Prediction-Constrained Markov Models for Medical Time Series with Missing Data and Few Labels

Preetish Rath[1], **Gabriel Hope**[2], **Kyle Heuton**[1], **Erik B. Sudderth**[2], and **Michael C. Hughes**[1]

[1]Dept. of Computer Science, Tufts University, Medford, MA, USA
[2]Dept. of Computer Science, University of California, Irvine, CA, USA

Abstract

When predicting outcomes for hospitalized patients, two key challenges are that the time series features are frequently missing and that supervisory labels may be available for only some sequences. While recent work has offered deep learning solutions, we consider a far simpler approach using the hidden Markov model (HMM). Our probabilistic approach handles missing features via exact marginalization rather than imputation, thereby avoiding predictions that depend on specific guesses of missing values that do not account for uncertainty. To add effective supervision, we show that a prediction-constrained (PC) training objective can deliver high-quality predictions as well as interpretable generative models. When predicting mortality risk on two large health records datasets, our PC-HMM's precision-recall performance is equal or better than the common GRU-D even with 100x fewer parameters. Furthermore, when only a small fraction of sequences have labels, our PC-HMM approach can beat time-series adaptations of MixMatch, FixMatch, and other state-of-the-art methods for semi-supervised deep learning.

1 Introduction

Predicting future patient outcomes from observed time series of labs and vitals in critical care settings has drawn significant recent research interest, catalyzed by the release of open-access datasets (Johnson et al., 2016; Pollard et al., 2018b), benchmark classification tasks (Purushotham et al., 2018; Harutyunyan et al., 2019; Wang et al., 2020b), and compelling clinical deployments (Sendak et al., 2020; Hyland et al., 2020). In this work, we consider two particular challenges in such models. First, labs and vitals are not densely sampled over time, leading to significant *feature missingness*. Second, for many tasks obtaining outcome labels to supervise training is prohibitively expensive, leading to *small labeled sets*. Both challenges require methods that inherently address uncertainty. We argue that simple probabilistic models – hidden Markov models (HMMs) (Rabiner & Juang, 1986; Ghahramani, 2001) with a few dozen states – have been under-explored yet provide a natural solution[1].

To address the problem of *feature missingness*, most efforts to date targeted at critical care time series consider two kinds of approaches: (1) a preprocessing step that imputes missing values via a simple heuristic like population mean filling or forward filling (Purushotham et al., 2018; Harutyunyan et al., 2019), followed by a standard classifier, or (2) a deep learning approach that learns to impute then predict in end-to-end fashion (Che et al., 2018; Yoon et al., 2018b; Cao et al., 2018). Unlike these approaches, HMMs elegantly handle missingness via exact marginalization, avoiding any imputation on the way to prediction. Recent work has also explored other *deep probabilistic* approaches to missingness that hybridize neural nets with Gaussian processes (Futoma et al., 2017), latent or stochastic differential equations (Rubanova et al., 2019; De Brouwer et al., 2019), Hawkes processes (Mei & Eisner, 2017), or encoder-decoder models (Li & Marlin, 2020). Unlike these approaches, HMMs do not need tens of thousands of parameters to make good predictions.

[1]Code URL: https://github.com/tufts-ml/pchmm-missing-data-limited-labels

Accepted paper at the Learning from Time Series for Health Workshop at NeurIPS 2022.

To address the *small labeled set* problem, one promising approach is to use semi-supervised learning (*SSL*) (van Engelen & Hoos, 2020), which trains models on the union of a small labeled set and a large unlabeled set of only features, no labels. While applications of SSL to critical care time series are limited, there are many tasks for which it would be applicable, such as automating preliminary diagnosis of disease. One line of work has produced effective SSL for end-to-end deep classifiers of *images* (Miyato et al., 2018; Berthelot et al., 2019; Sohn et al., 2020b). Recently, Goschenhofer et al. (2021) suggest these methods can be effective for multivariate time-series classification as well, assuming no features are missing. Unlike such deep SSL classifiers, HMMs as generative models can be easily trained on large unlabeled datasets with arbitrary feature missingness.

Contributions. To make HMMs succeed at clinical time-series prediction with feature missingness and small labeled sets, we cannot use off-the-shelf unsupervised or supervised approaches to training. The key methodological insight behind our work is *prediction constrained* (PC) training (Hope et al., 2021; Hughes et al., 2017). This optimization objective balances the HMM's useful generative properties (handling missingness via marginalization) with the essential need for its learned representations to enable effective prediction in the clinical task. Previous work by Hope et al. (2021) developed the PC-HMM but assumed no feature missingness. In this work, we show how PC training for HMMs offers a principled framework for effective prediction despite missing features *and* small labeled sets.

In careful experiments pursuing mortality prediction on two large health records datasets with significant feature missingness, we will show that PC training of HMMs of modest size can match or outperform much larger deep baselines designed for missingness such as GRU-D (Che et al., 2018) and BRITS (Cao et al., 2018). In the limited labels regime, we show gains against strong SSL baselines such as a time-series adaptations of MixMatch (Berthelot et al., 2019) and FixMatch (Sohn et al., 2020a). We further show how compact HMMs can be *interpreted* to yield useful clinical insights (Fig. 3). Our PC-HMM results suggest that simple, easy-to-train probabilistic approaches deserve consideration in future work. From the environmental costs of training deep models (Strubell et al., 2019) to avoiding expensive GPUs in bespoke hospital settings, there are many reasons to try the PC-HMM in clinical time-series applications with missing features or labels.

2 Prediction-Constrained HMMs

Here, we review the HMM generative model, how its resulting belief-state vector representations can be used for classification, and how prediction-constrained training can be used to balance discriminative and generative goals in settings with small labeled sets and feature missingness.

HMM generative model. The HMM is defined by a fixed number of states K, parameters governing *transitions* (outgoing probability vectors for all K states $\{\pi_k\}_{k=1}^K$ and the initial state π_0) and *emission* parameters $\{\phi_k\}_{k=1}^K$. Given (π, ϕ), the HMM defines a model for a sequence of regularly-spaced times of size T, where the random variables are the observable D-dimensional features over time $X \in \mathbb{R}^{T \times D}$ and the hidden discrete states $Z = [z_1, \dots z_T]$. The model $p(X, Z|\pi, \phi)$ decomposes as

$$p(Z|\pi) = \text{Cat}(z_1|\pi_0) \prod_{t=2}^{T} \text{Cat}(z_t|\pi_{z_{t-1}}), \quad p(X|Z, \phi) = \prod_{t,d} \mathcal{N}(x_{td}|\mu_{z_t,d}, \sigma^2_{z_t,d}). \tag{1}$$

In words, we draw Z from a first-order Markov model, where each z_t indicates one of the K states. Then, we generate each feature x_{td} given z_t via a state-specific emission model that *factorizes* over feature dimensions d and times t. Elegantly, this factorization assumption allows exact computation of the marginal of observed features under *any missingness pattern*. By applying the sum rule, the probability of the observed entries of X, denoted $X^{\mathcal{O}}$, given Z is $\prod_{t,d \in \mathcal{O}} \mathcal{N}(x_{td}|\mu_{z_t,d}, \sigma^2_{z_t,d})$.

Computing beliefs. We can obtain useful representations from the HMM even though we never observe Z by computing for each timestep t and state k the posterior belief of state occupancy: $b_{tk}(X^{\mathcal{O}}) \triangleq p(z_t = k|X^{\mathcal{O}})$. Efficient dynamic programming achieves this in $O(TK^2)$ runtime.

Predicting outcomes from beliefs. Beliefs represent sufficient statistics for the HMM's latent state at each timestep, and thus could be useful representations for classification. We condense all T belief vectors into an average occupancy vector $\bar{b} = [\bar{b}_1, \dots \bar{b}_K]$, where $\bar{b}_k = \frac{1}{T} \sum_{t=1}^{T} b_{tk}$. We can then build a simple linear classification model: $p(y = 1|X) = \sigma(\eta^T \bar{b}(X))$, where $\eta \in \mathbb{R}^K$ are the weight parameters and $\sigma(r) = \frac{1}{1+e^{-r}}$ is the logistic sigmoid function for real-valued inputs r.

Prediction constrained training. We wish to fit the parameters of the HMM to a labeled dataset $\mathcal{D}^L$ containing many pairs of feature time-series X and class label y as well as an optional unlabeled dataset $\mathcal{D}^U$ containing only X for many sequences. Prediction constrained training (Hope et al.,

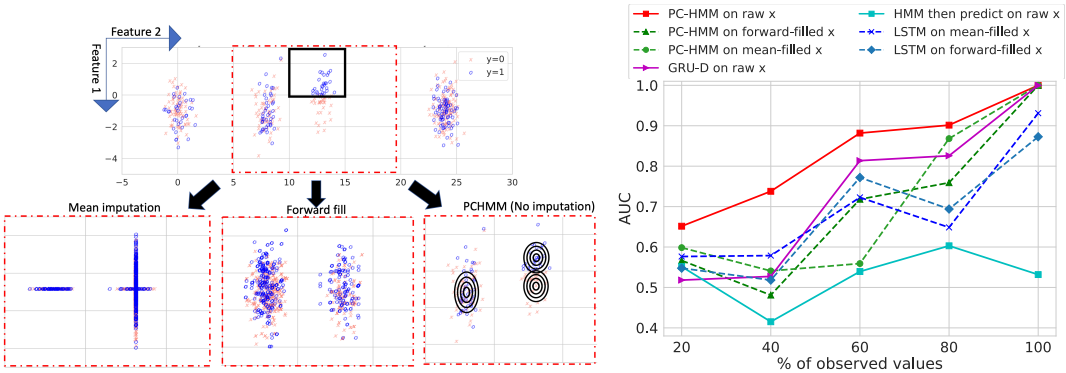

Figure 1: Demonstration of proposed PC-HMM on an "easy" binary classifier task where naive imputation fails. ***Top left:*** Visualization of all observed 2D features from 350 sequences (x-axis: feature 1 value, y-axis: feature 2), colored by labels and collapsed over time. Each sequence (T=8) comes from a Markov model that generates features from the 4 clusters shown in this 2D feature space. A sequence's class label depends on if *any observation* lands in the black box (yes means $y = 1$, no means $y = 0$). Without missingness, the two classes are clearly separable: compare black box to region below it. ***Bottom left:*** We sample features as missing with 40% probability i.i.d. across timesteps and features. We then apply mean-filling or forward-filling: each panel shows 2D values *after* imputation. Either strategy clearly destroys class separability. Our PC-HMM does not need imputation, and even at 40% missingness can learn to separate the black box from the region below it with separate Gaussian emission densities (black elipsoids). ***Right:*** AUROC vs. percentage of feature values observed. Across every tested percentage, our PC-HMM with compact state space (only 6 states, 80 total parameters) can outperform GRU-D or LSTM models even with many more parameters.

2021) aims to maximize generative performance but is unwilling to sacrifice prediction quality (the two aims are not treated equally). Our PC objective to minimize is

$$\mathcal{L}(\pi, \phi, \eta) \triangleq \sum_{X \in \mathcal{D}^L \cup \mathcal{D}^U} -\log p(X^{\mathcal{O}} | \pi, \phi) + \lambda \sum_{X, y \in \mathcal{D}^L} \ell(y, \sigma(\eta^T \bar{b}(X^{\mathcal{O}}; \pi, \phi))) \tag{2}$$

To emphasize the classification task's importance, we set tradeoff parameter $\lambda \gg 1$, selecting its specific value via a validation set. Two special cases are noteworthy: If we set $\lambda = 0$, we recover unsupervised training of the HMM via maximum likelihood. If we set $\lambda = 1$ and the loss ℓ is viewed as a negative Bernoulli likelihood over labels, we recover a "supervised" HMM that maximizes the joint likelihood of features and labels. See Hope et al. (2021) for further discussion, especially a formulation involving a constrained objective that better expresses our priority for prediction. Directly solving the constrained cost is impractical; thus we pursue the equivalent unconstrained objective. We optimize the objective via gradient descent. We can efficiently compute gradients because both terms are efficiently computed via well-known dynamic programming routines amenable to automatic differentiation. Our open-source code [1] uses the Tensorflow probability toolbox (Dillon et al., 2017).

3 Results and Discussion

Synthetic data demo. We analyze a toy dataset of many sequences generated by an HMM with $K = 4$ Gaussian clusters in a $D = 2$ feature space. The 2D features are illustrated in Fig. 1: Each sequence is labeled $y = 1$ only if any timestep's feature x_t enters the black square (more details in App. D.1). Fig. 1 (left) shows that even at 40% missingness (i.i.d. across features and timesteps) the PC-HMM with $K = 6$ learns separate states that cover the black box and its complement, and thus are *discriminative*. In contrast, both mean-filling and forward-filling imputation destroy the separability of red and blue classes. In Fig. 1 (right), at all missingness levels the PCHMM delivers better AUROC than alternatives, including an LSTM and the GRU-D. This dataset is class-balanced, thus AUROC is an appropriate metric. Later mortality prediction tasks are imbalanced (see Tab. C.1), so we use area under the precision recall curve (AUPRC) instead. All PC-HMM results, including "with mean-filled/forward-filled" baselines, assume diagonal covariances.

eICU experiments. To evaluate the PC-HMM on clinical tasks that require SSL with feature missingness, we predict in-ICU mortality after the first 24 hours on the eICU dataset (Pollard et al., 2018a) of vitals and labs (81% missing) from deidentified patient-stays at 59 critical care units throughout the U.S (details in App D.2). We train the model at various percentages p of labels available for training, approximately preserving the full dataset's label imbalance ($\sim 8.2\%$ mortality,

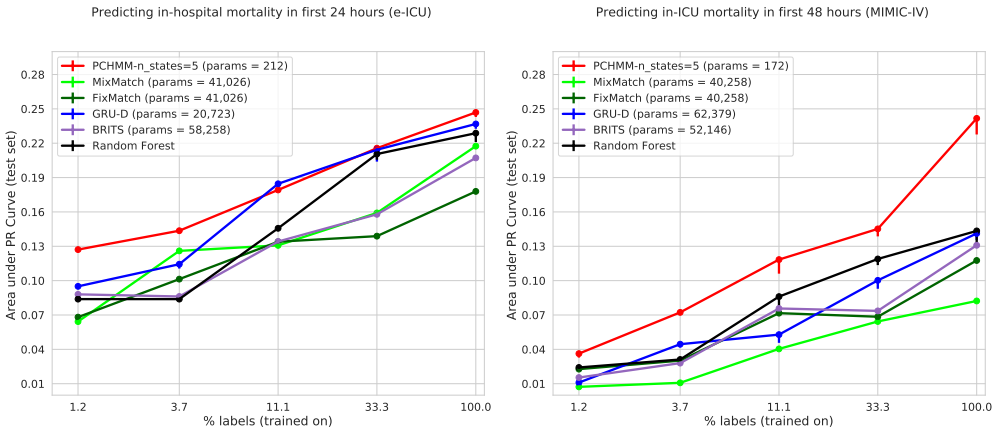

Figure 2: AUPRC (higher is better) versus amount of labeled data for in-hospital mortality early warning classifiers on two large EHR datasets (left: eICU, right: MIMIC-IV). X-axis: Percentage of all training sequences available with labels (SSL methods treat remaining sequences as unlabeled; other methods discard them). Y-axis: Area under precision-recall curve (AUPRC, higher is better). SSL methods, including our PC-HMM as well as MixMatch and FixMatch, learn from both labeled and unlabeled data. GRU-D, BRITS, and Random Forest use the labeled set only. The PC-HMM matches or beats the other models across all tested labeled set sizes, despite needing fewer parameters and only $1/10^{th}$ of the training time as the other models. App. D provides the train/valid/test splits and App. F gives hyperparameters for reproducibility.

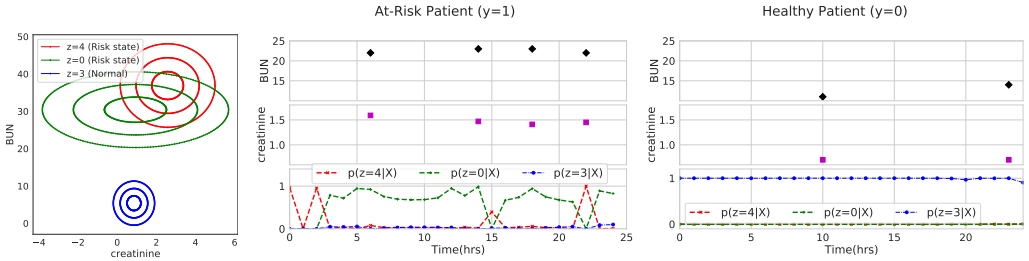

Figure 3: Interpretation of learned PC-HMM on eICU data. ***Left***: Gaussian emission distributions of states with highest and lowest η_k (predictor weights). The red and green states represent high risk (high η_k); the blue state indicates low risk. We show only 3 states and 2 features for clarity. High risk states seem to represent populations with high blood urea nitrogen (BUN) levels (above 20mg/dL) and high creatinine levels (above 1.45mg/dL), which could indicate kidney problems (Baha et al., 2021). ***Center***: For a patient labeled $y = 1$ (who eventually dies), high BUN and high creatinine are observed throughout the stay, and thus beliefs b_t give highest probability to high risk states (green and red). The probability of state 4 (red) peaks initially and towards the end of the stay because of high heart rate (> 150 bpm) and high blood glucose (> 170 mg/dL). ***Right***: For a patient labeled $y = 0$ with normal creatinine and BUN levels, our PC-HMM says the most likely state is low-risk (blue).

details in App Table C.1). Fig. 2 shows each method's test-set AUPRC (where $y = 1$ means death) as label availability p increases. Across all tested percentages p, the PC-HMM is competitive with deep alternatives, including BRITS and GRU-D as well as deep SSL such as FixMatch and MixMatch.

MIMIC-IV experiments. We further analyzed MIMIC-IV (Johnson et al., 2020), which contains over 60,000 de-identified ICU patient-stays from one hospital. In an effort to make the task more challenging, our preprocessing downsampled feature frequencies and death events (67% missingness, 1.2% mortality; details in App D.3 and App Table C.1). Fig. 2 shows again that PC-HMM results are quite competitive with deep learning baselines with 100x more parameters.

Interpreting learned PC-HMM models. Our PC-HMM framework enables us to interpret cohorts of vulnerable populations by visualizing the emission distributions for the states that have the highest predictor coefficients η_k. On eICU data, Fig. 3 shows our PC-HMM identifies states representing high blood urea nitrogen (BUN) and high creatinine (common indicators of kidney failure (Baha et al., 2021)) as the most vulnerable to in-ICU mortality. The model transitions to these 'high-risk' states as soon as high values of creatinine and BUN are observed in a patient who eventually dies.

Outlook. Our PC-HMM is a simple, effective way to address missingness in medical time series.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge computing hardware support from the U.S. National Science Foundation under grant NSF OAC-2018149. KH and MCH are partially supported by NSF IIS-1908617. GH and EBS are partially supported by NSF IIS-1816365.

References

Baha, A. D., Fendoglu, T. Z. I., Kokturk, N., Kilic, H., Hasanoglu, H. C., Arslan, S., Gulhan, M., Ogan, N., Akpinar, E. E., and Alhan, A. The effect of blood urea nitrogen/albumin ratio in the short-term prognosis of chronic obstructive pulmonary disease. *Erciyes Medical Journal*, 43(2): 184–189, 2021.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 2019. URL http://arxiv.org/abs/1905.02249.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. BRITS: Bidirectional Recurrent Imputation for Time Series. In *Advances in Neural Information Processing Systems*, 2018. URL https://papers.nips.cc/paper/2018/file/734e6bfcd358e25ac1db0a4241b95651-Paper.pdf.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1), 2018.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

De Brouwer, E., Simm, J., Arany, A., and Moreau, Y. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Futoma, J., Hariharan, S., and Heller, K. Learning to Detect Sepsis with a Multitask Gaussian Process RNN Classifier. In *International Conference on Machine Learning*, 2017. URL http://arxiv.org/abs/1706.04152.

Ghahramani, Z. An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Machine Intelligence*, 15(01):9–42, 2001.

Goschenhofer, J., Hvingelby, R., Rügamer, D., Thomas, J., Wagner, M., and Bischl, B. Deep semi-supervised learning for time series classification. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 422–428. IEEE, 2021.

Gupta, M., Phan, T.-L. T., Bunnell, H. T., and Beheshti, R. Concurrent imputation and prediction on ehr data using bi-directional gans: Bi-gans for ehr imputation and prediction. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 1–9, 2021.

Gupta, M., Gallamoza, B., Cutrona, N., Dhakal, P., Poulain, R., and Beheshti, R. An extensive data processing pipeline for mimic-iv. *arXiv preprint arXiv:2204.13841*, 2022.

Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G., and Galstyan, A. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019.

Hope, G., Hughes, M. C., Doshi-Velez, F., and Sudderth, E. B. Prediction-Constrained Hidden Markov Models for Semi-Supervised Classification. In *Time Series Workshop at ICML 2021*, 2021. URL https://www.michaelchughes.com/papers/HopeEtAl_TimeSeriesWorkshopAtICML_2021.pdf.

Hughes, M. C., Weiner, L., Hope, G., McCoy Jr., T. H., Perlis, R. H., Sudderth, E. B., and Doshi-Velez, F. Prediction-Constrained Training for Semi-Supervised Mixture and Topic Models. *arXiv:1707.07341 [cs, stat]*, July 2017. URL http://arxiv.org/abs/1707.07341. arXiv: 1707.07341.

Hyland, S. L., Faltys, M., Hüser, M., Lyu, X., Gumbsch, T., Esteban, C., Bock, C., Horn, M., Moor, M., et al. Early prediction of circulatory failure in the intensive care unit using machine learning. *Nature Medicine*, 26(3):364–373, 2020.

Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021)*, 2020.

Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035, May 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.35. URL https://www.nature.com/articles/sdata201635. Number: 1 Publisher: Nature Publishing Group.

Li, S. C.-X. and Marlin, B. M. Learning from Irregularly-Sampled Time Series: A Missing Data Perspective. In *International Conference on Machine Learning*, 2020.

Luo, Y., Cai, X., Zhang, Y., Xu, J., et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

Mei, H. and Eisner, J. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process, 2017. URL http://arxiv.org/abs/1612.09328.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Pollard, T. J., Johnson, A. E., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018a.

Pollard, T. J., Johnson, A. E. W., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1):180178, September 2018b. ISSN 2052-4463. doi: 10.1038/sdata.2018.178. URL https://www.nature.com/articles/sdata2018178. Number: 1 Publisher: Nature Publishing Group.

Purushotham, S., Meng, C., Che, Z., and Liu, Y. Benchmarking deep learning models on large healthcare datasets. *Journal of Biomedical Informatics*, 83:112–134, 2018.

Rabiner, L. R. and Juang, B.-H. An introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3 (1):4–16, 1986.

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. Latent ODEs for Irregularly-Sampled Time Series. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 11, 2019. URL https://arxiv.org/abs/1907.03907.

Sendak, M. P., Ratliff, W., Sarro, D., Alderton, E., Futoma, J., Gao, M., Nichols, M., Revoir, M., Yashar, F., et al. Real-World Integration of a Sepsis Deep Learning Technology Into Routine Clinical Care: Implementation Study. *JMIR Medical Informatics*, 8(7), 2020.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020a.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., Li, C.-L., et al. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020b. URL https://proceedings.neurips.cc/paper/2020/file/06964dce9addb1c5cb5d6e3d9838f733-Paper.pdf.

Strubell, E., Ganesh, A., and McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, Florence, Italy, 2019. Association for Computational Linguistics.

van Engelen, J. E. and Hoos, H. H. A survey on semi-supervised learning. *Machine Learning*, 109 (2):373–440, 2020.

Wang, S., McDermott, M. B., Chauhan, G., Ghassemi, M., Hughes, M. C., and Naumann, T. Code for eicu extract. https://github.com/MLforHealth/MIMIC_Extract/tree/eICU_Extract, 2020a.

Wang, S., McDermott, M. B. A., Chauhan, G., Hughes, M. C., Naumann, T., and Ghassemi, M. MIMIC-Extract: A Data Extraction, Preprocessing, and Representation Pipeline for MIMIC-III. *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 222–235, April 2020b. doi: 10.1145/3368555.3384469. URL http://arxiv.org/abs/1907.08322. arXiv: 1907.08322.

Yoon, J., Jordon, J., and Schaar, M. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pp. 5689–5698. PMLR, 2018a.

Yoon, J., Jordon, J., and van der Schaar, M. GAIN: Missing Data Imputation using Generative Adversarial Nets. In *International Conference on Machine Learning*, 2018b. URL http://proceedings.mlr.press/v80/yoon18a.html.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhu, Z., Luo, T., and Liu, Y. The Rich Get Richer: Disparate Impact of Semi-Supervised Learning. *International Conference on Learning Representations (ICLR)*, 2022.

A Prediction Tasks

A.1 Sequence classification given multivariate time-series

Our clinical classification tasks assume access to a labeled training dataset $\mathcal{D}^L$ of feature-time-series, outcome pairs: $\{\mathcal{S}, y\}$. Each binary label $y \in \{0, 1\}$ indicates the outcome of interest. For our applications, y indicates in-ICU mortality; that is, eventual death during the hospital stay. Each *irregular* multivariate time series $\mathcal{S}$ contains zero or more pairs $\tau_{d\ell}, v_{d\ell}$ of timestamps and values for each of D features indexed by d (for our applications, these are vital signs or labs).

We can transform $\mathcal{S}$ into a regular time series $X = [x_1, x_2, \ldots x_T]$ with time indices t spaced a fixed duration apart (our applications use 8 hours for MIMIC-IV and 1 hour for eICU), where $x_t \in \mathbb{R}^D$ denotes the feature vector for timestep t. When vitals and labs are not densely sampled, any pattern of missingness in X is possible. We denote the *observed* subset of X as $X^{\mathcal{O}}$.

A.2 Semi-supervised sequence classification.

In SSL tasks, we assume we have both a labeled dataset $\mathcal{D}^L$ as well as a (much larger) unlabeled dataset $\mathcal{D}^U$ of time-series features only. Each of these time series may exhibit significant feature missingness too.

B Baseline Methods

B.1 Deep Semi-supervised learning adapted to Time-Series

Recent progress in improving deep classifiers in the SSL setting has been made under the family of *consistency regularization*. Following the unified analysis in Zhu et al. (2022), these approaches train a deep network f_θ with weights θ to minimize a two-term loss:

$$\sum_{X,y \in a(\mathcal{D}^L)} \ell(y, f_\theta(X)) + \lambda \sum_{X \in a(\mathcal{D}^U)} \ell(y'(X), f_\theta(X)) \tag{3}$$

Here, $\ell(\cdot, \cdot)$ is a loss function, $\lambda > 0$ is a tradeoff hyperparameter, $y'(X)$ is a labeling transformation, and $a(\cdot)$ represents an (optional) data augmentation transformation. Below, we describe how both PseudoLabel and MixMatch fit this framework via concrete realizations of y', a, and ℓ.

In practice, to minimize this loss via minibatch gradient descent we start λ at zero for a few hundred epochs, and gradually ramp up to a small positive value over a few hundred more epochs. This ensures that early learning fits well to the labeled set, while letting the unlabeled set also influence results later on.

MixMatch for time series. MixMatch (Berthelot et al., 2019) is a recent state-of-the-art SSL algorithm based on two key ideas. First, smooth transitions between classes in feature space are desirable and achievable via an interpolating augmentation scheme known as MixUp (Zhang et al., 2017). Second, it is useful to ensure consistency in the predicted label across multiple augmentations of the same source features. Originally designed for images, it has recently been applied to time series (Goschenhofer et al., 2021).

We set the labeling function $y'(X)$ to produce temperature-sharpened probability vectors averaged across multiple augmentations of examples X. The augmentation transformation $a(\cdot)$ uses MixUp to interpolate between labeled examples (see (Berthelot et al., 2019) for details). As a basic augmentation procedure applicable to time series, we add Gaussian noise $\mathcal{N}(0, \epsilon^2)$, following Goschenhofer et al. (2021), with standard deviation ϵ set to 0.1 and 1. For the backbone architecture f_θ, we use a Gated Recurrent Unit(GRU) (Cho et al., 2014).

FixMatch for time series. Similar to MixMatch, FixMatch (Sohn et al., 2020a) is another recent state-of-the-art SSL algorithm based on consistency regularization and pseudo-labeling. Pseudo-labels for weakly augmented unlabeled examples are only retained if the model produces high-confidence predictions (we try 0.6 and 0.8 as thresholds for high confidence predictions). Consistency regularization and the augmentation procedure is the same as MixMatch. Again, we use GRU for the backbone architecture.

B.2 Random Forest Adapted to time series

To train the random forest, each patient-stay's multivariate time-series is transformed into a feature vector as follows. For each raw feature, we apply 7 summarization functions (missing indicator; time since last non-missing; plus min, max, median, slope, and variance of non-missing values). The resulting feature vector size for MIMIC-IV is 72 (10 vitals x 7 summary functions + 2 demographics) and for eICU is 101 (14 labs and vitals x 7 summary functions + 3 demographics). Demographics for each patient stay are appended only after the summary functions are computed for each of the time-varying measurements.

C Related Work

C.1 Previous work on PC training for HMMs

Hope et al. (2021) previously introduced the PC-HMM and showed some success in semi-supervised classification on a variety of domains, including one healthcare application. However, that work did not handle missing observations probabilistically and required forward-fill imputation to preprocess ICU time series. Here, we provide a robust solution that does not require imputing missing data, and compare to more competitive baselines for time-series SSL (such as MixMatch and FixMatch) as well as deep methods designed for feature missingness such as GRU-D (Che et al., 2018).

C.2 Generator Adversarial Networks (GAN) adapted to mutivariate time-series for imputation and prediction

Luo et al. (2018), Yoon et al. (2018a) and Gupta et al. (2021) propose methods adapting generative adversarial nets (GAN) to multivariate time-series with missing observations. These methods rely on a 2 stage process of imputing missing observations using a generator and discriminator first, and then using the imputed values for downstream prediction. On the other hand, the prediction constrained framework of our PC-HMM enables us to maximize the HMM's generative performance, while simultaneously enabling effective prediction in the downstream clinical task.

Dataset	% positive labels at various p(% labeled sequences)				
	$p = 1.2\%$	$p = 3.7\%$	$p = 11.1\%$	$p = 33.3\%$	$p = 100\%$
eICU	6.9%	8.0%	8.2%	8.2%	8.2%
MIMIC-IV	0.7%	1.0%	1.1%	1.2%	1.2%

Table C.1: Class imbalance at various percentages of labeled sequences for eICU and MIMIC-IV

		seconds/epoch to train each model				
Dataset		PC-HMM	GRU-D	MixMatch	FixMatch	BRITS
MIMIC-IV	$\Delta T = 8$ hr	4	41	64	55	48
eICU	$\Delta T = 1$ hr	10	70	115	90	75

Table E.1: Computation time (seconds/epoch) required by each model. An epoch is completed when every example in the training set is covered at-least once. Although computation time increases when measurements are more frequent, the PC-HMM still requires far lesser time to train due to significantly lesser parameters than the other models. Note that we perform experiments with $\Delta T = 8$ hr on MIMIC-IV to highlight that PC-HMMs perform well even in smaller hospital datasets where hourly measurements might be unavailable.

D Data and Pre-processing Details

D.1 Toy Data Generation

The toy data is generated from a Markov model with 4 states (states 0, 1, 2, 3) with the means of the emission distributions located at $(0, -1), (8, -1), (13, 0)$ and $(24, 0)$. The variance in x is 0.3 and variance in y is 1 for each of the clusters. Each sequence is initialized at state 0 and either remains in the current state or transitions to the next adjacent state with equal probability. Once state 2 is reached, it immediately transitions to state 3 in the next time-step and stays there for the remaining time-steps with probability 1. We simulate 350 sequences, each of length $T = 8$ from this model. A sequence is labeled as 1 if it enters the highlighted black box at any time-point.

D.2 eICU

eICU contains data from 59 critical care units throughout the U.S. For each patient-stay, we extract 3 demographics, 8 vitals, and 6 lab measurements discretized to hourly bins using eICU Extract (Wang et al., 2020a). Our train/valid/test splits have 43642/14509/14518 patient-stays, with $\sim 8.2\%$ resulting in death and 81% missing entries.

D.3 MIMIC-IV

We use 10 time-varying vital sign features, as well as age and gender features known at admission. We used the data pipeline developed by Gupta et al. (2022) to extract the chart-events and outcomes. In an effort to improve eventual transportability, our preprocessing downsampled vital sign frequencies and death events. Our train/valid./test splits have 42836/15443/15802 patient-stays, with $\sim 1.2\%$ patient stays resulting in death and 67% missing entries in the hourly time series.

Pre-processing vitals While MIMIC-IV contains vitals that are frequently sampled ($<$1 hour between measurements), we chose to downsample the vitals to 1 measurement taken every $8 - 16$ hours, to reflect health records in many general hospitals without automated data collection.

Lowering outcome rate Roughly 10% of ICU admissions result in death in MIMIC-IV. We instead lower the outcome rate to 1.2% (by excluding some ICU deaths) to ensure applicability of our model to a general hospital population rather than ICU patients only, where the outcome rate might be a lot lesser.

E Training Time

The training time for each model is shown in Table E.1.

Model	Hyperparameters searched	Selection criterion
PC-HMM	• learning rate : $0.001, 0.01, 0.1$ • states : $5, 10, 20$ • initialization seeds : 20 • l2 penalty : 10 log-spaced values between 10^-6 and 1 • λ : $50, 100, 500, 1000, 5000$	Best validation AUPRC
GRU-D	• learning rate : $0.001, 0.01, 0.1$ • hidden units : $32, 64, 128$ • hidden layers : $1, 2$ • initialization seeds : 20 • l2 penalty : 7 log-spaced values between 10^-6 and 100 • batch-size : $128, 256, 512$	Best validation AUPRC with early stopping
BRITS	• learning rate : $0.001, 0.01, 0.1$ • hidden units : $32, 64, 128$ • initialization seeds : 20 • hidden layers : $1, 2$ • dropout : $0, 0.1, 0.25$ • batch-size : $128, 256, 512$ • imputation-weight : $0.3, 0.6$ • label weight : 1.0	Best validation AUPRC with early stopping
FixMatch	• learning rate : $0.001, 0.01, 0.1$ • hidden units : 16, 32, 64 • initialization seeds : 20 • hidden layers : $1, 2$ • batch-size : $128, 256, 512$ • l2 penalty : 8 log-spaced values between 10^-8 and 10 • pseudo-label thresholds : $0.6, 0.8$ • augmentation noise variance : $0.1, 0.01$ • unlabeled strength : $50, 75$ • Temperature scaling : 0.5 • α(for MixUp): 0.75	Best validation AUPRC with early stopping
MixMatch	Same hyperparameter grid as FixMatch except no psuedo-label thresholds	Best validation AUPRC with early stopping
Random Forest	• Fraction of features : $0.33, 0.66, 1.0$ • Max leaf nodes : 32, 64, 128 • Min Samples per leaf : 64, 128, 512, 1024 • Estimators : 64	Best validation AUPRC

Table F.1: Hyperparameters for all the models

F Hyperparameters

The hyperparameters for all the experiments are shown in Table F.1. For GRU-D, Fixmatch and Mixmatch, the missing values are forward filled in time and then imputed with the population mean. For BRITS, the missing values initialized to 0 (as recommended in the paper). Except for the PC-HMM, the inputs to all the models are dimension-wise z-scored.

G Ablation

The importance of training the PC-HMM with high λ values are shown in Table G.1. Performance is better at $\lambda > 1$.

		AUPRC at various p(% labeled sequences)				
Dataset		$p = 1.2\%$	$p = 3.7\%$	$p = 11.1\%$	$p = 33.3\%$	$p = 100\%$
MIMIC-IV	$\lambda = 1$	0.031	0.046	0.087	0.097	0.178
	$\lambda > 1$	**0.040**	**0.072**	**0.118**	**0.145**	**0.242**
eICU	$\lambda = 1$	0.115	0.132	0.143	0.199	0.219
	$\lambda > 1$	**0.127**	**0.144**	**0.179**	**0.216**	**0.248**

Table G.1: Impact of λ on test AUPRC for MIMIC-IV and eICU. Performance is better at $\lambda > 1$, highlighting the significance of higher λ in our prediction constrained objective. The grid of λ values searched are shown in Table F.1