonic regression or BBQ was performed. Bolded values show significant results.

	LR	LR Platt scaling	LR Isotonic Regression	LR BBQ	SVM	SVM Platt scaling	SVM Isotonic Regression	SVM BBQ
AUROC	0.785	0.785	0.785	0.787	0.817	0.817	0.817	0.817
Brier	0.119	0.121	0.118	0.120	0.109	0.110	0.104	0.105
Spiegelhalter Z Score (p-value)	1.895 (0.029)	-0.081 (0.468)	0.316 (0.376)	-0.246 (0.402)	-1.698 (0.044)	-0.064 (0.542)	0.175 (0.351)	-0.383 (0.313)
Average Absolute Error	0.230	0.241	0.234	0.240	0.221	0.227	0.213	0.217
H-L C Statistics (p-value)	39.473 (4.01E-06)	43.439 (7.26E-07)	13.7447 (0.111)	12.351 (0.136)	50.540 (3.21E-08)	67.228 (1.746e-11)	10.760 (0.215)	10.865 (0.209)
H-L H Statistics (p-value)	38.084 (7.263e-06)	61.353 (2.527e-10)	10.456 (0.234)	6.683 (0.571)	146.129 (1<e-22)	69.947 (5.036e-12)	9.556 (0.297)	8.804 (0.359)
MCE	0.119	0.124	0.061	0.069	0.096	0.133	0.061	0.060
ECE	0.031	0.043	0.025	0.022	0.044	0.050	0.016	0.017
Cox’s Slope	0.560	0.946	0.889	0.923	0.863	1.001	0.902	1.019
Cox’s Intercept	-0.601	-0.177	-0.230	-0.203	-0.374	-0.149	-0.257	-0.123
ICI	0.027	0.038	0.018	0.025	0.052	0.061	0.015	0.015

a)



b)

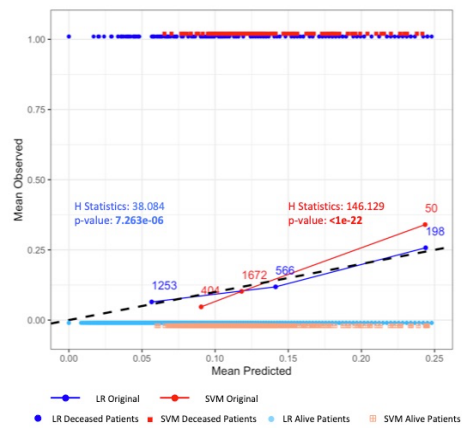
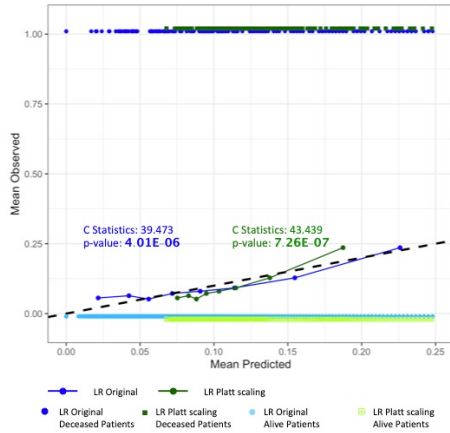


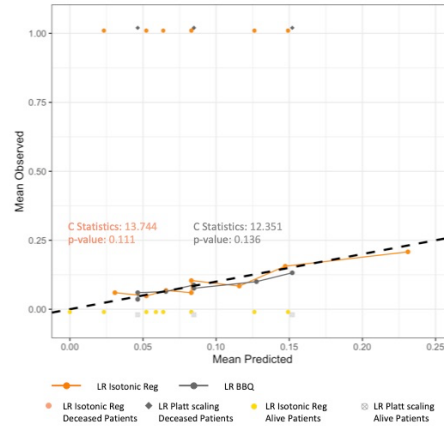
Figure 3.7. Reliability diagrams of test set estimates produced by Logistic Regression (LR) and Support Vector Machine (SVM) models for NIS dataset

Data points of estimates produced by the models and their actual binary outcomes are plotted to show the distribution of the actual data. Note that there were no groups with average estimates above 0.25. No major therapeutic procedure during stay is indicated as 0 and major therapeutic procedure during stay as 1. Corresponding Hosmer-Lemeshow statistics and p-values are shown in the graphs. **a)** Grouped for the calculation of the H-L C statistic. The number of patients within each bin is the same (250). **b)** Grouped for the calculation of the H-L H statistic. The numbers of patients are labeled in the graph for **b)**.

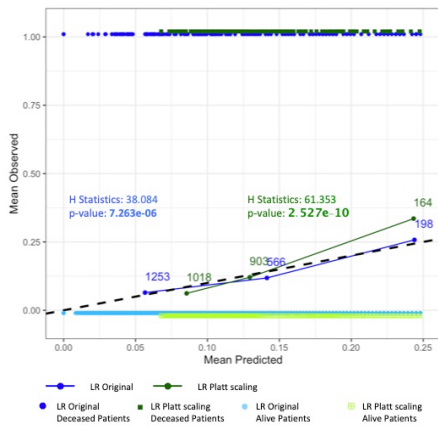
a)



b)



c)



d)

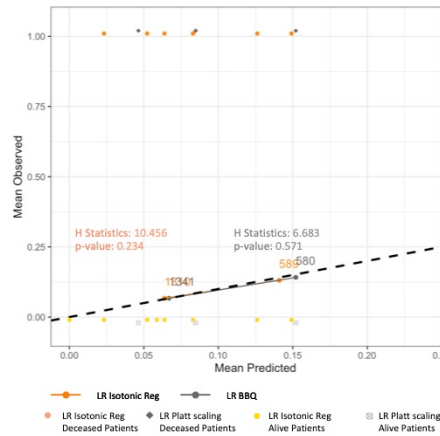
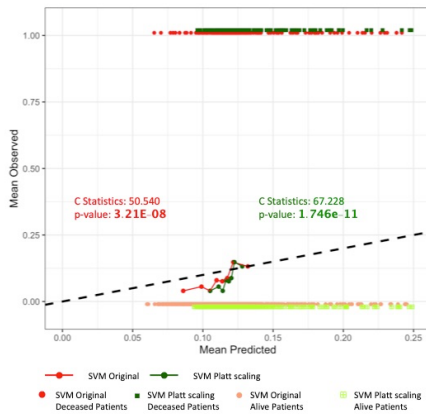


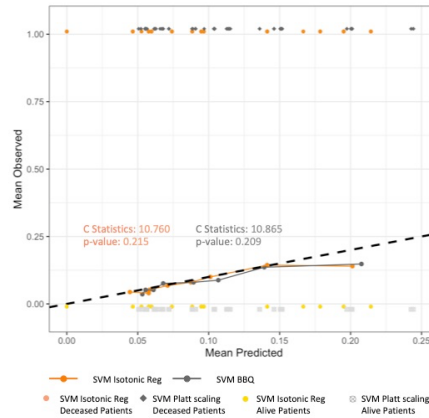
Figure 3.8 Reliability diagrams of test set estimates produced by Logistic Regression (LR) models recalibrated with Platt scaling, Isotonic regression, or BBQ for NIS dataset

Data points of estimates produced by the models and their actual binary outcomes are plotted to show the distribution of the actual data. No major therapeutic procedure during a stay is indicated as 0 and a major therapeutic procedure during a stay as 1. Corresponding Hosmer-Lemeshow statistics and p-values are shown in the graphs. a) and b) are grouped for the calculation of the H-L C statistic. The number of patients within each bin is the same (250). c) and d) are grouped for the calculation of the H-L H statistics. After recalibration with isotonic regression or BBQ, the LR estimates show proper calibration.

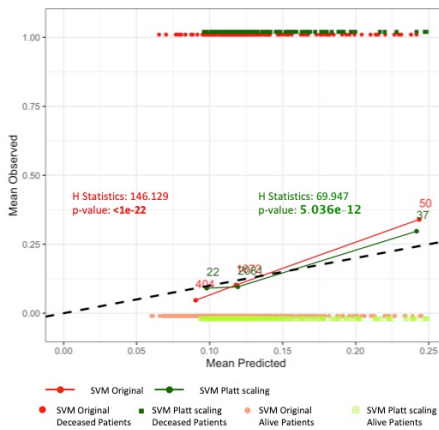
a)



b)



c)



d)

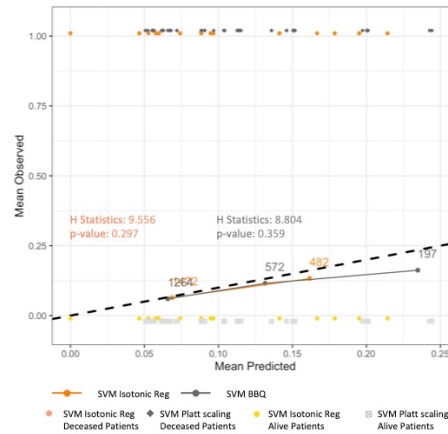


Figure 3.9. Reliability diagrams of test set estimates produced by Support Vector Machine (SVM) model recalibrated with Platt scaling, Isotonic regression, or BBQ, grouped for the H-L C statistics and the H-L H statistics for NIS dataset

Data points of estimates produced by the models and their actual binary outcomes are plotted to show the distribution of the actual data. No major therapeutic procedure during a stay is indicated as 0 and a major therapeutic procedure during a stay as 1. Corresponding Hosmer-Lemeshow statistics and p-values are shown in the graphs. **a)** and **b)** are grouped for the calculation of the H-L C statistic. The number of patients within each bin is the same (250). **c)** and **d)** are grouped for the calculation of the H-L H statistics. After recalibration with isotonic regression or BBQ, the SVM estimates show proper calibration.

3.8 Discussion

The H-L test continues to be a very well-known proxy for a calibration measure. From 2014 to 2019, there have been over 874 mentions on Pubmed. The H-L test is a starting point to measuring calibration and needs to be considered. It has shortcomings, however, including the susceptibility to increase in statistical power as sample size increases and the arbitrariness of the number of groups. The H-L test's probability of rejecting a poorly fitted model increases as the sample size increases. To remedy the problem, *Paul et al.* created a function to calculate the number of groups according to sample size.⁸⁵ The formula was able to keep the power consistent as the sample size increased, but it could only handle sample sizes <25,000. For larger sample sizes, more complex techniques have been proposed.^{86,87} As for the arbitrariness of the number of groups, it is a problem shared by other measures. MCE, ECE, and reliability diagrams all require grouping. The Loess function in ICI also requires an adjustable window parameter in order to calculate calibration. In model comparison, this is not a huge problem, as one can use the same test set and the same number of groups when comparing the calibration of different models. It is a problem when asserting whether model estimates are well-calibrated, as changing the number of groups can alter the p-values.

Calibration measurements can be categorized into two groups. The H-L test, ECE, and MCE take an approach that requires grouping based on estimates. Cox intercept and slope, and ICI regress the estimates to the true outcomes. In terms of interpretability, the grouping methods are more readily understandable by the medical community. More recent methods to measure calibration are increasingly being used, and new guidelines on how to assess whether they are

adequate for a particular use case will develop over time. I summarized the pros and cons of the methods I have presented in this Chapter in **Table 3.11**.

Table 3.11 Summary of Advantages and Disadvantages of Calibration Measurement Methods Presented in this Chapter

Calibration measure (examples of studies in which the measure was used)	Pros	Cons
Brier Score ⁽⁸⁸⁻⁹⁰⁾	Easy calculation. Measures a combination of discrimination and calibration.	The contribution of each component (discrimination, calibration) is not easy to calculate or interpret.
Spiegelhalter's Z-Test ^(91,92)	Extension of Brier score that measures calibration only. P-value can serve as a guide for how calibrated a model is.	Not intuitive.
Average Absolute Error	Easy calculation. Intuitive.	Same problems as Brier Score. Rarely used.
Hosmer-Lemeshow Test ^(93,94)	Widely used in the biomedical literature. P-value can serve as a guide for how calibrated a model is.	Not designed to handle sample sizes >25,000. Use of HL-C and HL-H can result in different significance.
Reliability Diagram ^(72,73,95,96)	Allows for visualization of regions of miscalibration and the "direction" of miscalibration (i.e., underestimation, overestimation)	Not a continuous graph. Hard to see when estimates are clustered in certain regions (zoom into a portion of the graph may be needed).
Expected Calibration Error and Maximum Calibration Error ^(72,97,98)	Intuitive.	No statistical test to help determine whether a model is adequately calibrated or not.
Cox's Slope and Intercept ⁽⁹⁹⁻¹⁰¹⁾	Summarizes the direction of miscalibration (i.e., overall underestimation or overestimation).	Can still result in perfect calibration even if regions are miscalibrated.
Integrated Calibration Index	Can capture regions of miscalibration that Cox's slope and intercept cannot.	Requires Loess to build the calibration model. Not intuitive.

In our simple example, application of Platt scaling and isotonic regression on the SVM-derived estimates had good results. While there are other methods that were built upon such techniques, and more are being created for modern deep learning,^{71–73,84,102–104} Platt scaling and isotonic regression are relatively easy to understand and implement. They can act as the benchmark for subsequent calibration models to be compared to. While ease of interpretation is the main advantage of these two techniques, they are not without faults. As illustrated in our example, Platt scaling may fail when models are already well-calibrated. It performs best under the assumption that the estimates are close to the midpoint and away from the extremes.⁸⁴ Therefore, Platt scaling may not be suitable for estimates produced by Naive Bayes or Adaboost models, which tend to produce extreme estimates close to 0 and 1.¹⁰⁵ In terms of isotonic regression, the criticism is that it lacks continuousness. Since the fitted regression function is a piecewise function, a slight change in the uncalibrated estimates can result in dramatic differences in the recalibrated estimates (i.e., a change in step). Also, due to the stepwise nature of the function, uncalibrated estimates that fall on the same ‘step’ end up having the same calibrated value, eliminating any distinction between those patient estimates (similarly to quantile binning). However, there are smoothing techniques to make the isotonic regression recalibrated estimates continuous.¹⁰⁶

Finally, the available packages currently used to measure calibration are sparse and missing some key documentation. There is a need for better descriptions of how such techniques were implemented.

3.9 Conclusion

While discrimination is the most commonly used measure of how well a predictive model

performs, calibration of estimates is also important, particularly when predictions are used for individual patients (e.g., precision medicine). With the help of R packages, it is not difficult to measure calibration alongside discrimination when reporting on a model's predictive performance. Also, there are simple techniques that can improve calibration without the need to retrain a model. To improve discrimination, parameters need to be tuned or a completely different model may be required, whereas to improve calibration there are techniques that do not require retraining. In this Chapter, I raised awareness of calibration measures and calibration models in clinical predictive modeling, providing simple and readily reproducible examples. In the next Chapter, I will show how calibration can be measured and improved in a distributed setting using novel methods.

Chapter 3, in part, has been accepted for publication of the material as it may appear in *A Tutorial on Calibration Measurements and Calibration Models for Clinical Prediction Models*. by Huang, Yingxiang; Li, Wentao; Macheret, Fima; Gabriel, Rodney; Ohno-Machado, Lucila. Journal of the American Medical Informatics Association, 2020. The dissertation author was the primary investigator and author of this paper.

4. Calibration Measurement and Calibration Model in a Federated Manner

4.1 Overview

Existing calibration models, i.e., models that improve the calibration of estimates derived from predictive models, have been shown to improve calibration.^{64,68,73,82,95,105} Such models are often post-processing models where estimates produced by a predictive model and the true outcomes are used to build calibration models. However, the constraint of building calibration models locally may limit the degree of improvement. The process of building and evaluating a predictive model typically involves iterations in which different variables are considered, as well as tuning of parameters. While a plethora of articles have been written about building robust predictive models in a distributed setting (see for example some methods in references [xxx] and our innovative embedding-based method in Chapter 2), none have been proposed to build calibration models in a distributed setting. In Chapter 3 I explained why calibration is a critical component of clinical predictive model evaluation. Here I propose a Secure Multiparty Computation (SMC) method to build a global isotonic regression calibration model using data from a few medical systems, without sharing patient-level data. To show that the global model outperforms local calibration models and keep the principle of not sharing the individual-level data, I also propose measuring calibration in a distributed manner.

4.2 Calibration Models

As explained in the Introduction Chapter, gathering large samples is difficult. Recently, there have been efforts to build high-quality centralized models with training data distributed over a large number of sites. Some existing examples and our new method were described in Chapter 2 of this dissertation. Although logistic regression are generally well-calibrated, but this is not necessarily true and other models, like SVM and Naive Bayes, are often not well calibrated.⁶⁴ Random Forests and Neural Networks are also prone to improper calibration.^{64,72} Calibration models are therefore needed. Furthermore, it is unclear whether calibration models that operate in a federated, distributed manner result in higher-quality models than local calibration models.

Building calibration models in a way that protects privacy requires hiding sensitive data such as estimates and true outcomes. Sharing estimates and true outcomes can be detrimental to both institutional privacy and patient privacy. In terms of institutional privacy, sharing actual outcomes of sensitive information could put institutions at a disadvantage with respect to their competitors.^{107,108} Such sensitive information could include specific mortality rates, healthcare-associated infections rates, unplanned readmissions, etc. While sharing these data does not violate patient privacy, medical systems may be reluctant to share such data due to concerns over leakage of sensitive information about the institution. In terms of patient privacy, sharing the outcomes of a predictive model could even help identify the patient, for example, when the outcome is unique to a patient in a certain medical system. Due to these reasons, it is attractive for medical systems to have the ability to create the best calibration model while keeping data private.

4.3 Methodology

I chose to extend the Smooth Isotonic Regression (SIR) model in a federated manner. SIR is done by first training an isotonic regression model by optimizing the isotonic regression function as shown in Equation (3.9). Isotonic regression is used as a calibration model, but due to the stepwise nature of the function, there is room for improvement. Therefore, a smoothing technique can be implemented in the SIR model and the isotonic regression function can be solved with either the Pool-Adjacent-Violators Algorithm (PAVA) or the active set method,¹⁰⁹ resulting in a monotonically increasing step function. For SIR, a point is randomly chosen from each 'step' (flat region of the isotonic regression function) and a Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)¹¹⁰ function is used to obtain a smooth monotonically increasing function as the final calibration model. A PCHIP function is a third-degree polynomial spline function that interpolates a function based on the existing datasets. In the case of an isotonic function, each flat region and the randomly chosen point act as the existing datasets for the interpolation.

An example of the resulting functions is given in **Figure 4.1**. The 'Data' points (circles) are plotted as the estimates vs. outcomes. An isotonic and a smooth isotonic function are fitted according to the 'Data.' Recalibration is done with the application of isotonic function or SIR on uncalibrated estimates.

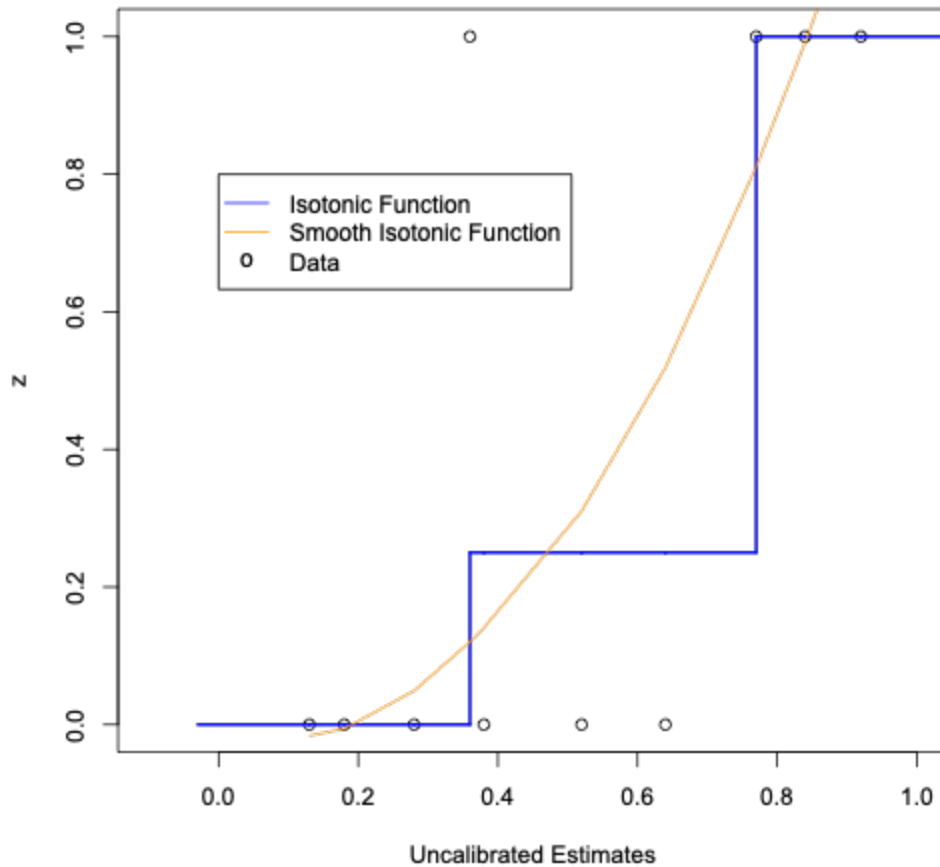


Figure 4.1 Example of Isotonic and Smooth Isotonic Functions

'Data' points are the predicted estimates vs. true outcomes. Isotonic and smooth isotonic functions are derived according to the 'Data.'


To obtain the smooth isotonic function, I first need to derive the isotonic function. The isotonic function is a least square problem with constraints. The numerical data for **Figure 4.1** and the overall process of deriving the isotonic function are shown in **Figure 4.2**. There are 4 phases to the algorithm:

- 1) sort instances by estimates;
- 2) find an isotonic solution;

- 3) find the breakpoint of isotonic steps and derive a new isotonic function; and
- 4) recursively derive optimal solution given the constraints at each isotonic step until the final optimal solution is found.

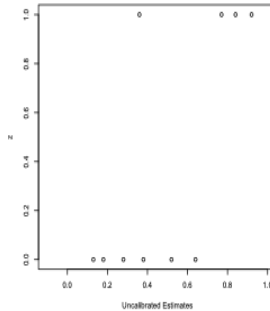
Mtx 1


ID	1	2	3	4	5	6	7	8	9	10
Est.	0.77	0.84	0.92	0.52	0.64	0.36	0.38	0.13	0.18	0.28
y	1	1	1	0	0	1	0	0	0	0

Phase (1)  Sort according to the estimates

Mtx 2

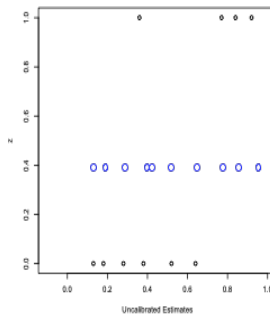
ID	8	9	10	6	7	4	5	1	2	3
Est.	0.13	0.18	0.28	0.36	0.38	0.52	0.64	0.77	0.84	0.92
y	0	0	0	1	0	0	0	1	1	1




Phase (2)  Find isotonic function (z)

Mtx 3

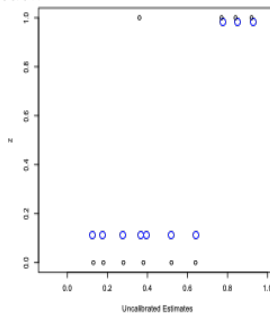
ID	8	9	10	6	7	4	5	1	2	3
Est.	0.13	0.18	0.28	0.36	0.38	0.52	0.64	0.77	0.84	0.92
y	0	0	0	1	0	0	0	1	1	1
z	4/10	4/10	4/10	4/10	4/10	4/10	4/10	4/10	4/10	4/10




Phase (3)  Find breakpoint of isotonic steps and derive new isotonic function (z)

Mtx 4

ID	8	9	10	6	7	4	5	1	2	3
Est.	0.13	0.18	0.28	0.36	0.38	0.52	0.64	0.77	0.84	0.92
y	0	0	0	1	0	0	0	1	1	1
z	1/7	1/7	1/7	1/7	1/7	1/7	1/7	1	1	1



Phase (4)  Recursively derive isotonic function for each step

Mtx 5

ID	8	9	10	6	7	4	5	1	2	3
Est.	0.13	0.18	0.28	0.36	0.38	0.52	0.64	0.77	0.84	0.92
y	0	0	0	1	0	0	0	1	1	1
z	0	0	0	1/4	1/4	1/4	1/4	1	1	1

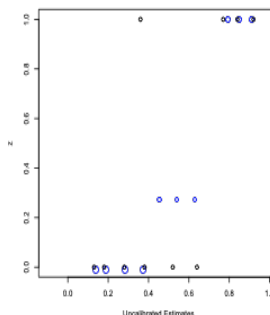


Figure 4.2 Process for Deriving the Isotonic Function

The estimates are first sorted. Then breakpoints of the steps in the isotonic functions are located, and the isotonic function is calculated with the average of positive cases within each step.

Figure 4.2 shows an overview of how to derive the isotonic function. More formally, isotonic regression for our small example with 10 constraints is defined in Equation (4.2),

$$f(z) = \sum_{i=1}^{10} (y_i - z_i)^2 \rightarrow \min!$$

constraints :

$$z_1 \leq z_2$$

$$z_2 \leq z_3$$

$$z_3 \leq z_4$$

$$(4.2) \quad \ddots \leq z_{10}$$

where y_i are the binary outcomes and z_i are the final values. In our example, there are 10 inequality constraints (\leq), forcing the final solution to be monotonically increasing. After Phase 1 where the instances are sorted by estimates, this optimization problem can be solved by the active set method.

The active set method begins by setting all constraints to equality constraints as follows in Equation (4.3),

$$f(z) = \sum_{i=1}^{10} (y_i - z_i)^2 \rightarrow \min!$$

constraints :

$$z_1 = z_2$$

$$z_2 = z_3$$

$$z_3 = z_4$$

$$(4.3) \quad \ddots = z_{10}$$

Under such constraints, the optimal solutions for z are the fraction of the total number of cases with the outcome '1' as shown in **Figure 4.2** Mtx 3 row z .

To calculate whether the optimal solutions can be improved, the active set method then utilizes Lagrange multipliers. Lagrange multipliers are used to find the local minima of the isotonic function, according to Equation (4.4).

$$(4.4) \quad L(z, y, \lambda_1, \lambda_2 \dots \lambda_{10}) = \sum_{i=1}^{10} (y_i - z_i)^2 + \lambda_1(z_1 - z_2) + \lambda_2(z_2 - z_3) \dots + \lambda_{10}(z_9 - z_{10})$$

The value of the Lagrange multiplier or λ at the solution of the problem is equal to the rate of change in the minimum value of Equation (4.3), as the constraint is ‘relaxed’ from equality to inequality constraints. The most negative λ indicates the breakpoint of the isotonic step in Phase 3 of **Figure 4.2**. I am looking for negative λ because I am minimizing the isotonic function. Therefore, I want to ‘relax’ the constraint that would lower the minimum value of the equation (4.3). For our example, λ_7 was the smallest λ , so the breakpoint of the isotonic step in Phase 3 of **Figure 4.2** is set at the seventh instance. The optimal solution is calculated for each step where z are equal to the fraction of cases with outcome ‘1’ at each step, as shown in row z of Mtx 4 in **Figure 4.2**. The process is repeated for each step of the isotonic function, recursively, until there are no more negative λ 's.

The optimal solution, given the constraints for isotonic regression, is equal to the average positive cases in Equation (4.5),

$$(4.5) \quad z = \left\{ \frac{1}{n} \sum_i^n y_i \right\}_n$$

where y_i are the observed outcomes and n is the number of instances. At the start of the algorithm, there are no breakpoints, therefore the solution, or z , is a vector of length n , and the value is the proportion of positive cases. Phase 3 in **Figure 4.2** computes the Lagrange multipliers, λ , and finds the minimum λ , which is generally difficult when data involves continuous values for the observed outcomes, but it becomes easier when applied to calibration where the observed outcomes are either 0 or 1. The Lagrange multipliers are calculated as in Equation (4.6).

$$(4.6) \quad \lambda = M \times c$$

Where

$$M = \begin{bmatrix} -\frac{n-1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ -\frac{n-2}{n} & -\frac{n-2}{n} & \frac{2}{n} & \frac{2}{n} & \cdots & \frac{2}{n} \\ -\frac{n-3}{n} & -\frac{n-3}{n} & -\frac{n-3}{n} & \frac{3}{n} & \cdots & \frac{3}{n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -\frac{n-n}{n} & -\frac{n-n}{n} & -\frac{n-n}{n} & -\frac{n-n}{n} & \cdots & -\frac{n-n}{n} \end{bmatrix}$$

And

$$c = z - y.$$

M is the QR decomposition (orthogonal matrix and upper triangular matrix) derived according to the constraints. A visual example is given in **Figure 4.3**.

To evaluate the calibration of model estimates before and after transforming the estimates using a calibration model, the measurement of calibration can also be done distributedly. I describe the distributed calculation of the Hosmer-Lemeshow (H-L) test, the Expected Calibration Error (ECE), and the Maximum Calibration Error (MCE) that were previously described in Chapter 3.

4.4 Secure Multiparty Computation (SMC) Sorting

For both SIR and calibration evaluations, I first need to sort estimates and then I compare them to the true outcomes. This is trivial, except that I need to sort estimates that are local to each site, without sharing the estimates or the true outcomes across sites. That is, the estimates and true outcomes stay local, as did the patient data in Chapter 2, and only sums or other aggregate statistics are shared. I therefore need a sorting and counting methods for a distributed data setting. While there are data-oblivious sorting algorithms,^{111–113} they require setting up cryptographic primitives and a secret-sharing scheme where secured addition, multiplication, compare-and-swap, and etc. are assumed. While efficient, such operations are hard to implement and interpret, often treated as black-boxes in the cited algorithms. Here I propose a peer-to-peer privacy-preserving SMC radix sort algorithm that assumes no such complex operations, allowing for easy reproduction.

Our federated radix sort algorithm is based on the most significant digit (MSD) radix sort. Sorting starts at the first MSD. Using Algorithm 1 from *Wu et al.*,¹⁰⁸ I first collect the sum of each digit 0-9 at the first MSD from all the sites. In our cases, since estimates range from (0, 1), this is the first digit after the decimal point (e.g., '9' in 0.923, '7' in 0.737, etc). Algorithm 1 from *Wu et al.* begins from a central server that does not contribute data, but sends a vector of random

values that is passed from one local site to the next. An example is given in **Figure 4.4**. Each local site adds its count of each number (0-9) at the MSD digit to the random vector (**Figure 4.4 a**). The purpose of the random vector is to mask values contributed by the first site so that information is not disclosed to the second site receiving the vector, and so on. Once the random vector has accumulated total counts of each number at the current digit from all the sites (**Figure 4.4 b-d**) and returns this sum to the central server, the original random numbers are then subtracted (**Figure 4.4 e**). This allows the central server to rank all estimates without knowing which site contributed which values by calculating a counting sort matrix (**Figure 4.4 f**). This counting sort matrix has the temporary rank of each estimate. By distributing the 'Rank' row of the counting sort matrix to each site, each site can update the temporary rank of its observations in terms of the digit under study. **Figure 4.4** shows the result of a toy example with three sites.

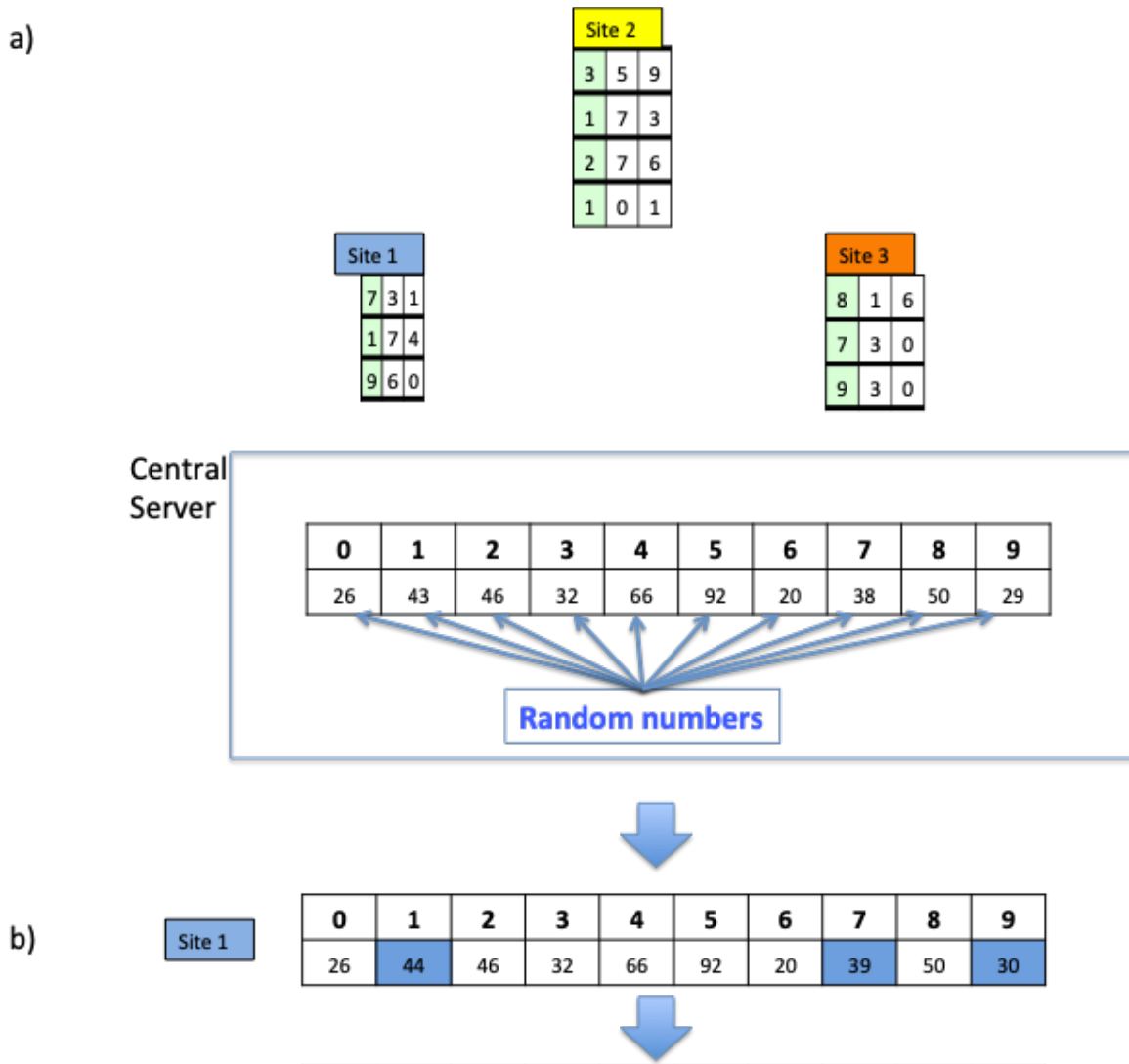
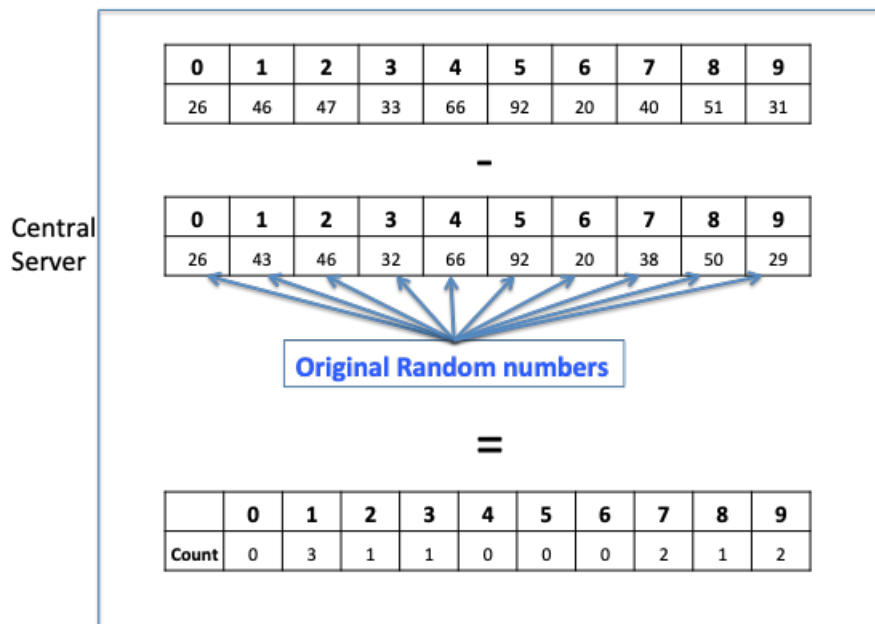
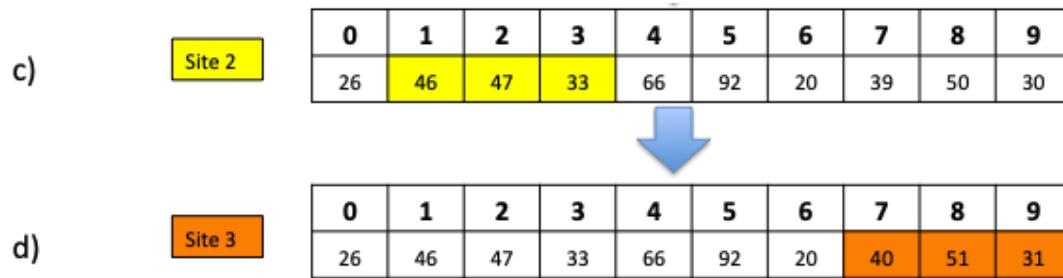


Figure 4.4 The First Iteration of Federated Radix Sorting

Temporary ranks are calculated after one iteration of secure multiparty computation radix Sorting using Algorithm 1 from Wu et al. (a) First, a random vector of numbers is created in the central server. The random vector is passed to Site 1, (b) which adds counts of each number at the MSD. Site 1 then passes the sums to Site2, (c) which does the same and passes to Site 3, (d) which does the same. Finally, (e) the random vector is returned to the central server and the original random vector is subtracted. To derive the Counting Sort Matrix, the row Rank can be derived by first cumulatively adding the counts of the digits to derive the Cumulative row, then by shifting the Cumulative row one digit over and appending 0 to the beginning. The columns highlighted in yellow in the counting sort matrix are the digits that have more than 0 counts. The count Sort Matrix is passed from site to site in a sequential manner as shown in (f). Each site updates its ranks according to the 'Rank' row of the Counting Sort Matrix.



e)

Figure 4.4 The First Iteration of Federated Radix Sorting, continued

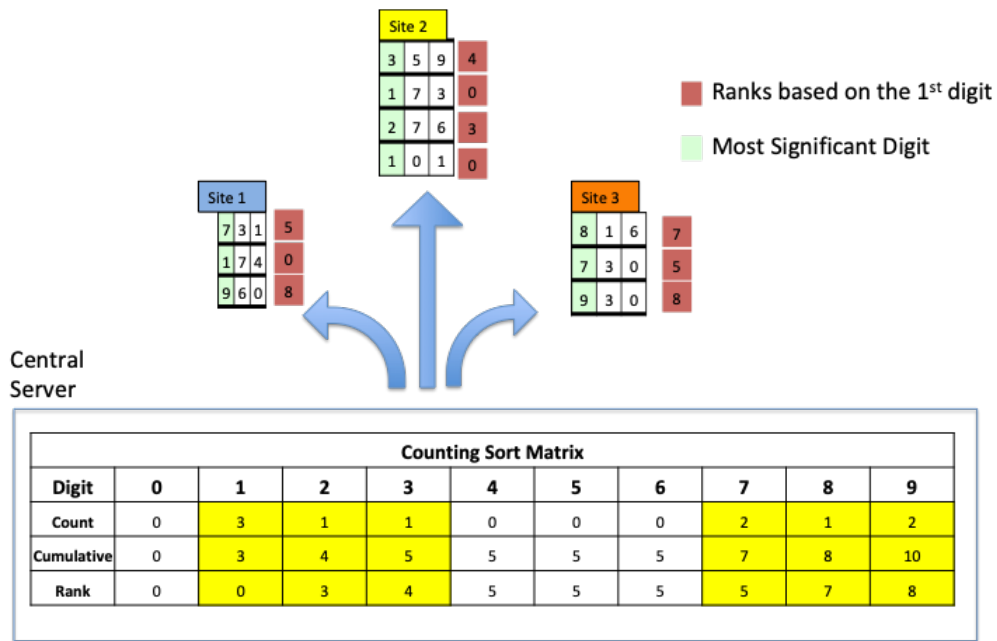


Figure 4.4 The First Iteration of Federated Radix Sorting, continued

I then move on to the second MSD. A counting sort matrix is constructed again with the same secure technique used in the first MSD, but this time instead of a counting sort matrix for digits 0-9, a counting sort matrix for 0-99 is derived. The example in **Figure 4.4** continues for the second MSD as shown in **Figure 4.5**. To lower the communication complexity, a counting sort matrix was not created for any number that could not exist based on the counting sort matrix of the first MSD. For example, in our example in **Figure 4.4**, the counting sort matrix did not count any 4, 5, or 6 at the first MSD. Therefore, no counting matrix was made for numbers 40-69. The new counting sort matrix is distributed to each site again. However, unlike the first MSD, the second MSD and subsequent digits update the rank by adding to the rank derived from the previous digit. The algorithm continues on to the next most significant digits until the least

significant digit, typically resulting in few to no 'ties' in ranks. In our case, the last digit is the third digit, as shown in **Figure 4.6**.

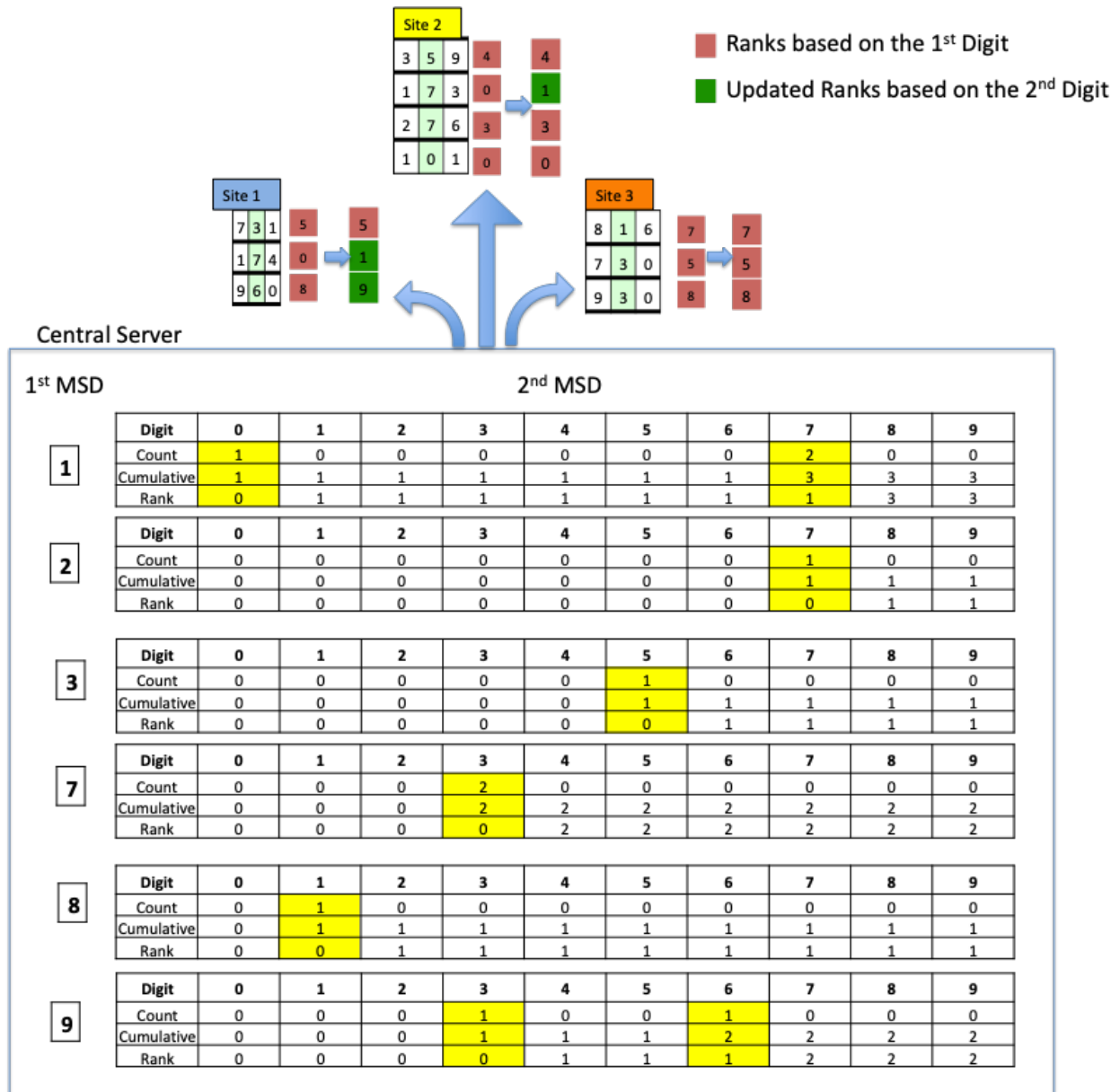


Figure 4.5 The Second Iteration of Federated Radix Sorting

Temporary ranks after two iterations of secure multiparty computation radix Sorting. Counting sort matrix is calculated for the 2nd MSD, using the same method.



1 st and 2 nd MSD		3 rd MSD										
		Digit	0	1	2	3	4	5	6	7	8	9
1 7	Count	0	0	0	1	1	0	0	0	0	0	0
	Cumulative	0	0	0	1	2	2	2	2	2	2	2
	Rank	0	0	0	0	1	2	2	2	2	2	2
7 3	Count	1	1	0	2	0	0	0	0	0	0	0
	Cumulative	1	2	2	2	2	2	2	2	2	2	2
	Rank	0	1	2	2	2	2	2	2	2	2	2

Figure 4.6 The Third Iteration of Federated Radix Sorting

Final rank after three iterations of secure multiparty computation radix Sorting. There are no 'ties.' Sites know what ranks belong to their estimates, but don't know the ranks of the other sites.

4.5 Secure Multiparty Computation Isotonic Regression

With estimates sorted, the next step to fit a monotonically increasing regression step function using the active set method is to find an optimal isotonic function given the current constraints, as previously shown in **Figure 4.2** Phase 2. The optimal solution given the constraints for isotonic regression is equal to the proportion of positive cases, as previously shown in

Equation (4.5). At the start of the algorithm, there are no breakpoints, therefore the solution, or \mathcal{Z} , is a vector of a constant.

At Phase 1, where the potential fitted regression \mathcal{Z} is calculated, the total number of positive cases are counted. The total again can be counted in a private manner using Algorithm 1 from *Wu et. al.*

At Phase 3 of **Figure 4.2**, to find the smallest Lagrange multiplier in a privacy-preserving manner, partial results λ_{ps} from each local site can be calculated and added to find λ in Equation (4.7)

$$\lambda_{ps} = M_{*,r} \times C_r$$

$$(4.7) \quad \lambda = \sum^n \lambda_{ps}$$

where r is the corresponding rank or column, $M_{*,r}$ represents the columns corresponding to the ranks that each local site has, and C_r represents the C_i values at each local site with the corresponding rank. C_r can be derived in each local site as shown in **Figure 4.7**. $M_{*,r}$ is sent to each local site and λ_{ps} results are added from site to site to obtain the final λ vector, which is shown in **Figure 4.8**. With a breakpoint found, the next recursive step can begin.

$$z =$$

rank	1	2	3	4	5	6	7	8	9	10
x	4/10	4/10	4/10	4/10	4/10	4/10	4/10	4/10	4/10	4/10

$$y =$$

Site 1		Site 2		Site 3		Site 4	
Rank	y_i	Rank	y_i	Rank	y_i	Rank	y_i
1	0	2	0	6	0	7	0
4	1	3	0	9	1	8	1
10	1	5	0				

$$c =$$

Rank	c_r	Rank	c_r	Rank	c_r	Rank	c_r
1	4/10	2	4/10	6	4/10	7	4/10
4	-6/10	3	4/10	9	-6/10	8	-6/10
10	-6/10	5	4/10				

Figure 4.7 Initial Step of Federated Isotonic Regression

z is derived by using Algorithm 1 from *Wu et. al.* to obtain the number of positive cases and the total number of instances. By subtracting y from z , c is derived at each site and can be used to find the least Lagrange multiplier.

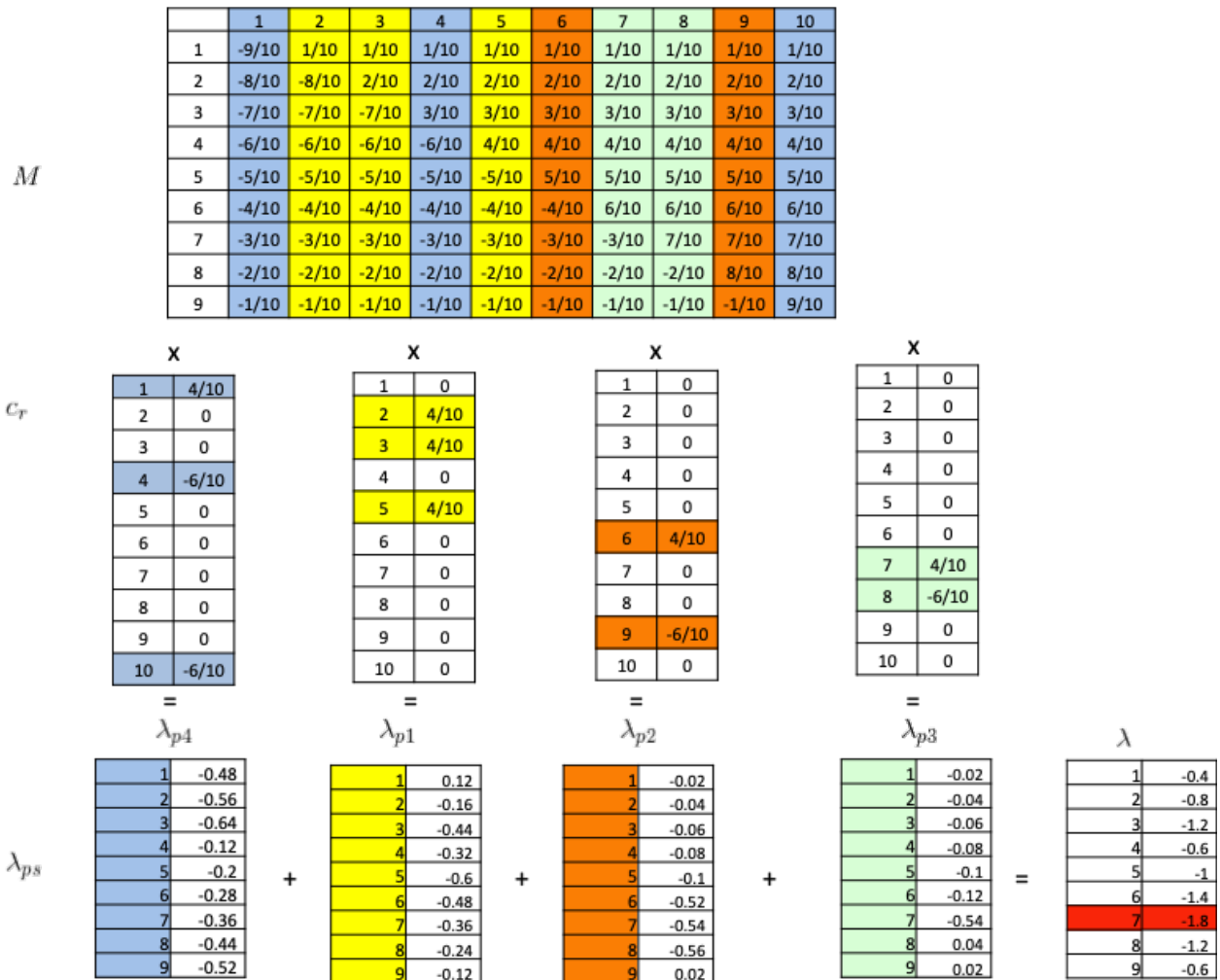


Figure 4.8 Finding the least Lagrange Multipliers

Multiplying c from each local site by the corresponding columns in M , each site can derive a vector of partial Lagrange multipliers, the summation of which results in the overall Lagrange multiplier. The sum is done by SMC so no site exposes its λ . The least Lagrange multiplier is the breakpoint where two steps of the isotonic regression are formed.

The rank of the smallest λ is where the breakpoint for the step is. After distributing the breakpoint to all sites and the central server, a new iteration can start. Each step created by the breakpoint can be seen as its own isotonic regression problem and solved with the above method, recursively.

4.6 Smooth Isotonic Regression

After the method above, each site should now have the “rank” and the new estimates \mathcal{Z} . To obtain smooth isotonic regression in a federated network, I modified the method from *Jiang et al.*¹⁰⁶ outlined in **Algorithm 4.1**. For SIR, the data point from each ‘step’ is used in a PCHIP to form a smooth monotonically increasing function. Using our toy example from **Figure 4.2**, where the breakpoints are the 3rd and 7th ranked instances, the y values for the ‘steps’ are the \mathcal{Z} . In this case, they are 0, $\frac{1}{4}$, and 1. The x values for the ‘steps’ are the average of the original x values at each step. To obtain the average at each ‘step’, Algorithm 1 from *Wu et al.*¹⁰⁸ is used. An outline of the steps is given in **Figure 4.9**.

Algorithm 4.1

1. The central server sends out a random vector of length S , where S is the number of steps in the isotonic regression. Each step spans breakpoints.
2. Each site, in sequence, adds the estimates to the random vector in the corresponding steps according to the rank of the steps and sends the vector to the next site. The final site sends the vector to the server.
3. The central server subtracts the initial random values from the vector.
4. The central server constructs a Piecewise Cubic Hermite Interpolating Polynomial function¹¹⁰ using the \mathcal{Z} at each step and the vector to obtain the final smooth isotonic regression function.

Rank	e_i	z_i
1	0.18	0
4	0.43	1/4
10	0.99	1

Rank	e_i	z_i
2	0.23	0
3	0.38	0
5	0.48	1/4

Rank	e_i	z_i
6	0.74	1/4
9	0.91	1

Rank	e_i	z_i
7	0.79	1/4
8	0.88	1

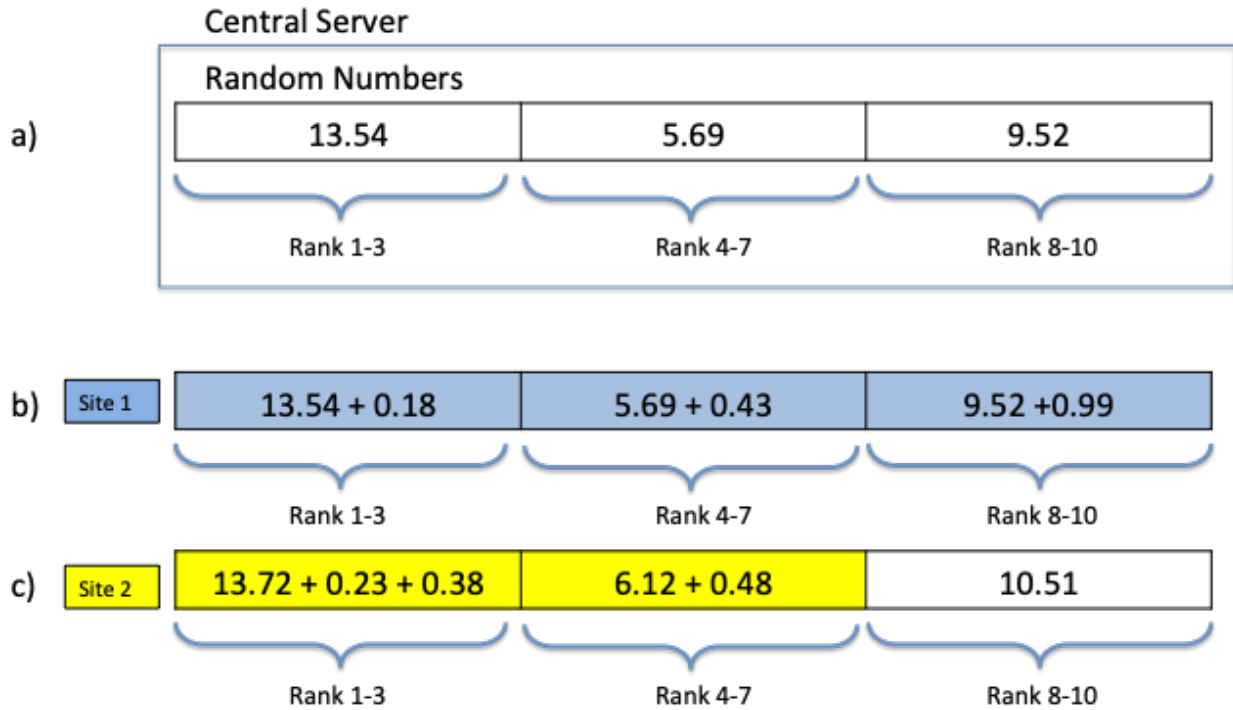
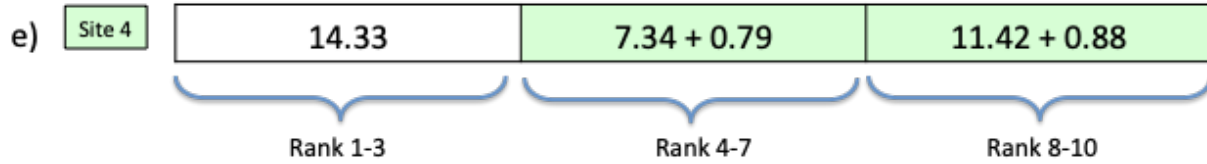
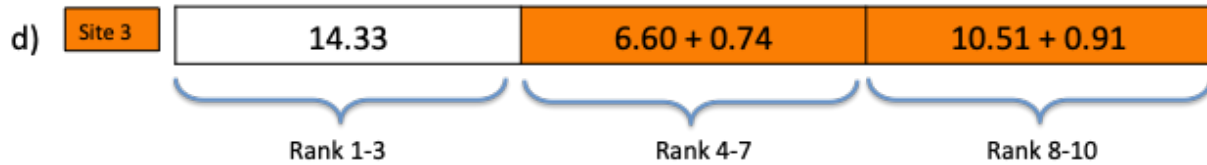


Figure 4.9 Smooth Isotonic Regression

For Smooth Isotonic Regression, a data point from each 'step' is used in a PCHIP to form a smooth monotonically increasing function. The x values of the data points are collected using Algorithm 1 from Wu et al. The process evolves as follows: (a) First, a random vector of numbers is created in the central server, where the length of the vector is the number of 'steps.' The central server also sends the ranks of each of the 'step.' The random vector is passed to Site 1, (b) which adds estimates to the random number in the corresponding rank. Then the vector is passed on to Site2, (c) which does the same and passes to Site 3, (d) which does the same and passes to Site 4, (e) which does the same. (f) The random vector is returned to the central server and the original random vector is subtracted. Finally, (g) the central server applies PCHIP to the x values and the y values to obtain a Smooth Isotonic Regression, shown here.



f) Central Server

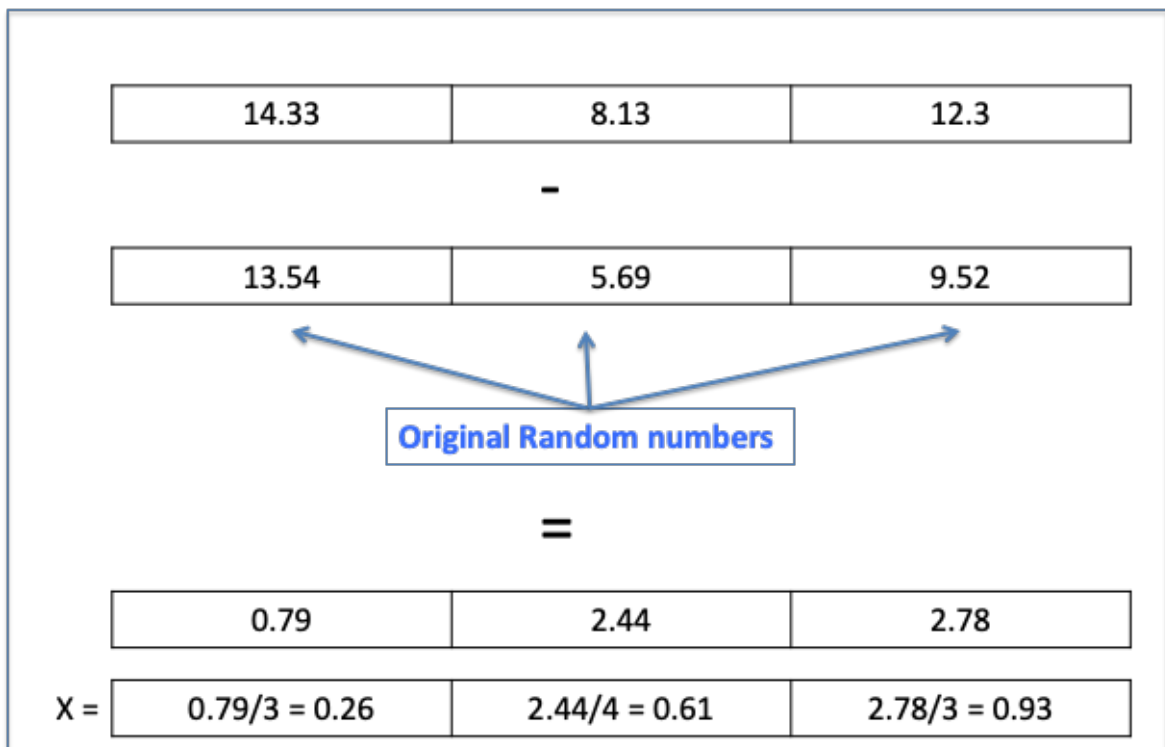


Figure 4.9 Smooth Isotonic Regression, continued

g)

Central
Server

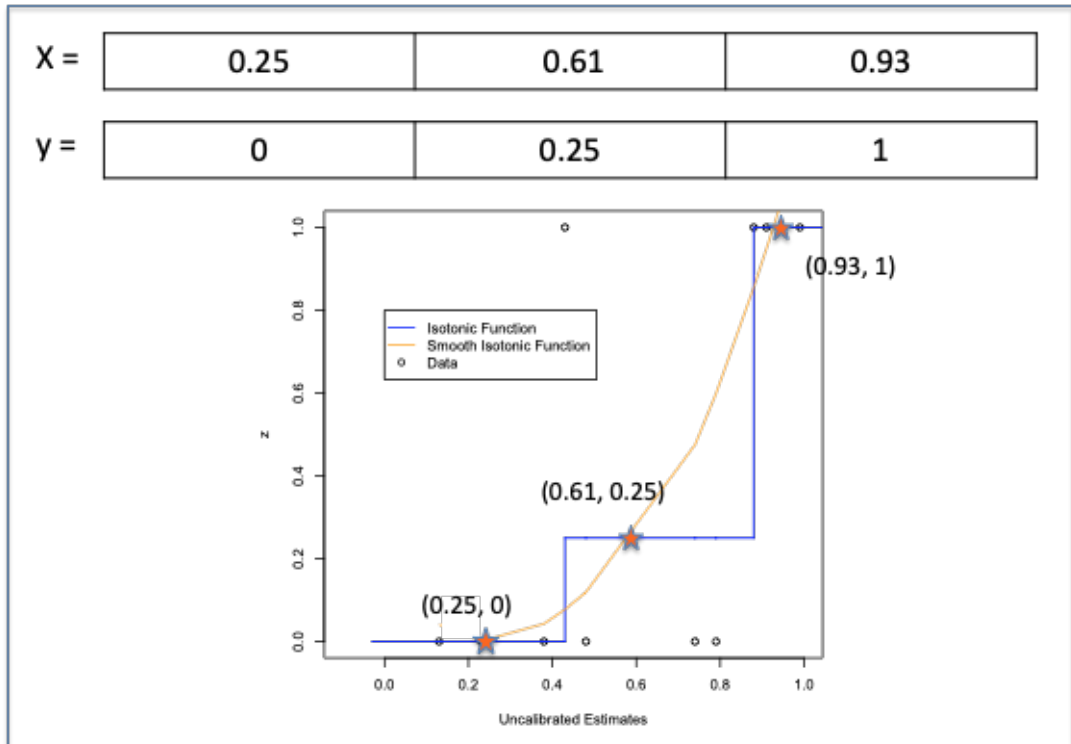


Figure 4.9 Smooth Isotonic Regression, continued

Since the sorting and counting performed across the sites using secure multiparty computation algorithms produces the same results as those that would be produced were all data be centralized, it is clear that there is no cost in terms of accuracy. However, since the algorithm requires aggregate data to be sent around, it is important to understand the communication costs.

4.7 Communication complexity

The communication complexity of the federated radix sort is at worst case $\mathcal{O}((\frac{n}{10^d} + \frac{n}{10^{d-1}} + \frac{n}{10^{d-2}} + \dots + \frac{n}{10^{d-d}}) \times s)$, where n is the number of instances, d is the number of the maximum number of digits, and s is the number of local sites. For each iteration or digit, the random number matrix used to aggregate the count of digits to create the counting sort grows by a factor of 10. The communication complexity of the federated SIR is $\mathcal{O}(n \log n \times s)$. For one iteration, a vector of length n is passed from site to site $2s + 1$ times. In the worst case, there are $\log n$ iterations, which happens in the extreme case of 0's and 1's alternating.

So far it is clear that it is possible to build calibration models in a distributed manner, but knowing whether or not a model is well calibrated is pre-requisite.

4.8 Measuring Calibration

Measuring calibration in a distributed manner is important because institutions may want to keep estimates and outcomes private. In order to evaluate a predictive model and a calibration model, I should assess calibration across all institutions. I can measure each local site's calibration separately, but to describe the model's overall performance, the global calibration must be assessed. For this Chapter, I extend calibration measurements in a federated manner: the H-L (C and H) test, the ECE, and the MCE that were shown in Chapter 3.

Algorithms 4.2 and **4.3** describe the distributed computation for the H-L test C version and the H version, respectively. To decide the number of bins, I used the method from *Paul et al.*⁸⁵ as shown in equation (4.8),

$$(4.8) \quad g = \max(10, \min\{\frac{m}{2}, \frac{n-m}{2}, 2 + 8(\frac{n}{1000})^2\})$$

where m is the number of positive cases and n is the total number of observations.

Algorithm 4.2: H-L C test

1. Each site obtains ranks by applying Secure Multiparty Radix Sorting to estimates. In the process, the central server records the total number of instances.
2. The central server calculates the number of instances for each bin by dividing the total number of instances by the number of bins, g .
3. The central server sends the following items to the first site:
 - a. Four random vectors of length g : $[E_1^+, E_2^+, \dots, E_g^+]$, $[E_1^-, E_2^-, \dots, E_g^-]$, $[O_1^+, O_2^+, \dots, O_g^+]$, and $[O_1^-, O_2^-, \dots, O_g^-]$. These four vectors correspond to the sum of the estimates of instances that are positive events within each bin (E^+), the sum of estimates of instances that are negative events within each bin (E^-), the number of positive events within each bin (O^+), and the number of negative events within each bin (O^-), respectively.
 - b. Range of ranks for each bin (i.e. if there are 10 bins and 200 instances, then the ranks for bin 1 would range from 1 to 20, ranks for bin 2 would range from 21 to 40, and so on.)
4. The first site adds the estimates of positive events to $[E_1^+, E_2^+, \dots, E_g^+]$ into the bin with the corresponding range of ranks, adds estimates of negative events to $[E_1^-, E_2^-, \dots, E_g^-]$, adds the number of positive cases into $[O_1^+, O_2^+, \dots, O_g^+]$, and adds the number of negative cases into $[O_1^-, O_2^-, \dots, O_g^-]$. The first site sends the vectors to the next site. The next site adds the estimates, number of positive cases, and number of negative cases to the corresponding vectors, then sends the vector to the next site, and so on. The last site sends the vectors to the central server.
5. The central server subtracts the initial random numbers from the four vectors. These aggregate values are used to calculate the H-L C test statistics.

Algorithm 4.3 H-L H Test

1. The central server calculates the number of instances in each bin by dividing the total number of instances by the number of bins, g .

2. The central server sends relevant items to the first site:
 - a. Four random vectors of length g : $[E_1^+, E_2^+, \dots, E_g^+]$, $[E_1^-, E_2^-, \dots, E_g^-]$, $[O_1^+, O_2^+, \dots, O_g^+]$, and $[O_1^-, O_2^-, \dots, O_g^-]$. These four vectors correspond to the sum of the estimates of instances that are positive events within each bin (E^+), the sum of estimates of instances that are negative events within each bin (E^-), number of positive events within each bin (O^+), and number of negative events within each bin (O^-), respectively.
 - b. Range of the estimates for each bin (i.e. if there are 10 bins, then bin 1 would have a range of 0.0 - 0.1, bin 2 would have a range of 0.1 - 0.2, and so on.)
3. The first site adds the estimates of positive events to $[E_1^+, E_2^+, \dots, E_g^+]$ into the bin with the corresponding range of ranks, adds estimates of negative events to $[E_1^-, E_2^-, \dots, E_g^-]$, adds the number of positive cases into $[O_1^+, O_2^+, \dots, O_g^+]$, and adds the number of negative cases into $[O_1^-, O_2^-, \dots, O_g^-]$. The first site sends the vectors to the next site. The next site adds estimates, number of positive cases, and number of negative cases to the corresponding vectors, then send to the next site, and so on. The last site sends the vector to the central server.
4. The central server subtracts the random numbers from the four vectors. These aggregate values are used to calculate H-L H test statistics.

The other measures I extend in a federated manner are the Maximum Calibration Error (MCE) and Expected Calibration Error (ECE).⁷¹⁻⁷³ **Algorithm 4.4** shows the calculation of ECE and MCE in a federated manner.

Algorithm 4.4

1. Each site obtains ranks by applying Secure Multiparty Radix Sorting to estimates.
2. The central server calculates the number of instances in each bin by dividing the total number of instances by the number of bins, g , calculated from equation (4.8).
3. The central server sends the following two relevant items to the first site:
 - a. Two random vectors of length g , $[E_1^+, E_2^+, \dots, E_g^+]$ and $[O_1^+, O_2^+, \dots, O_g^+]$. These two vectors correspond to the sum of the estimates of instances that are positive events within each bin (E^+) and the number of positive events within each bin (O^+), respectively.
 - b. Range of ranks in each bin (i.e. if there are 10 bins and 200 instances, then

ranks of bin 1 would range from 1 to 20, ranks of bin 2 would range from 21 to 40, and so on).

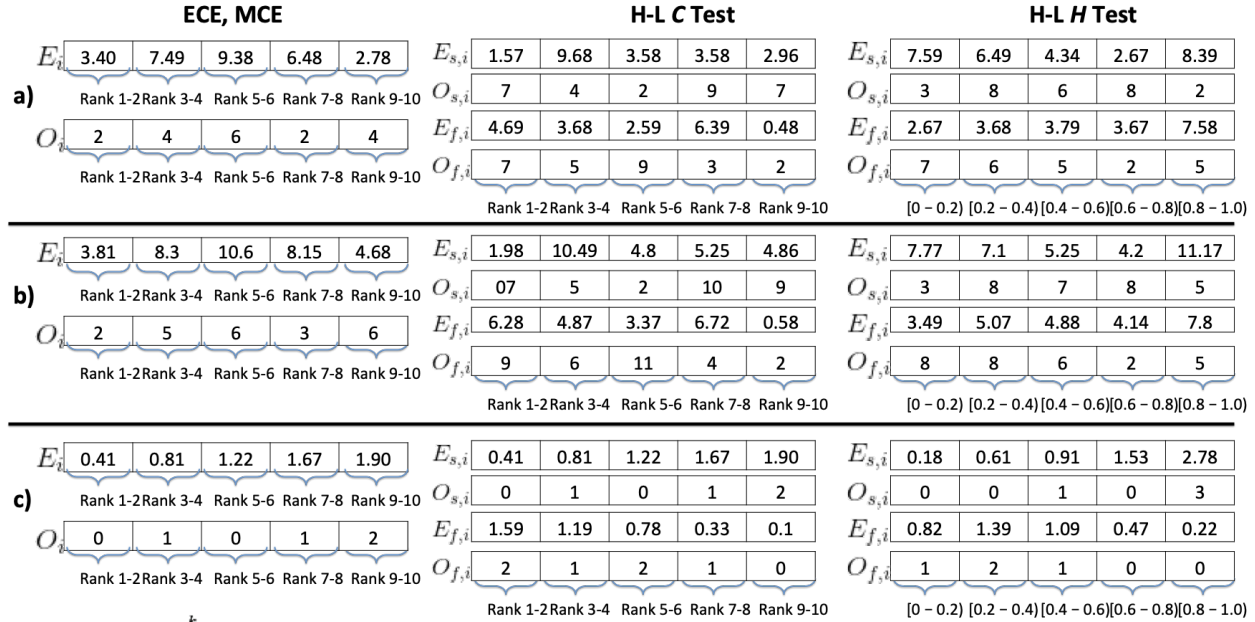
4. The first site adds the estimates of positive events to $[E_1^+, E_2^+, \dots, E_g^+]$ into the bin with the corresponding range of ranks and adds the number of positive cases into $[O_1^+, O_2^+, \dots, O_g^+]$. The first site sends the vectors to the next site. The next site adds the estimates and number of positive cases to the corresponding vector, then sends the vector onto the next site, and so on. The last site sends the vectors to the central server.
5. The central server subtracts the random numbers from the two vectors. These aggregate values are used to calculate the ECE and the MCE.

Rank	e_i	y_i
1	0.18	0
4	0.43	1
10	0.99	1

Rank	e_i	y_i
2	0.23	0
3	0.38	0
5	0.48	0

Rank	e_i	y_i
6	0.74	0
9	0.91	1

Rank	e_i	y_i
7	0.79	0
8	0.88	1



$$ECE = \sum_{i=1}^k P(i) \cdot |O_i - E_i|$$

$$MCE = \max_{i=1, \dots, K} (|O_i - E_i|)$$

$$test\ statistic = \sum_{i=1}^g \left[\frac{(O_{s,i} - E_{s,i})^2}{E_{s,i}} + \frac{(O_{f,i} - E_{f,i})^2}{E_{f,i}} \right]$$

Figure 4.10 Distributed Calibration Measurements

For calculation of distributed calibration measurements of ECE, MCE, H-L C Test, and the H-L H tests, (a) the central server sends random vectors to the first sites. The first site adds estimates, the number of positive cases, and the number of negative cases to the corresponding vectors. The first site then sends the vectors to the next site, and so on. After the last site has added its values, (b) is the resulting vectors, then the vectors are sent to the central server. The central server subtracts the random vectors from (a), resulting in (c), which the central server can use to calculate the ECE, MCE, H-L C Test, and H-L H Test.

4.9 Federated Calibration Model Experiments

Simulated and clinical data were used to validate the federated smooth isotonic regression model. To construct artificial data, 23 artificial features were constructed with 20 binary and 3 continuous independent variables, as in Chapter 3, Section 3.4. A uniform distribution was then used to determine the dependent variable. A total of 5,000 instances were

generated. To assess the usefulness and robustness of the federated calibration model, all data were separated into local sites. Artificial data were randomly divided into 6 sites.

For clinical data, I used the National Inpatient Sample (NIS), where six medical systems contributed a total of 17,184 instances. Thirty-eight features were selected, including age, sex, race, admission month, elective or non-elective admission, expected primary payer, median household income quartile range, and presence or absence of 30 chronic conditions. I predicted all-cause mortality, which occurred for 1.64% of the observations.

The second clinical data set was from the American College of Surgeons National Surgical Quality Improvement Program (ACS NSQIP), specifically focused on colectomy surgeries. I used 35,362 instances and 140 features, including preoperative features and features related to the surgery. They included demographics, preoperative lab values, pre-existing conditions, concurrent procedures, etc. The dependent variable was readmission within 30 days, which occurred for 5.18% of the observations. The instances from ACS NSQIP were divided according to the top 6 operative approaches: *Open (planned)*, *Laparoscopic*, *Laparoscopic with assistance*, *Laparoscopic with unplanned conversion to open*, *Robotic*, and *Robotic with assistance assist*.

Classifiers for each dataset were built using logistic regression. I applied a federated classifier, GLORE,¹⁸ to obtain estimates for each local site. Fifty percent of each site's data was used to train a classifier (Training Set Part 1), 25% was used to train a calibration model (Training Set Part 2), and 25% was used to measure calibration (Test Set). Ten-fold cross-validation was performed. Four different calibration models were trained for comparison:

- 1) No Calibration Model.

2) Benchmark: Build a calibration model assuming all the data from all the local sites are in a central server (global calibration model). This calibration model is applied to all test sets.

3) Local Calibration Model: Each local site builds its own calibration model using only its own data and applies it to its own estimates.

4) New Calibration Model: Build a calibration model in a federated manner (federated calibration model) then apply to all test sets.

All of the calibrations were measured in a federated manner using the federated H-L test (*C* and *H* versions), ECE, and MCE shown in the previous Sections.

The results for the simulated dataset, NIS mortality dataset, and ACS NSQIP colectomy dataset are shown in **Tables 4.1, 4.2, and 4.3**, respectively. These tables show one representative example for the H-L test, ECE, MCE. As expected, AUROCs did not change with the application of calibration models that used all data, since the monotonic transformations did not change the order of the estimates, as explained in Chapter 3. The AUROCs changed slightly with the use of local calibration models because the breakpoints for the local SIR were different from the global, federated one. Comparing Experiments 1, 2, and 4, global and federated calibration models lowered the H-L *C* statistics, H-L *H* statistics, MCE and ECE in the clinical datasets, while worsening (i.e., increasing) the H-L *C* statistics and H-L *H* statistics of the simulated data slightly, but not enough to reject the null hypothesis of the H-L test. The MCE and ECE did not change significantly in Experiments 1, 2, and 4 in the simulated data. Experiment 3 shows that, if each local site builds its own calibration model with local data and applies to its estimates, the results are not properly calibrated, increasing H-L *C* statistics, H-L *H* statistics, MCE and ECE. The federated calibration model helped improve calibration in this case.

Table 4.1 Results for the Simulated Dataset.

	No Calibration Model	Benchmark	Local Calibration Model	Federated Calibration Model
AUROC	0.871	0.871	0.853	0.871
H-L C Statistics (p-value)	20.456 (0.084)	6.438 (0.892)	1046.948 (<1e-22)	6.438 (0.892)
H-L H Statistics (p-value)	16.577 (0.166)	3.830 (0.975)	34.307 (0.001)	3.830 (0.975)
ECE	0.143	0.090	0.132	0.090
MCE	0.268	0.264	0.269	0.264

Table 4.2 Results for the NIS Mortality Dataset.

	No Calibration Model	Benchmark	Local Calibration Model	Federated Calibration Model
AUROC	0.862	0.862	0.837	0.862
H-L C Statistics (p-value)	27.875 (0.143)	18.566 (0.885)	34.383 (0.023)	18.566 (0.885)
H-L H Statistics (p-value)	36.108 (0.014)	18.347 (0.191)	92.697 (5.505e-11)	18.347 (0.191)
ECE	0.304	0.195	0.298	0.195
MCE	0.838	0.576	0.973	0.576

Table 4.3 Results for the ACS NSQIP colectomy Readmission Data set.

	No Calibration Model	Centralized Calibration Model	Local Calibration Model	Federated Calibration Model
AUROC	0.674	0.674	0.682	0.674
H-L C Statistics (p-value)	472.588 (0.308)	445.848 (0.649)	524.833 (0.016)	445.848 (0.649)
H-L H Statistics (p-value)	9334.108 (<1e-22)	182.842 (0.220)	1746.332 (<1e-22)	182.842 (0.220)
ECE	0.197	0.138	0.192	0.138
MCE	0.766	0.668	0.767	0.668

4.10 Discussion

A smooth isotonic regression model can be built in a federated manner. The method I developed used peer-to-peer aggregation of intermediary results to first sort the estimates and then derive the smooth isotonic regression function. The results show that a calibration model built with local data may not be enough to improve calibration. Leveraging data in a federated manner such that no private data are revealed is very important to medical systems that desire to collaborate but are reluctant to share individual-level data. Sharing intermediary results is less risky but not completely guaranteed to protect privacy, particularly in rare cases, therefore additional studies of privacy risks involving the intermediary aggregations are still necessary.

The current algorithms can safeguard against semi-honest adversaries (i.e., they are honest but ‘curious’), but not against malicious adversaries. The participants of the distributed calibration model building process need to follow the prescribed protocol of the algorithm in

order for the model to be built correctly. A malicious adversary, on the other hand, can disrupt the entire process. In terms of data privacy, no observation-level data are leaked to semi-honest participants. However, if more than one participating member is malicious, it is possible to collude and potentially gain some information about other participating members. No safeguard is in place to combat malicious adversaries. Even if all participants are just semi-honest adversaries, the federated sorting algorithm requires more than two participating members in order for data to remain private. Because the Counting Sort Matrix is distributed to all sites, if there are only two sites and if a site subtracts the counts of its own digits from the Counting Sort Matrix, the other site's counts would be revealed. In order to protect the count of the digits, the central server can inject random numbers to create an artificial third site or each local site can create artificial instances. However, to properly account for these random numbers in the final SIR, more research is required.

I showed here that a calibration model built in a federated manner produced the same results as a calibration model built from centralized data. Such a method is important to curtail the difficulties in model building due to the lack of data interoperability in healthcare. It provides researchers who may have too few instances with a reliable method to construct calibration models, avoiding bias and noise with the help of a large sample of data in a privacy-preserving manner.

Chapter 4, in part, is in preparation for submission as *Isotonic Regression Calibration through a Federated Network of Private Localized Patient Data*. Huang, Yingxiang; Gabriel, Rodney; Jiang, Xiaoqian; Ohno-Machado, Lucila, 2020. The dissertation author was the primary investigator and author of this paper.

5. Final Remarks

Data sharing has been a big challenge in biomedical informatics due to privacy or other concerns. Patient data privacy is an important issue since the leakage of sensitive material could lead to social and economical damage. In order to protect data privacy, patient data are generally protected in localized silos, which makes consolidation of data from different medical systems difficult. However, in order to build predictive models, a large sample size is often needed.

I described innovative methods to (1) build federated predictive models using contextual embeddings, (2) measure calibration in a federated manner, and (3) build calibration models in a federated manner. Contextual embedding models have demonstrated a very strong representative capability to describe medical concepts (and their context), and they have shown promise as an alternative way to support deep learning applications without the need to disclose original data. However, contextual embedding models acquired from individual medical systems cannot be directly combined because their embedding spaces are different and naive pooling renders combined embeddings useless. I presented a novel approach to address these issues to promote sharing representations without sharing individual-level data. I built a global model from representations learned from local private data without sacrificing privacy and synchronize information from multiple sources.

The proposed method harmonizes different local contextual embeddings into a global model. I used a popular NLP tool to generate contextual embeddings from each source and the Procrustes method to fuse different vector models into one common space by using a list of corresponding pairs as anchor points. With harmonized embeddings, I made predictions based on cosine similarity to a diagnosis vector.

I used sequential medical events extracted from the MIMIC-III database to evaluate the proposed methods in predicting the next likely diagnosis of a new patient using structured data and unstructured data. Under different experimental scenarios, I confirmed that the global model built from harmonized local representation models achieved more accurate predictions than local models. Such an aggregation of local models using our unique representation harmonization can serve as a proxy for a global model, combining information from a wide range of institutions and information sources to generate diagnosis vectors. It allows information unique to a certain medical system to become available to other sites, in an aggregate fashion.

In addition to building models in a federated manner, it is also important to evaluate them this way. While measuring discrimination in a federated manner has previously been described,^{18,108} measuring calibration this way has not been previously described. Measuring calibration is paramount, especially in the field of biomedical informatics. Existing calibration models have been shown to improve calibration. Such models are often post-processing models where estimates produced by a predictive model and the true outcomes are used to build calibration models. However, the constraint of building models locally limits the degree of calibration improvement. I used a Secure Multiparty Computation (SMC) method to build a global isotonic regression calibration model using data from different medical systems without sharing individual-level data. To show that the federated model could outperform local models, I also proposed measuring calibration in such a federated manner.

With the help of a central server that does not contribute any private data nor sees data from the sites, I first sorted the estimates produced by a predictive model without sharing the actual estimates. The sorted estimates were useful in both building calibration models and

measuring calibration in a federated manner. To train an isotonic regression model in a federated manner, I found that the breakpoints at which each step of the isotonic regression function can be determined by a combination of adding random numbers and sharing partial results from each local site with the central server. I conducted experiments on both simulated and clinical data and compared the calibration from a centralized isotonic regression and from federated Isotonic regression. The results are essentially the same, but privacy protection comes at the expense of computational efficiency.

Valuable models have been built from large multi-institutional studies, but such collaborations take considerable effort to establish due to the concerns over patient and institutional privacy. Federated learning allows for aggregation of knowledge, while alleviating some concerns on privacy by keeping medical system data in their respective sites as much as possible. With the possibility of building both classifier and calibration models in a federated manner, I can expect model building and evaluation to occur without sharing of sensitive information. Continued development in federated learning models and their evaluations will encourage more collaborations between institutions, fostering an environment for new discoveries from big data that otherwise would not be possible using only localized data.

Finally, the novel solutions I developed and describe here do not limit themselves to applications in healthcare, although they were inspired by real scenarios of clinical data research networks that cannot share individual-level data but still want to collaborate in research. For example, networks such as pSCANNER¹¹⁴ and many networks based on i2b2 software¹¹⁵ operate under these circumstances. It is possible to envision equivalent situations in other fields. The solutions I developed could be used for applications in finance, marketing, and many other areas

in which there is a need to use data from multiple institutions or countries to personalize predictions, sharing estimates and outcomes is not possible, but it is important to use as much data as possible to verify that the predictive models are well calibrated (and consequently try to improve calibration if necessary).

6. Bibliography

1. 111th Congress, *American Recovery and Reinvestment Act of 2009* (2009).
2. E. Jamoom, *Table of Electronic Health Record Adoption and Use among Office-based Physicians in the U.S., by State* (2015 National Electronic Health Records Survey, 2016).
3. 114th Congress, *Medicare Access and CHIP Reauthorization Act of 2015* (2015).
4. I. S. Kohane, J. M. Drazen, E. W. Campion, A glimpse of the next 100 years in medicine., *N. Engl. J. Med.* **367**, 2538–2539 (2012).
5. I. S. Patten, S. Rana, S. Shahul, G. C. Rowe, C. Jang, L. Liu, M. R. Hacker, J. S. Rhee, J. Mitchell, F. Mahmood, P. Hess, C. Farrell, N. Koulisis, E. V. Khankin, S. D. Burke, I. Tudorache, J. Bauersachs, F. del Monte, D. Hilfiker-Kleiner, S. A. Karumanchi, Z. Arany, Cardiac angiogenic imbalance leads to peripartum cardiomyopathy., *Nature* **485**, 333–338 (2012).
6. S. I. Berndt, S. Gustafsson, R. Mägi, A. Ganna, E. Wheeler, M. F. Feitosa, A. E. Justice, K. L. Monda, D. C. Croteau-Chonka, F. R. Day, T. Esko, T. Fall, T. Ferreira, D. Gentilini, A. U. Jackson, J. Luan, J. C. Randall, S. Vedantam, C. J. Willer, T. W. Winkler, et al., Genome-wide meta-analysis identifies 11 new loci for anthropometric traits and provides insights into genetic architecture., *Nat. Genet.* **45**, 501–512 (2013).
7. Council of the European Union, *Proposal for a Regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)* (2015).
8. D. Tran, R. THOMAS, Department of Veterans Affairs Handbook, (2017).
9. Privacy Policy, All of Us (available at <https://www.joinallofus.org/en/privacy-policy>).
10. Strong Heart Study (available at <https://strongheartstudy.org/Research/Study-Data-and-Study-Samples/Study-Data#391231915-data-and-summary-statistics-request-policy>).
11. Office for Civil Rights, HHS, Standards for privacy of individually identifiable health information. Final rule., *Fed. Regist.* **67**, 53181–53273 (2002).
12. M. M. Douglass, G. D. Clifford, A. Reisner, W. J. Long, G. B. Moody, R. G. Mark, in *Computers in Cardiology, 2005*, (IEEE, 2005), pp. 331–334.
13. A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, MIMIC-III, a freely accessible critical care database., *Sci. Data* **3**, 160035 (2016).

14. I. Neamatullah, M. M. Douglass, L. H. Lehman, A. Reisner, M. Villarroel, W. J. Long, P. Szolovits, G. B. Moody, R. G. Mark, G. D. Clifford, Automated de-identification of free-text medical records., *BMC Med. Inform. Decis. Mak.* **8**, 32 (2008).
15. F. Deroncourt, J. Y. Lee, O. Uzuner, P. Szolovits, De-identification of patient notes with recurrent neural networks., *J. Am. Med. Inform. Assoc.* **24**, 596–606 (2017).
16. B. R. South, D. Mowery, Y. Suo, J. Leng, Ó. Ferrández, S. M. Meystre, W. W. Chapman, Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual de-identification of clinical text., *J. Biomed. Inform.* **50**, 162–172 (2014).
17. K. El Emam, E. Jonker, L. Arbuckle, B. Malin, A systematic review of re-identification attacks on health data., *PLoS ONE* **6**, e28071 (2011).
18. Y. Wu, X. Jiang, J. Kim, L. Ohno-Machado, Grid Binary LOGistic REGression (GLORE): building shared models without sharing data., *J. Am. Med. Inform. Assoc.* **19**, 758–764 (2012).
19. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and their Compositionality, *Adv Neural Inf Process Syst* **26**, 3111–3119 (2013).
20. W. Jiang, P. Li, S. Wang, Y. Wu, M. Xue, L. Ohno-Machado, X. Jiang, WebGLORE: a web service for Grid LOGistic REGression., *Bioinformatics* **29**, 3238–3240 (2013).
21. S. Toh, R. Wellman, R. Y. Coley, C. Horgan, J. Sturtevant, E. Moyneur, C. Janning, R. Pardee, K. J. Coleman, D. Arterburn, K. McTigue, J. Anau, A. J. Cook, Combining distributed regression and propensity scores: a doubly privacy-protecting analytic method for multicenter research., *Clin. Epidemiol.* **10**, 1773–1786 (2018).
22. S. Toh, S. L. Rifas-Shiman, P.-I. Lin, L. C. Bailey, C. B. Forrest, C. E. Horgan, D. Lunsford, E. Moyneur, J. L. Sturtevant, J. G. Young, J. P. Block, PCORnet Antibiotics and Childhood Growth Study Group, Privacy-protecting multivariable-adjusted distributed regression analysis for multi-center pediatric study., *Pediatr. Res.* (2019), doi:10.1038/s41390-019-0596-0.
23. H. Yu, X. Jiang, J. Vaidya, in *Proceedings of the 2006 ACM symposium on Applied computing-SAC '06*, (ACM Press, New York, New York, USA, 2006), p. 603.
24. J. Que, X. Jiang, L. Ohno-Machado, A collaborative framework for Distributed Privacy-Preserving Support Vector Machine learning., *AMIA Annu. Symp. Proc.* **2012**, 1350–1359 (2012).
25. M. Shaneck, Y. Kim, V. Kumar, in *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, (IEEE, 2006), pp. 541–545.

26. A. Amir-Khalili, S. Kianzad, R. Abugharbieh, I. Beschastnikh, in *Machine learning in medical imaging*, Lecture notes in computer science. Q. Wang, Y. Shi, H.-I. Suk, K. Suzuki, Eds. (Springer International Publishing, Cham, 2017), vol. 10541, pp. 176–184.
27. C.-L. Lu, S. Wang, Z. Ji, Y. Wu, L. Xiong, X. Jiang, L. Ohno-Machado, WebDISCO: a web service for distributed cox model learning without patient-level data sharing., *J. Am. Med. Inform. Assoc.* **22**, 1212–1219 (2015).
28. Y. Vilks, Z. Zhang, J. Young, Q. L. Her, J. M. Malenfant, S. Malek, S. Toh, A distributed regression analysis application based on SAS software Part II: Cox proportional hazards regression, *arXiv* (2018).
29. Y. Kim, J. Sun, H. Yu, X. Jiang, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining- KDD '17*, (ACM Press, New York, New York, USA, 2017), pp. 887–895.
30. J. A. Rassen, J. Avorn, S. Schneeweiss, Multivariate-adjusted pharmacoepidemiologic analyses of confidential information pooled from multiple health care utilization databases., *Pharmacoepidemiol. Drug Saf.* **19**, 848–857 (2010).
31. S. Toh, S. Shetterly, J. D. Powers, D. Arterburn, Privacy-preserving analytic methods for multisite comparative effectiveness and patient-centered outcomes research., *Med. Care* **52**, 664–668 (2014).
32. S. Toh, M. E. Reichman, M. Houstoun, X. Ding, B. H. Fireman, E. Gravel, M. Levenson, L. Li, E. Moyneur, A. Shoaibi, G. Zornberg, S. Hennessy, Multivariable confounding adjustment in distributed data networks without sharing of patient-level data., *Pharmacoepidemiol. Drug Saf.* **22**, 1171–1177 (2013).
33. K. El Emam, S. Samet, L. Arbuckle, R. Tamblyn, C. Earle, M. Kantarcioglu, A secure distributed logistic regression protocol for the detection of rare adverse drug events., *J. Am. Med. Inform. Assoc.* **20**, 453–461 (2013).
34. S. Wang, X. Jiang, Y. Wu, L. Cui, S. Cheng, L. Ohno-Machado, EXpectation Propagation LOGistic REgression (EXPLORER): distributed privacy-preserving online model learning., *J. Biomed. Inform.* **46**, 480–496 (2013).
35. J. Reiter, C. Kohnen, A. Lin, A. Sanil, Secure Regression for Vertically Partitioned, Partially Overlapping Data, *Proceedings of the American Statistical Association* (2004).
36. A. Karr, X. Lin, A. Sanil, J. Reiter, Privacy-Preserving Analysis of Vertically Partitioned Data Using Secure Matrix Products, *J Off Stat* (2009).

37. Y. Li, X. Jiang, S. Wang, H. Xiong, L. Ohno-Machado, VERTICAL Grid Logistic regression (VERTIGO)., *J. Am. Med. Inform. Assoc.* **23**, 570–579 (2016).
38. W. Farhan, Z. Wang, Y. Huang, S. Wang, F. Wang, X. Jiang, A predictive model for medical events based on contextual embedding of temporal sequences., *JMIR Med. Inform.* **4**, e39 (2016).
39. E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, J. Sun, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, (ACM Press, New York, New York, USA, 2016), pp. 1495–1504.
40. Y. Wu, M. Jiang, J. Lei, H. Xu, Named entity recognition in chinese clinical text using deep neural network., *Stud. Health Technol. Inform.* **216**, 624–628 (2015).
41. Y. Liu, T. Ge, K. Mathews, H. Ji, D. McGuinness, in *Proceedings of BioNLP 15*, (Association for Computational Linguistics, Stroudsburg, PA, USA, 2015), pp. 92–97.
42. P. Nguyen, T. Tran, N. Wickramasinghe, S. Venkatesh, DeepR: A convolutional net for medical records., *IEEE J. Biomed. Health Inform.* **21**, 22–30 (2017).
43. Z. Che, Exploiting Convolutional Neural Network for Risk Prediction with Medical Feature Embedding, *NIPS ML4HC* (2017).
44. B. Shickel, P. J. Tighe, A. Bihorac, P. Rashidi, Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis., *IEEE J. Biomed. Health Inform.* **22**, 1589–1604 (2018).
45. T. Inc., M. View, International Conference on Learning Representations, *ICLR* (2013).
46. J. Pennington, R. Socher, C. Manning, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Association for Computational Linguistics, Stroudsburg, PA, USA, 2014), pp. 1532–1543.
47. J. Choo, S. Bohn, G. C. Nakamura, A. M. White, H. Park, in *Proceedings of the 2012 SIAM International Conference on Data Mining*, J. Ghosh, H. Liu, I. Davidson, C. Domeniconi, C. Kamath, Eds. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012), pp. 177–188.
48. A. R. Aronson, Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program., *Proc. AMIA Symp.*, 17–21 (2001).
49. A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, S. Jaiswal, A. Narayanan, graph2vec: Learning Distributed Representations of Graphs, *ACM* (2017).

50. E. W. Steyerberg, A. J. Vickers, N. R. Cook, T. Gerds, M. Gonen, N. Obuchowski, M. J. Pencina, M. W. Kattan, Assessing the performance of prediction models: a framework for traditional and novel measures., *Epidemiology* **21**, 128–138 (2010).
51. A. C. Alba, T. Agoritsas, M. Walsh, S. Hanna, A. Iorio, P. J. Devereaux, T. McGinn, G. Guyatt, Discrimination and calibration of clinical prediction models: users' guides to the medical literature., *JAMA* **318**, 1377–1384 (2017).
52. E. W. Steyerberg, *Clinical Prediction Models* (Springer New York, New York, NY, 2009).
53. M. D. Hurd, P. Martorell, A. Delavande, K. J. Mullen, K. M. Langa, Monetary costs of dementia in the United States., *N. Engl. J. Med.* **368**, 1326–1334 (2013).
54. S. Licher, P. Yilmaz, M. J. G. Leening, F. J. Wolters, M. W. Vernooij, B. C. M. Stephan, M. K. Ikram, M. A. Ikram, External validation of four dementia prediction models for use in the general community-dwelling population: a comparative analysis from the Rotterdam Study., *Eur. J. Epidemiol.* (2018), doi:10.1007/s10654-018-0403-y.
55. J. M. Firnhaber, Estimating Cardiovascular Risk., *Am. Fam. Physician* **95**, 580–581 (2017).
56. MELD Calculator - OPTN (available at <https://optn.transplant.hrsa.gov/resources/allocation-calculators/meld-calculator/>).
57. C. Fenlon, L. O'Grady, M. L. Doherty, J. Dunnion, A discussion of calibration techniques for evaluating binary and categorical predictive models., *Prev. Vet. Med.* **149**, 107–114 (2018).
58. C. G. Walsh, K. Sharman, G. Hripcsak, Beyond discrimination: A comparison of calibration methods and clinical usefulness of predictive models of readmission risk., *J. Biomed. Inform.* **76**, 9–18 (2017).
59. E. W. Steyerberg, Y. Vergouwe, Towards better clinical prediction models: seven steps for development and an ABCD for validation., *Eur. Heart J.* **35**, 1925–1931 (2014).
60. B. S. Wessler, L. Lai Yh, W. Kramer, M. Cangelosi, G. Raman, J. S. Lutz, D. M. Kent, Clinical prediction models for cardiovascular disease: tufts predictive analytics and comparative effectiveness clinical prediction model database., *Circ. Cardiovasc. Qual. Outcomes* **8**, 368–375 (2015).
61. F. E. Harrell, K. L. Lee, R. M. Califf, D. B. Pryor, R. A. Rosati, Regression modelling strategies for improved prognostic prediction., *Stat. Med.* **3**, 143–152 (1984).
62. F. E. Harrell, Evaluating the yield of medical tests, *JAMA* **247**, 2543 (1982).

63. A. A. Kramer, J. E. Zimmerman, Assessing the calibration of mortality benchmarks in critical care: The Hosmer-Lemeshow test revisited., *Crit. Care Med.* **35**, 2052–2056 (2007).
64. A. Niculescu-Mizil, R. Caruana, in *Proceedings of the 22nd international conference on Machine learning - ICML '05*, (ACM Press, New York, New York, USA, 2005), pp. 625–632.
65. K. Rufibach, Use of Brier score to assess binary predictions., *J. Clin. Epidemiol.* **63**, 938–9; author reply 939 (2010).
66. G. W. Brier, Verification of forecasts expressed in terms of probability, *Mon. Wea. Rev.* **78**, 1–3 (1950).
67. D. W. Hosmer, S. Lemeshow, Goodness of fit tests for the multiple logistic regression model, *Communications in Statistics - Theory and Methods* **9**, 1043–1069 (1980).
68. D. W. Hosmer, *Applied Logistic Regression* (Wiley-Interscience Publication, ed. 2nd, 2000).
69. S. R. Lele, A New Method for Estimation of Resource Selection Probability Function, *Journal of Wildlife Management* **73**, 122–127 (2009).
70. E. A. Freeman, G. Moisen, PresenceAbsence: an R package for presence absence analysis, *J. Stat. Softw.* **23** (2008), doi:10.18637/jss.v023.i11.
71. Y. Wang, L. Li, C. Dang, Calibrating Classification Probabilities with Shape-restricted Polynomial Regression., *IEEE Trans. Pattern Anal. Mach. Intell.* (2019), doi:10.1109/TPAMI.2019.2895794.
72. C. Guo, G. Pleiss, Y. Sun, K. Weinberger, On Calibration of Modern Neural Networks, *Proceedings of the 34 th International Conference on Machine Learning* (2017).
73. M. P. Naeni, G. F. Cooper, M. Hauskrecht, Obtaining well calibrated probabilities using bayesian binning., *Proc. Conf. AAAI Artif. Intell.* **2015**, 2901–2907 (2015).
74. P. C. Austin, E. W. Steyerberg, The Integrated Calibration Index (ICI) and related metrics for quantifying the calibration of logistic regression models., *Stat. Med.* **38**, 4051–4065 (2019).
75. X. Jiang, A. Menon, S. Wang, J. Kim, L. Ohno-Machado, Doubly Optimized Calibrated Support Vector Machine (DOC-SVM): an algorithm for joint optimization of discrimination and calibration., *PLoS ONE* **7**, e48823 (2012).
76. C. G. Walsh, J. D. Ribeiro, J. C. Franklin, Predicting risk of suicide attempts over time through machine learning, *Clinical Psychological Science* **5**, 457–469 (2017).

77. C. G. Walsh, J. D. Ribeiro, J. C. Franklin, Predicting suicide attempts in adolescents with longitudinal clinical data and machine learning., *J. Child Psychol. Psychiatry* **59**, 1261–1270 (2018).
78. B. Van Calster, D. Nieboer, Y. Vergouwe, B. De Cock, M. J. Pencina, E. W. Steyerberg, A calibration hierarchy for risk models was defined: from utopia to empirical data., *J. Clin. Epidemiol.* **74**, 167–176 (2016).
79. R. D. Riley, J. Ensor, K. I. E. Snell, T. P. A. Debray, D. G. Altman, K. G. M. Moons, G. S. Collins, External validation of clinical prediction models using big datasets from e-health records or IPD meta-analysis: opportunities and challenges., *BMJ* **353**, i3140 (2016).
80. J. Platt, Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, *In Advances in Large Margin Classifiers* (1999).
81. J. de Leeuw, K. Hornik, P. Mair, Isotone Optimization in R : Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods, *J. Stat. Softw.* **32** (2009), doi:10.18637/jss.v032.i05.
82. B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers, *ICML* , 609–616 (2001).
83. D. Heckerman, D. Geiger, D. M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, *Mach. Learn.* **20**, 197–243 (1995).
84. M. Kull, T. de M. e S. Filho, P. Flach, Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers, *AISTATS* **54** (2017).
85. P. Paul, M. L. Pennell, S. Lemeshow, Standardizing the power of the Hosmer-Lemeshow goodness of fit test in large data sets., *Stat. Med.* **32**, 67–80 (2013).
86. W. Yu, W. Xu, L. Zhu, A modified Hosmer–Lemeshow test for large data sets, *Communications in Statistics - Theory and Methods* **46**, 1–13 (2017).
87. X. Lai, L. Liu, A simple test procedure in standardizing the power of Hosmer–Lemeshow test in large data sets, *J. Stat. Comput. Simul.* **88**, 2463–2472 (2018).
88. B. Ambale-Venkatesh, X. Yang, C. O. Wu, K. Liu, W. G. Hundley, R. McClelland, A. S. Gomes, A. R. Folsom, S. Shea, E. Guallar, D. A. Bluemke, J. A. C. Lima, Cardiovascular Event Prediction by Machine Learning: The Multi-Ethnic Study of Atherosclerosis., *Circ. Res.* **121**, 1092–1101 (2017).
89. F. Sahm, D. Schrimpf, D. Stichel, D. T. W. Jones, T. Hielscher, S. Schefzyk, K. Okonechnikov, C. Koelsche, D. E. Reuss, D. Capper, D. Sturm, H.-G. Wirsching, A. S. Berghoff, P. Baumgarten, A. Kratz, K. Huang, A. K. Wefers, V. Hovestadt, M. Sill, H. P. Ellis, A. von Deimling, DNA methylation-based classification and grading system for meningioma: a multicentre, retrospective analysis., *Lancet Oncol.* **18**, 682–694 (2017).

90. P. K. Bendapudi, S. Hurwitz, A. Fry, M. B. Marques, S. W. Waldo, A. Li, L. Sun, V. Upadhyay, A. Hamdan, A. M. Brunner, J. M. Gansner, S. Viswanathan, R. M. Kaufman, L. Uhl, C. P. Stowell, W. H. Dzik, R. S. Makar, Derivation and external validation of the PLASMIC score for rapid assessment of adults with thrombotic microangiopathies: a cohort study., *Lancet Haematol.* **4**, e157–e164 (2017).
91. B. N. Manktelow, E. S. Draper, D. J. Field, Predicting neonatal mortality among very preterm infants: a comparison of three versions of the CRIB score., *Arch. Dis. Child. Fetal Neonatal Ed.* **95**, F9–F13 (2010).
92. D. J. Spiegelhalter, Probabilistic prediction in patient management and clinical trials., *Stat. Med.* **5**, 421–433 (1986).
93. N. Khavanin, C. S. Qiu, A. S. Mlodinow, M. M. Vu, R. G. Dorfman, N. A. Fine, J. Y. S. Kim, External validation of the breast reconstruction risk assessment calculator., *J. Plast. Reconstr. Aesthet. Surg.* **70**, 876–883 (2017).
94. M. S. Nascimento, C. M. Rebello, L. A. P. A. Vale, É. Santos, C. do Prado, Spontaneous breathing test in the prediction of extubation failure in the pediatric population., *Einstein (Sao Paulo)* **15**, 162–166 (2017).
95. J. Bröcker, L. A. Smith, Increasing the reliability of reliability diagrams, *Wea. Forecasting* **22**, 651–661 (2007).
96. S. Yao, Y. Zhao, A. Zhang, S. Hu, H. Shao, C. Zhang, L. Su, T. Abdelzaher, Deep learning for the internet of things, *Computer* **51**, 32–41 (2018).
97. K. Lee, K. Lee, H. Lee, J. Shin, A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, *NeurIPS 2018* (2018).
98. W. Maddox, T. Garipov, P. Izmailov, D. Vetrov, A. G. Wilson, A Simple Baseline for Bayesian Uncertainty in Deep Learning, *arXiv* (2019).
99. E. W. Steyerberg, D. Nieboer, T. P. A. Debray, H. C. van Houwelingen, Assessment of heterogeneity in an individual participant data meta-analysis of prediction models: An overview and illustration., *Stat. Med.* **38**, 4290–4309 (2019).
100. D. C. Norvell, M. L. Thompson, E. J. Boyko, G. Landry, A. J. Littman, W. G. Henderson, A. P. Turner, C. Maynard, K. P. Moore, J. M. Czerniecki, Mortality prediction following non-traumatic amputation of the lower extremity., *Br. J. Surg.* **106**, 879–888 (2019).
101. B. B. Nelson, R. N. Dudovitz, T. R. Coker, E. S. Barnert, C. Biely, N. Li, P. G. Szilagyi, K. Larson, N. Halfon, F. J. Zimmerman, P. J. Chung, Predictors of poor school readiness in children without developmental delay at age 2., *Pediatrics* **138** (2016), doi:10.1542/peds.2015-4477.

102. B. Zadrozny, C. Elkan, in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, (ACM Press, New York, New York, USA, 2002), p. 694.
103. X. Jiang, M. Osl, J. Kim, L. Ohno-Machado, Calibrating predictive model estimates to support personalized medicine., *J. Am. Med. Inform. Assoc.* **19**, 263–274 (2012).
104. O. V. Demler, N. P. Paynter, N. R. Cook, Tests of calibration and goodness-of-fit in the survival setting., *Stat. Med.* **34**, 1659–1680 (2015).
105. A. Niculescu-Mizil, Obtaining Calibrated Probabilities from Boosting, *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence* (2012).
106. X. Jiang, M. Osl, J. Kim, L. Ohno-Machado, Smooth isotonic regression: a new method to calibrate predictive models., *AMIA Jt Summits Transl Sci Proc* **2011**, 16–20 (2011).
107. M. Kantarcioğlu, J. Jin, C. Clifton, in *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, (ACM Press, New York, New York, USA, 2004), p. 599.
108. Y. Wu, X. Jiang, L. Ohno-Machado, Preserving Institutional Privacy in Distributed binary Logistic Regression., *AMIA Annu. Symp. Proc.* **2012**, 1450–1458 (2012).
109. M. J. Best, N. Chakravarti, Active set algorithms for isotonic regression; A unifying framework, *Math. Program.* **47**, 425–439 (1990).
110. D. J. Higham, Monotonic piecewise cubic interpolation, with applications to ODE plotting, *J Comput Appl Math* **39**, 287–294 (1992).
111. M. T. Goodrich, Randomized Shellsort, *J. ACM* **58**, 1–26 (2011).
112. K. Hamada, D. Ikarashi, K. Chida, K. Takahashi, Oblivious Radix Sort: An Efficient Sorting Algorithm for Practical Secure Multi-party Computation, *Cryptology ePrint Archive* (2014).
113. T.-H. H. Chan, Y. Guo, W.-K. Lin, E. Shi, Cache-Oblivious and Data-Oblivious Sorting and Applications, *SODA* (2018).
114. L. Ohno-Machado, Z. Agha, D. S. Bell, L. Dahm, M. E. Day, J. N. Doctor, D. Gabriel, M. K. Kahlon, K. K. Kim, M. Hogarth, M. E. Matheny, D. Meeker, J. R. Nebeker, pSCANNER team, pSCANNER: patient-centered Scalable National Network for Effectiveness Research., *J. Am. Med. Inform. Assoc.* **21**, 621–626 (2014).

115. K. B. Waghlikar, M. Mendis, P. Dessai, J. Sanz, S. Law, M. Gilson, S. Sanders, M. Vangala, D. S. Bell, S. N. Murphy, Automating installation of the integrating biology and the bedside (i2b2) platform., *Biomed. Inform. Insights* **10**, 1178222618777749 (2018).