Towards Dynamic Feature Acquisition on Medical Time Series by Maximizing Conditional Mutual Information

Fedor Sergeev [1] **Paola Malsot** [1 2] **Gunnar Rätsch** [1] **Vincent Fortuin** [3 4 5]

Abstract

Knowing which features of a multivariate time series to measure and at what time is a key task in medicine, wearables, and robotics. Better acquisition policies can reduce costs while maintaining or even improving the performance of downstream predictors. Inspired by the maximization of conditional mutual information, we propose an approach to train acquirers end-to-end using only the downstream loss. We show that our method outperforms random acquisition policy, is close to the performance of a model with an unrestrained budget, but can't match a static acquisition strategy (likely due to the simplicity of its architecture). We highlight the assumptions and outline avenues for future work.

1. Introduction

In the medical setting, clinicians often need to monitor patients over time during their hospital stay, especially in Intensive Care Units (ICUs; Hyland et al., 2020). They try to improve the patient's state by administering drugs while relying on continuous measurements of vital signs (e.g., heart rate) and occasional lab tests (e.g., blood test, X-ray). While the continuous measurements are automatic and practically free, performing lab tests takes the clinical staff's time and incurs additional costs. We aim to develop a method for recommending which lab tests to perform, in order to best monitor the patient's state, while decreasing workload and costs.

More formally, the hospital stay of a patient i can be represented as a multivariate (or even multi-modal) time series $\boldsymbol{x}^i = \{x^i_{t,f}\}$ with the features f at time t being the values of either vital signs, lab tests, or administered drugs. Usually, these data are used for time series classification (e.g., predict hospital readmission in the next five years, based on the entire stay) (Hyland et al., 2020), early event prediction (e.g., at each moment of the stay, predict kidney failure in the next 8 hours) (Yèche et al., 2023; Kuznetsova et al., 2023), or intervention recommendation (Liu et al., 2020).

We consider the *dynamic feature acquisition* (DFA) task — based on the observed patient state $\{x^i_{t,f}\}_{t\leq\tau}$ at time τ, recommend which feature(s) f should be measured at some future time τ' at known cost $c_{\tau,f}$ (see Figure 1). The aim is to reduce the total measurement cost $\sum_{t,f} c_{t,f}$ while keeping the performance of the downstream predictor the same or even improving it.

Note that DFA is also relevant for other application domains. For example, in wearable devices, activating sensors consumes battery power. Consequently, DFA can improve the battery life of these devices (Possas et al., 2018; Merrill et al., 2023). Similarly, DFA applies to active perception (Bajcsy et al., 2018) and computationally efficient video classification (Yang et al., 2022) problems.

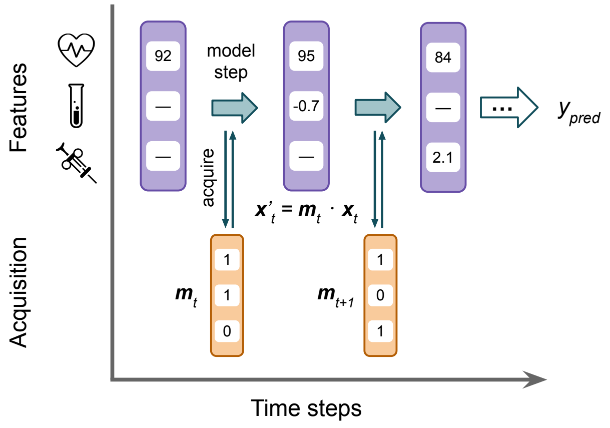


Figure 1. Sketch of DFA on a regular time series in medicine[2].

While dynamic feature acquisition has been considered for static data (Ma et al., 2019; Covert et al., 2023; Yu et al., 2023), to the best of our knowledge, there is only one previous work that considered it on time series (Kossen et al., 2023). Unlike Kossen et al. (2023), we do not use reinforce-

[1]ETH Zurich, Zurich, Switzerland [2]ETH AI Center, ETH Zurich, Switzerland [3]Helmholtz AI, Munich, Germany [4]Technical University of Munich, Germany [5]Munich Center for Machine Learning, Germany. Correspondence to: Fedor Sergeev <fedor.sergeev@inf.ethz.ch>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria. Copyright 2024 by the author(s).

ment learning. Instead, we greedily select the next features using conditional mutual information (CMI; Covert et al., 2023). This allows for end-to-end training, using only the loss from the downstream task. Additionally, our method is recurrent, allowing for clinically relevant downstream tasks, including time series classification and early event prediction.

Our contributions are:

- We propose a novel CMI-based approach for dynamic feature acquisition. It is compatible with clinically-relevant downstream prediction tasks, does not use reinforcement learning, and can be trained end-to-end.
- We test on benchmark time series classification datasets with fake features and show that our method outperforms random, but falls short behind static selection methods.

2. Problem Setting

Let us assume that the time series are *regular*, indexed with $t \in \{0, \dots, T^i\}$, where T^i is the length of the series $\boldsymbol{x}^i$. We consider the case when an acquisition recommendation is made for features that will become available at the next time step (*"next-step" assumption*): $\tau' = \tau + 1$. This is the case when the time it takes to measure requested features is smaller than the time step. Both the regularity and the "next-step" assumptions are plausible for ICU when a bigger resolution (e.g., one hour) is chosen (Hyland et al., 2020).

For simplicity, we assume that the *measurement cost is constant* over time and features: $\forall t, f \hookrightarrow c_{t,f} =: c$. Without loss of generality, we set $c = 1$. Similarly to Kossen et al. (2023), we assume that the *data are fully observed*. Both of these assumptions do not hold for medical data, and we leave generalization to future work.

The cost of performed acquisitions should be below a given budget. For static data, it is enough to know a budget per sample, but for time series, we could also consider a budget per time step $b(\boldsymbol{x}_t, t)$. In a general setting, the budget per time step should be predicted by the acquirer. In our experiments, we consider a simplified setting, where the time step budget is constant and given a priori: $b(\boldsymbol{x}_t, t) = b$.

The acquisition and prediction cycle under these assumptions is shown in Figure 1 and Algorithm 1.

Here, the acquirer is a model that outputs the *acquisition vector* $\boldsymbol{m}_\tau$ at each time step τ. It is a binary vector with ones indicating which features should be acquired at the next time step $\tau + 1$. Since the data are fully observed, we imitate the measurement procedure with an element-wise product. The measured data are then passed to the classifier.

[2]Icons: Rockicon, Dilich, Lorc, CC BY 3.0 https://creativecommons.org/licenses/by/3.0, via Wikimedia Commons.

Algorithm 1 Next-step DFA on regular time series

```
Input: time series $\boldsymbol{x}$ of length $T$, time step budget function $b$, acquirer, classifier
$t = 0$
$\boldsymbol{m}_0$ = acquirer.init()    // initial acquisition request
while $t < T$ do
    $\boldsymbol{x}'_t = \boldsymbol{m}_t \cdot \boldsymbol{x}_t$    // measure requested features
    classifier.step($\boldsymbol{x}'_t, \boldsymbol{m}_t, t$)
    $\boldsymbol{m}_{t+1}$ = acquirer.step ($\boldsymbol{x}'_t, \boldsymbol{m}_t, b(\boldsymbol{x}_t, t), t$)
    $t = t + 1$
end while
$y_{pred}$ = classifier.predict()    // make the prediction
$C = \sum_{t=0}^{T} \boldsymbol{m}_t$    // calculate the cost
```

If the classifier receives new data at each time step, it can be used for both time series classification and early event prediction. Note that this does not limit the architecture of the classifier to only recurrent models. Indeed, the classifier may store the values of the features it receives and then apply any architecture that is suitable for variable-length data (e.g., a transformer from Vaswani et al. (2017)).

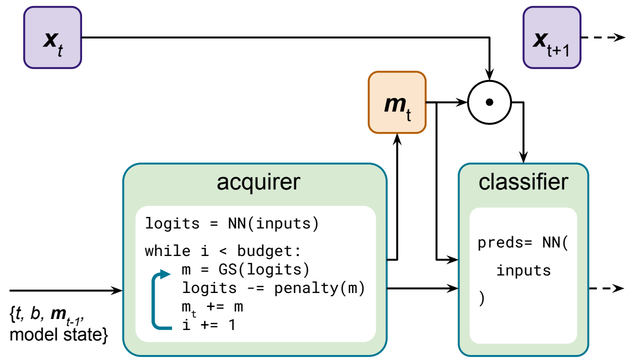
Note that the acquirer and classifier may have access to each other's internal state. For example, if they are implemented with recurrent neural networks, they could receive each other's hidden states as input. We do not specify this in Algorithm 1 to avoid notation clutter. From this point on, for simplicity, we consider only classification objectives.

3. Method

DFA usually follows one of two approaches: use the cost estimate as a penalty function to train the acquisition model using reinforcement learning (Kossen et al., 2023; Yu et al., 2023), or use some acquisition function to rank and select the most meaningful features (Ma et al., 2019; Covert et al., 2023). The latter approach often uses CMI. Estimating CMI directly (e.g., using a partial variational autoencoder) can be challenging (Ma et al., 2019), so instead, CMI can be approximated (Covert et al., 2023).

In Covert et al. (2023), the authors use a neural network and Categorical distribution to sequentially predict the feature with the largest CMI, and perform (greedy) selection on static data.

Similarly, we use the *acquirer* neural network to predict logits of the approximate CMI at each time step of the



```
logits = NN(inputs)
while i < budget:
    m = GS(logits)
    logits -= penalty(m)
    m_t += m
    i += 1
```

```
preds= NN(
  inputs
)
```

Figure 2. Proposed acquisition + classification mechanism.

time series (see Figure 2). We then iteratively (until the budget b is reached) sample a one-hot vector indicating the selected feature using the Gumbel-Softmax (GS; Jang et al., 2016). To avoid selecting the same feature twice, we subtract a penalty vector from the acquirer prediction. Further details are available in Appendix B.1. This approach is differentiable and therefore can be trained end-to-end with the classifier using backpropagation.

4. Experiments

4.1. Setup

We test the proposed method on the *FordA* and *SpokenArabicDigits* datasets from the UCR and UEA time series classification archives (Dau et al., 2018; Bagnall et al., 2018). The datasets' summary and samples are shown in Table A.1 and Figure A.1.

For both datasets, we consider balanced classification tasks (binary and multiclass respectively) and use accuracy as the performance metric. By default, no features are considered observed; they all have to be explicitly acquired.

The FordA dataset is univariate, so, to imitate a multivariate dataset, we take $m = 10$ consecutive time steps from FordA and set them as one time step with m features of a new *m-FordA* dataset (short for multivariate or multi-step FordA). In contrast, SpokenArabicDigits is multivariate and variable length by design.

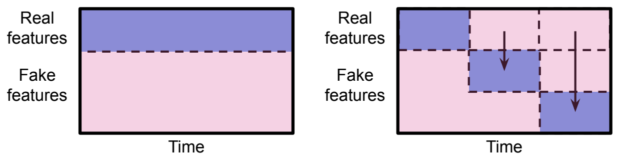
We use a much simpler training procedure compared to Covert et al. (2023): the temperature in the Gumbel distribution is fixed, and we do not pre-train the classifiers.

We implement the acquirer using a fully connected neural network, concatenating the previous observation, acquisition mask, and current time step into one vector. Note that we do not pass the internal classifier state to the acquirer. The classifiers are implemented using long short-term memory networks (LSTMs; Hochreiter & Schmidhuber, 1997). We train using the Adam optimizer (Kingma & Ba, 2014) with cross-entropy loss in PyTorch (Paszke et al., 2017). Additionally, we use scikit-learn for generating samples from a Gaussian process and training a random forest (Pedregosa et al., 2011).

4.2. Fake features

The features in the datasets we consider are quite similar (e.g., the same univariate feature for FordA, and different audio frequencies measured by the same device in SpokenArabicDigits). Therefore, due to correlations, it might be hard to tell whether a specific feature is more informative than another.

In order to reliably test whether our model learns to dynamically acquire reasonable features, we add fake features (see Figure 3a). Since the added features do not hold any information about the class label, a proper acquisition policy will ignore them.



(a) Normal (b) Shifted

Figure 3. Addition of fake features to the data.

We try three different variations of fake features: zeros, Gaussian noise (with 0 mean and 0.5 standard deviation), and samples from a Gaussian process (GP) with an RBF kernel (amplitude coefficient 0.5, length scale 1.5, length scale bounds $[0.1, 10]$). The parameters of the Gaussian distribution and GP are selected so that the fake features visually resemble the real ones. The examples of modified data are shown in Figures A.2 and A.3.

We set a constant budget per time step of $b = 5$ and compare our method to a random acquisition policy (selects b features at random at each step) and a "complete" acquisition policy (selects all features at each step). This means that ours and the random acquirers obtain 8 times fewer features than the complete acquirer.

We add 30 fake features of different types and train for a set number of epochs (500 and 1000 for the two datasets, respectively). The results for m-FordA and SpokenArabicDigits are presented in Tables 1 and 2.

Our acquirer consistently outperforms the random acquisition policy, and sometimes even matches the performance of the complete acquirer. We also notice that in some cases, the complete acquirer exhibits overfitting, while our acquirer avoids it (e.g., in the case of noise fake features on m-FordA,

Acquirer	Type of fake features		
	Zeros	Noise	GP
Random	0.76 ± 0.04	0.74 ± 0.04	0.73 ± 0.02
Ours	0.86 ± 0.03	0.86 ± 0.03	0.84 ± 0.02
Complete	0.93 ± 0.00	0.84 ± 0.02	0.80 ± 0.03

Table 1. Test classification accuracy (%) on m-FordA with 30 fake features of various types for different acquirers. Accuracies are reported as mean ± standard deviation, calculated over 5 different seeds.

Acquirer	Type of fake features		
	Zeros	Noise	GP
Random	0.84 ± 0.06	0.82 ± 0.05	0.87 ± 0.02
Ours	0.87 ± 0.05	0.90 ± 0.03	0.91 ± 0.04
Complete	0.90 ± 0.06	0.91 ± 0.03	0.90 ± 0.03

Table 2. Test classification accuracy (%) on Spoken Arabic Digits with 30 fake features of various types for different acquirers. Accuracies are reported as mean ± standard deviation, calculated over 4 different seeds.

shown in Figure B.1).

The examples of acquisition patterns are presented in Figure 4. It is easy to see that our acquirer starts selecting the real features (notice the horizontal lines). While far from perfect, it is a promising result, especially accounting for the simplistic architecture and training setup (e.g., fixed Gumbel temperature, no classifier pretraining).

4.3. Shifted fake features

Next, we investigate whether the learned policy is truly dynamic. As a static feature selection baseline, we use a random forest (RF), as it is often used for feature importance analysis (e.g., in Hyland et al. (2020) for ICU features).

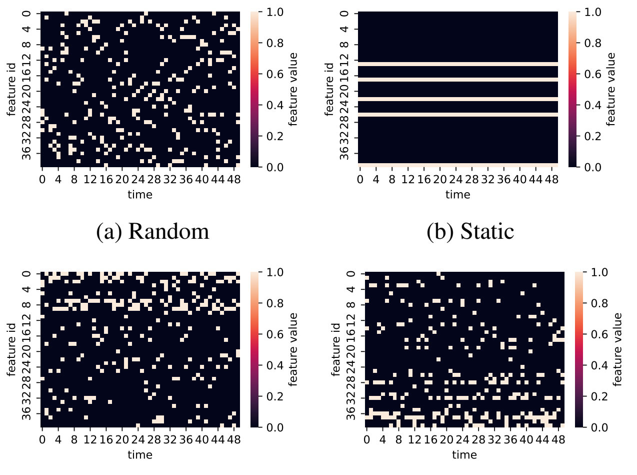
We modify the fake feature datasets by shifting the real features so that the acquisition pattern would have to change over time (see Figure 3b). If the number of the real features is R and the number of fake features is F, the indices i of real features will shift to $i + R$ every $\left\lfloor \frac{R}{R+F} \right\rfloor \cdot T$ time steps. For example, for m-FordA with $m = 10$ with 20 fake features, the real features will have indices 0 to 10 during the first third of the time steps, 10 to 20 during the second third, and 20 to 30 for the rest of the series.

The dynamic policy should be able to learn that shift, while a static acquirer will only select the same set of features throughout the time series.

The results and the acquisition patterns are shown in Table 3 and Figure 4. Our acquirer outperforms the random policy, but is outperformed by the static policy. The acquisition pattern shows that the model does not manage to capture the shift in fake features.

Acquirer	Accuracy, %
Random	0.708
Static (RF)	0.842
Ours	0.740
Complete	0.897

Table 3. Test accuracy for classifiers trained with various acquirers on m-FordA with 30 shifted fake features (zeros).



(a) Random (b) Static

(c) Learned (d) Learned (shifted features)

Figure 4. Acquisition patterns on m-FordA with 30 fake features (zeros). In (a), (b) and (c), the real features occupy the top ten rows, whereas in (d) they are shifted to fill a block diagonal pattern.

We hypothesize that the underperformance of our acquirer is due to its simplistic architecture. Although time is passed to the acquirer, it does not use the hidden state of the classifier. A more sophisticated architecture (e.g., an LSTM) that receives a classifier state as input will likely perform better.

5. Related work

To the best of our knowledge, the only prior work that has considered dynamic feature acquisition on time series data is Kossen et al. (2023). They use reinforcement learning and focus on multimodal data.

Feature acquisition on static data has received wider attention. Both methods using mutual information (Ma et al., 2019; Lewis et al., 2021; Covert et al., 2023) and reinforcement learning (Yu et al., 2023) have been developed.

In Ma et al. (2019), CMI is estimated by training a partial variational autoencoder (P-VAE). This allows the model to perform imputation from any subset of observed features and select the features associated with high-value information. In Lewis et al. (2021), this approach has been developed further with the use of transformers for processing sets of observed features. The main challenge with using

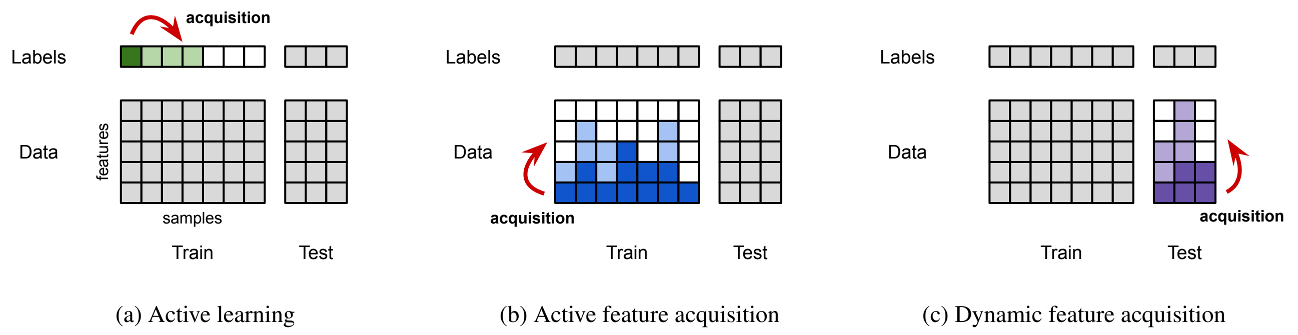


Figure 5. Comparison of active learning, active and dynamic feature acquisition tasks on static data.

the P-VAE is its training. Training generative models can be challenging, especially for more complex data such as images (Covert et al., 2023).

An alternative approach presented in Covert et al. (2023) aims to approximate CMI instead of estimating it precisely. They propose using a Categorical distribution and greedily select the feature with the largest CMI at each step. Unlike Ma et al. (2019), they use only simple (dense) architectures. However, their approach accepts set-based models as well.

For static ICU data, deep reinforcement learning has been used for DFA training (Yu et al., 2023). The authors took into account that medical tests are usually done in panels (i.e., provide multiple features at the same time) and differ in cost. They also produce the accuracy-cost Pareto fronts, which help analyze the trade-off made when setting a specific acquisition budget.

Dynamic feature acquisition using CMI is closely related to active learning and active feature acquisition (see Figure 5). Recent works show that Bayesian models can perform well in active learning (Sharma et al., 2023). It has been shown that Bayesian acquisition functions such as Bayesian active learning by disagreement (BALD) are connected to CMI (Ma et al., 2019). Perhaps other Bayesian acquisition functions, such as expected predictive information gain (EPIG; Smith et al., 2023), could be adapted for use in dynamic feature acquisition.

For ICU time series data, feature importance has been studied using random forests (Hyland et al., 2020). In Hyland et al. (2020); Yèche et al. (2023) authors showed that deep learning architectures can achieve state-of-the-art performance in early event prediction. Tokenization of observed ICU features has been shown to improve the performance of such models (Kuznetsova et al., 2023). Tokenization of observed features is a natural part of the set-based approaches (Ma et al., 2019), and could be applied in DFA.

6. Conclusion

Dynamic feature acquisition is a challenging problem that arises for temporal data across various applications: medicine, wearable sensors, active perception, etc. It has seen little attention, with previous work considering only reinforcement learning approaches.

In this work, we propose using approximated CMI to dynamically select the most informative features. We show that the acquirer trained using our approach learns to distinguish fake features from real ones for time series classification. Although our model outperformed a random acquisition policy, it did not surpass the static acquisition. This performance gap is likely due to the simplicity of the used architectures.

We hope that this work will be continued, as a wide many questions are still open. Future work may consider more advanced architectures, compare the performance of our training approach to reinforcement learning (Kossen et al., 2023), and loosen the assumptions we adopted: fixed time step budget, fully observed training data, and equal feature acquisition cost.

7. Acknowledgments

FS thanks Shkurta Gashi and Manuel Burger for helpful discussions. Computational data analysis was performed at Leonhard Med secure trusted research environment at ETH Zurich. FS was supported by grant #902 of the Strategic Focus Area “Personalized Health and Related Technologies (PHRT)” of the ETH Domain (Swiss Federal Institutes of Technology). VF was supported by a Branco Weiss Fellowship.

References

Bagnall, A., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., and Keogh, E. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Bajcsy, R., Aloimonos, Y., and Tsotsos, J. K. Revisiting active perception. *Autonomous Robots*, 42:177–196, 2018.

Covert, I. C., Qiu, W., Lu, M., Kim, N. Y., White, N. J., and Lee, S.-I. Learning to maximize mutual information for dynamic feature selection. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 6424–6447. PMLR, 2023. URL https://proceedings.mlr.press/v202/covert23a.html. ISSN: 2640-3498.

Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., and Batista, Gustavo, a. The UCR time series classification archive, October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kossen, J., Cangea, C., Vértes, E., Jaegle, A., Patraucean, V., Ktena, I., Tomasev, N., and Belgrave, D. Active acquisition for multimodal temporal data: A challenging decision-making task, 2023. URL http://arxiv.org/abs/2211.05039.

Kuznetsova, R., Pace, A., Burger, M., Yèche, H., and Rätsch, G. On the importance of step-wise embeddings for heterogeneous clinical time-series, 2023. URL http://arxiv.org/abs/2311.08902. version: 1.

Lewis, S., Matejovicova, T., Li, Y., Lamb, A., Zaykov, Y., Allamanis, M., and Zhang, C. Accurate imputation and efficient data acquisition with transformer-based VAEs. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=N_OwBEYTcKK.

Liu, S., See, K. C., Ngiam, K. Y., Celi, L. A., Sun, X., and Feng, M. Reinforcement learning for clinical decision support in critical care: comprehensive review. *Journal of medical Internet research*, 22(7):e18477, 2020.

Ma, C., Tschiatschek, S., Palla, K., Hernández-Lobato, J. M., Nowozin, S., and Zhang, C. EDDI: Efficient dynamic discovery of high-value information with partial VAE, 2019. URL http://arxiv.org/abs/1809.11142.

Merrill, M. A., Safranchik, E., Kolbeinsson, A., Gade, P., Ramirez, E., Schmidt, L., Foshchini, L., and Althoff, T. Homekit2020: A benchmark for time series classification on a large mobile sensing dataset with laboratory tested ground truth of influenza infections. In *Conference on Health, Inference, and Learning*, pp. 207–228. PMLR, 2023.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Possas, R., Caceres, S. P., and Ramos, F. Egocentric activity recognition on a budget. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5967–5976, 2018.

Sharma, M., Rainforth, T., Teh, Y. W., and Fortuin, V. Incorporating unlabelled data into bayesian neural networks. *arXiv preprint arXiv:2304.01762*, 2023.

Smith, F. B., Kirsch, A., Farquhar, S., Gal, Y., Foster, A., and Rainforth, T. Prediction-oriented bayesian active learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 7331–7348. PMLR, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yang, M., Chen, Y., and Kim, H.-S. Efficient deep visual and inertial odometry with adaptive visual modality selection. In *European Conference on Computer Vision*, pp. 233–250. Springer, 2022.

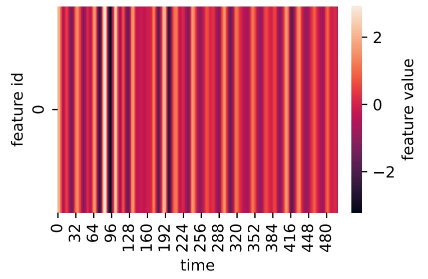
Yèche, H., Pace, A., Ratsch, G., and Kuznetsova, R. Temporal label smoothing for early event prediction. In *International Conference on Machine Learning*, pp. 39913–39938. PMLR, 2023.

Yu, Z., Li, Y., Kim, J., Huang, K., Luo, Y., and Wang, M. Deep reinforcement learning for cost-effective medical diagnosis, 2023. URL http://arxiv.org/abs/2302.10261.

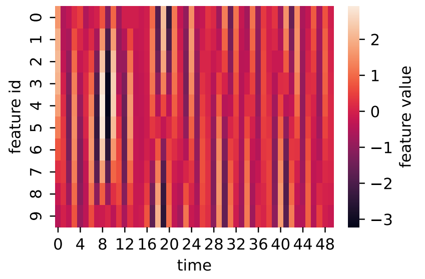
A. Datasets

Dataset	Task	Classses	Domain	Train size	Test size	Number of features	Length	Class balance
FordA	Classification	2	Sensor	3601	1320	1	500	Balanced
m-FordA (m=10)	Classification	2	Sensor	3601	1320	10 (m)	50	Balanced
SpokenArabicDigits	Classification	10	Speach recognition	6600	2200	13	4-93	Balanced

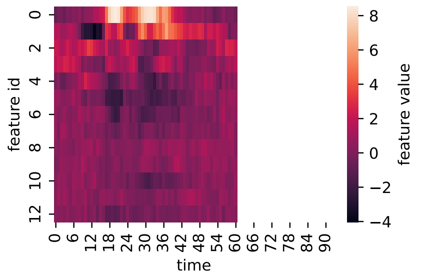
Table A.1. Summary of the datasets.



(a) FordA

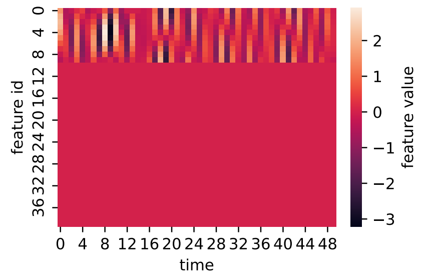


(b) m-FordA

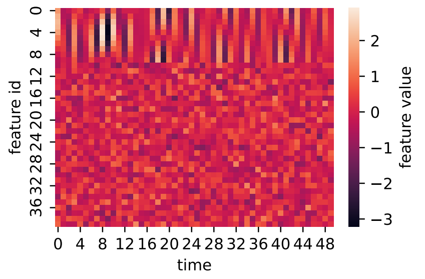


(c) SpokenArabicDigits

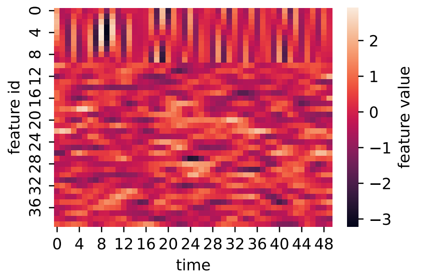
Figure A.1. Samples from the datasets.



(a) Zeros

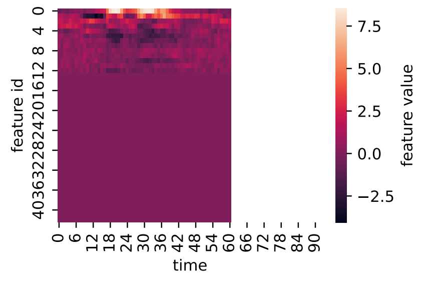


(b) Gaussian noise

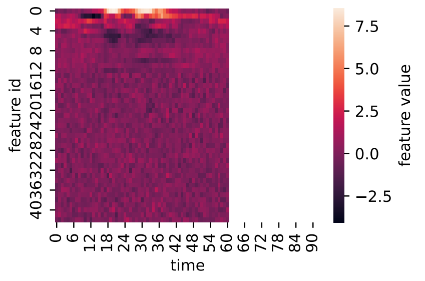


(c) Samples from a GP

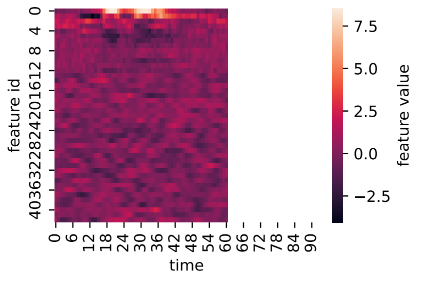
Figure A.2. A sample from the m-FordA dataset with 30 fake features of different kinds.



(a) Zeros



(b) Gaussian noise



(c) Samples from a GP

Figure A.3. A sample from the SpokenArabicDigits dataset with 30 fake features of different kinds.

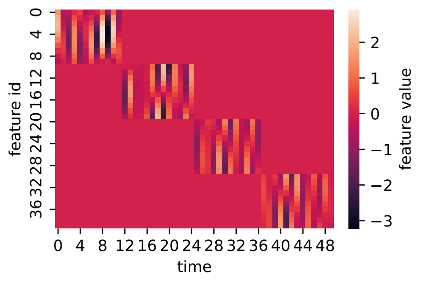


Figure A.4. A sample from the m-FordA dataset with 30 shifted fake features (zeros).

B. Experiments

B.1. Setup details

The classifiers are LSTMs with 16 hidden units, 2 layers for m-FordA and 3 for SpokenArabicDigits, and a linear dimension of 8, followed by one linear layer outputting class logits.

The acquirers are fully connected neural networks with 1 hidden layer. All the inputs are simply concatenated. We use 4 hidden units for m-FordA, and 8 for SpokenArabicDigits.

For logits vector l, the penalty function R is

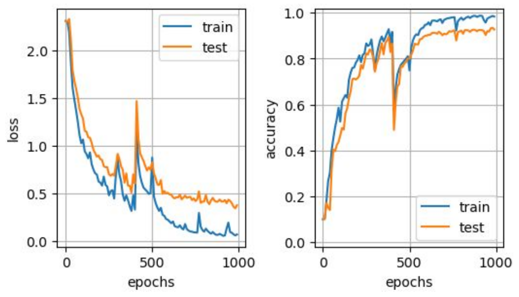
$$R(\boldsymbol{l}) = 100 \cdot \boldsymbol{m}_t \cdot |\boldsymbol{l}|,$$

where the absolute value is taken elementwise.

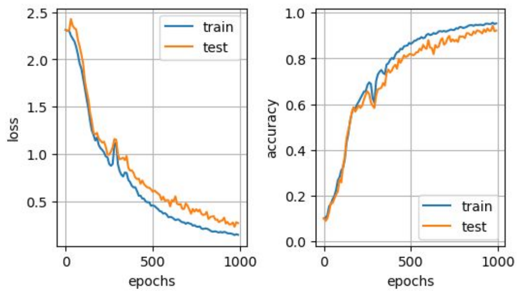
The batch size is 1000, and the learning rate is 0.001 in all experiments.

We use a random forest (static feature selector) with 1000 esimators, leaving the other parameters as defaults provided by scikit-learn (Pedregosa et al., 2011).

B.2. Additional results



(a) Complete acquirer



(b) Our acquirer

Figure B.1. Training curves on m-FordA with 30 fake features (zeros).