Benefit-aware Early Prediction of Health Outcomes on Multivariate EEG Time Series

Shubhranshu Shekhar
Heinz College & Machine Learning Dept., Carnegie Mellon University

Dhivya Eswaran∗
Department of Computer Science, Carnegie Mellon University

Jonathan Elmer
Departments of Emergency Medicine, Critical Care Medicine and Neurology, University of Pittsburgh

Christos Faloutsos
Department of Computer Science, Carnegie Mellon University

Bryan Hooi†
School of Computing, National University of Singapore

Leman Akoglu
Heinz College of Info. Sys., Carnegie Mellon University

ABSTRACT

Given a cardiac-arrest patient being monitored in the ICU (intensive care unit) for brain activity, how can we predict their health outcomes as early as possible? Early decision-making is critical in many applications, e.g. monitoring patients may assist in early intervention and improved care. On the other hand, early prediction on EEG data poses several challenges: (i) earliness-accuracy trade-off: observing more data often increases accuracy but sacrifices earliness, (ii) large-scale (for training) and streaming (online decision-making) data processing, and (iii) multi-variate (due to multiple electrodes) and multi-length (due to varying length of stay of patients) time series. Motivated by this real-world application, we present BeneFitter that infuses the incurred savings from an early prediction as well as the cost from misclassification into a unified domain-specific target called benefit. Unifying these two quantities allows us to directly estimate a single target (i.e. benefit), and importantly, dictates exactly when to output a prediction: when benefit estimate becomes positive. BeneFitter is efficient and fast, with training time linear in the number of input sequences, and can operate in real-time for decision-making, (b) can handle multi-variate and variable-length time-series, suitable for patient data, and (c) is effective, providing up to 2× time-savings with equal or better accuracy as compared to competitors.

1 INTRODUCTION

Early decision making is critical in a variety of application domains. In medicine, earliness in prediction of health outcomes for patients in ICU allows the hospital to redistribute their resources (e.g., ICU bed-time, physician time, etc.) to in-need patients, and potentially achieve better health outcomes overall within the same amount of time. Of course, another critical factor in play is the accuracy of such predictions. Hastily but incorrectly predicting unfavorable health outcome (e.g. withdrawal of life-sustaining therapies) could hinder equitable decision making in the ICU, and may also expose hospitals to very costly lawsuits.

A clinician considers patient history, demographics, etc. in addition to large amounts of real-time sensor information for taking a decision. Our work is motivated by the this real-world application that would help in alleviating the information overload on clinicians and aid them in early and accurate decision making in

∗ Work done while at Carnegie Mellon University
† Now at Amazon Inc.
because patients might not survive or be discharged after varying length of stay at the ICU. BeneFitter addresses these additional challenges such as handling (i) multi-variate and (ii) variable-length signals (i.e., time series), (iii) space-efficient modeling, (iv) scalable training, and (v) constant-time prediction.

We summarize our contributions as follows.

- **Novel, cost-aware problem formulation**: We propose BeneFitter, which infuses the incurred savings/gains \( S(t) \) from an early prediction at time \( t \), as well as the cost \( M \) from each misclassification into a unified target called benefit \( \text{benefit} \). Unifying these two quantities allows us to directly estimate a single target, i.e., benefit, and importantly dictates BeneFitter exactly when to output a prediction: whenever estimated benefit becomes positive.

- **Efficiency and speed**: The training time for BeneFitter is linear in the number of input sequences, and it can operate under a streaming setting to update its decision based on incoming observations. Unlike existing work that train a collection of prediction models for each \( t = 1, 2, \ldots, 24 \), BeneFitter employs a single model for each possible outcome, resulting in much greater space-efficiency.

- **Multi-variate and multi-length time-series**: Due to hundreds of measurements from EEG signals collected from patients with variable length stays at the ICU, BeneFitter employs models that are designed to handle multiple time sequences, of varying length, which is a more general setting.

- **Effectiveness on real-world data**: We apply BeneFitter on real-world (a) multi-variate health care data (our main motivating application for this work is predicting survival/death of cardiac-arrest patients based on their EEG measurements at the ICU), and (b) other 11 benchmark datasets pertaining to various early prediction tasks. On ICU application, BeneFitter can make decisions with up to 2× time-savings as compared to competitors while achieving equal or better performance on accuracy metrics. Similarly, on benchmark datasets, BeneFitter provides the best spectrum for trading-off accuracy and earliness (e.g. see Figure 1).

**Reproducibility**. Our dataset is publicly available at https://bit.ly/3n4wL0N. Our EEG dataset involves real ICU patients and is under NDA with our collaborating healthcare institution.

**Suitability for Real-world Use Cases**

- **Business problem**: Customizable, cost-aware formulation. The benefit allows domain experts to explicitly control for \( S(t) \) and \( M \), thereby enabling them to incorporate into the predictions the knowledge gathered through experience or driven by the application domain.

- **Usability**: Early prediction of health outcomes to aid physicians in their decisions.

## 2 DATA AND PROBLEM SETTING

### 2.1 Data Description

Our use case data are obtained from 725 comatose patients who are resuscitated from cardiac arrest and underwent post-arrest electroencephalography (EEG) monitoring at a single academic medical center between years 2010–2018.

The raw EEG data are recorded at 256 Hz from 22 scalp electrodes; 11 electrodes in each hemisphere of the brain placed according to 10–20 International System of Electrode Placement. The raw data is then used to collect quantitative EEG (qEEG) features at an interval of ten seconds that amounts to about 900 GB of disk space for 72 patients. For our experiments, we selected 107 qEEG signals that physicians find informative from the electrode measurements corresponding to different brain regions. The 107-dimensional qEEG measurements from different electrodes on both left and right hemisphere, including the amplitude-integrated EEG (aEEG), burst suppression ratio (SR), asymmetry, and rhythmicity, form our multivariate time-series for analysis. We also record qEEG for each hemisphere as average of qEEG features from 11 electrodes on the given hemisphere.

As part of preprocessing, we normalize the qEEG features in a range [0, 1]. The EEG data contains artifacts caused due to variety of informative (e.g. the patient wakes up) or arbitrary (e.g. device unplugged/unavailability of devices) reason. This results in missing values, abnormally high or zero measurements. We filter out the zero measurements, typically, appearing towards the end of each sequence as well as abnormally high signal values at the beginning of each time series from the patient records. The zero measurements towards the end appear because of the disconnection. Similarly, abnormally high readings at the start appear when a patient is being plugged for measurements. The missing values are imputed through linear interpolation.

In this dataset, 225 patients (≈ 31%) out of total 725 patients survived i.e. woke up from coma. Since the length of stay in ICU depends on each individual patient, the dataset contains EEG records of length 24–96 hours. To extensively evaluate our proposed approach, we create 3 versions of the dataset by median sampling [9] the sequences at one hour, 30 minutes and 10 minutes intervals (as summarized in §5, Table 4).

### 2.2 Notation

A multi-variate time-series dataset is denoted as \( X = \{(X_i, l_i)\}_{i=1}^n \), consisting of observations and labels for \( n \) instances. Each instance \( i \) has a label \( l_i \in \{1, \ldots, C\} \) where \( C \) is the number of labels or classes. For example, each possible health outcome at the ICU is depicted by a class label as survival or death. The sequence of observations is given by \( X_i = \{X_{i1}, \ldots, X_{iL_i}\} \) for \( L_i \) equi-distant time ticks. Here, \( L_i \) is the length of time-series \( i \) and varies from one instance to another in the general case. It is noteworthy that our proposed BeneFitter can effectively handle variable-length series in a dataset, whereas most existing early prediction techniques are limited to fixed length time-series, where \( L_i = L \) for all \( i \in \{1, \ldots, n\} \). Each observation \( X_{i1} \in \mathbb{R}^d \) is a vector of \( d \) real-valued measurements, where \( d \) is the number of variables or signals. We denote \( X_i \)’s observations from the start until time tick \( t \) by \( X_{i[1:t]} \).

### 2.3 Problem Statement

Early classification of time series seeks to generate a prediction for input sequence \( X \) based on \( X_{[1:t]} \) such that \( t \) is small and \( X_{[1:t]} \) contains enough information for an accurate prediction. Formally,
Problem 1 (Early classification). Given a set of labeled multivariate time series $X = \{(X_i, l_i)\}_{i=1}^n$, learn a function $F_{\theta}(\cdot)$ which assigns label $l_i$ to a given time series $X_{[1:t]}$ i.e. $F_{\theta}(X_{[1:t]}) \mapsto l_i$ such that $t$ is small.

Challenges The challenges in early classification are two-fold: domain-specific and task-specific, discussed as follows.

- **Domain-specific** Data preprocessing is non-trivial since raw EEG data includes various biological and environmental artifacts. Observations arrive incrementally across multiple signals where the characteristics that are indicative of class labels may occur at different times across signals which makes it difficult to find a decision time to output a label. Moreover, each time series instance can be of different length due to varying length of stay of patients at the ICU which requires careful handling.

- **Task-specific** Accuracy and earliness of prediction are competing objectives (as noted above) since observing for a longer time, while cuts back from earliness, provides more signals that is likely to yield better predictive performance.

In this work, we propose BeneFitter (see §4) that addresses all the aforementioned challenges.

3 BACKGROUND AND RELATED WORK

The initial mention of early classification of time-series dates back to early 2000s [2, 16] where the authors consider the value in classifying prefixes of time sequences. However, it was formulated as a concrete learning problem only recently [21, 22]. Xing et al. [21] mine a set of sequential classification rules and formulate an early-prediction utility measure to select the features and rules to be used in early classification. Later they extend their work to a nearest-neighbor based time-series classifier approach to wait until a certain level of confidence is reached before outputting a decision [22]. Parrish et al. [15] delay the decision until a reliability measure indicates that the decision based on the prefix of time-series is likely to match that based on the whole time-series. Xing et al. [23] advocate the use of interpretable features called shapelets [25] which have a high discriminatory power as well as occur earlier in time cuts back from earliness, provides more signals that is likely to yield better predictive performance.

Here, the labels of time-series encode characteristics that are worth distinguishing between. Therefore, Hartvigsen et al. [6] employ recurrent neural network (RNN) based discriminator for classification paired with a reinforcement learning task to learn halting policy. The closest in spirit to our work is the recently proposed end-to-end learning framework for early classification [17] that employs RNNS. They use a cost function similar to [13] in a fine-tuning framework to learn a classifier and a stopping rule based on RNN embeddings for partial sequences.

Our proposed BeneFitter is a substantial improvement over all the above prior work on early classification of time series along a number of fronts, as summarized in Table 1. BeneFitter jointly optimizes for earliness and accuracy using a cost-aware benefit function. It seamlessly handles multi-variate and varying-length time-series and moreover, leads to explainable early predictions, which is important in high-stakes domains like health care.

4 BENEFITTER: PROPOSED METHOD

4.1 Modeling Benefit

How should an early prediction system trade-off accuracy vs. earliness? In many real-world settings, there is natural misclassification cost, denoted $M$, associated with an inaccurate prediction and certain savings, denoted $S(t)$, obtained from early decision-making. We propose to construct a single variable called $benefit$ which captures the overall value (savings minus cost) of outputting a certain decision (i.e., label) at a certain time $t$, given as

$$benefit = S(t) - M$$

We directly incorporate benefit into our model and leverage it in deciding when to output a decision; when the estimate is positive.

4.1.1 Outcome vs. Type Classification. There are two subtly different problem settings that arise in time-series classification that are worth distinguishing between.

- **Outcome classification**: Here, the labels of time-series encode the observed outcome at the end of the monitoring period of each instance. Our motivating examples from predictive health care and system maintenance fall into this category. Typically, there are two outcomes: favorable (e.g., survival or no-failure) and unfavorable (e.g., death or catastrophic-failure); and we are interested in

| Property / Method | ECTS [22] | C-ETS [14, 19] | EDSC [26] | M-EDSC [5] | Relax [15] | Ezel [17] | BeneFitter |
|-------------------|-----------|----------------|----------|-----------|-----------|-----------|------------|
| Jointly optimize earliness & accuracy | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Distance metric agnostic | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Multivariate | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Constant decision time | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Handles variable length series | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Explainable model | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Explainable hyper-parameter | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cost aware | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
knowing when an unfavorable outcome is anticipated. In such cases, predicting an early favorable outcome does not incur any change in course of action, and hence does not lead to any discernible savings or costs. For example, in our ICU application, a model predicting survival (as opposed to death) simply suggests to the physicians that the patient would survive provided they continue with their regular procedures of treatment. That is because \( l = \text{survival} \) labels we observe in the data are at the end of the observed period only after all regular course of action have been conducted. In contrast, \( l = \text{death} \) instances have died despite the treatments.

In outcome classification, predicting the favorable class simply corresponds to the ‘default state’ and therefore we model benefit \( b \) and actively make predictions only for the unfavorable class.

- **Type classification:** Here, the time-series labels capture the underlying process that gives rise to the sequence of observations. In other words, the class labels are prior to the time-series observations. The standard time-series early classification benchmark datasets fall into this category. Examples include predicting the type of a bird from audio recordings or the type of a flying insect (e.g., a mosquito) from their wingbeat frequencies [1]. Here, prediction of any label for a time-series at a given time has an associated cost in case of misclassification (e.g., inaccurate density estimates of birds/mosquitoes) as well as potential savings for earliness (e.g., battery life of sensors). In type classification, we separately model benefit \( b \) for each class.

### 4.1.2 Benefit Modeling for Outcome Classification

Consider the 2-class problem that arises in predictive health care of ICU patients and predictive maintenance of systems. Without loss of generality, let us denote by \( l = 0 \) the survival class where the patient is discharged alive from the ICU at the end of their stay; and let \( l = 1 \) denote the death class where the patient is deceased.

As discussed previously, \( l = 0 \) corresponds to the ‘default state’ in which regular operations are continued. Therefore, predicting survival would not incur any time savings or misclassification cost. In contrast, predicting death would suggest the clinician to intervene to optimize quality of life for the patient. In case of an accurate prediction, say at time \( t \), earliness would bring savings (e.g., ICU bed-time), denoted \( S(t) \). Here we use a linear function of time for savings on accurately predicting death for a patient \( i \) at time \( t \), specifically

\[
S(t) = (L_i - t)s
\]

where \( s \) denotes the value of savings per unit time.\(^3\) On the other hand, an inaccurate death flag at \( t \), while comes with the same savings, would also incur a misclassification cost \( M \) (e.g., a lawsuit).

All in all, the benefit model for the ICU scenario is given as in Table 2, reflecting the relative savings minus the misclassification cost for each decision at time \( t \) on time-series instance \( i \). As we will detail later in §4.3, the main idea behind BeneFitter is to learn a single regressor model for the death class, estimating the corresponding benefit \( b \) at each time tick \( t \).

**Specifying \( s \) and \( M \).** Here, we make the following two remarks. First, unit-time savings \( s \) and misclassification cost \( M \) are value estimates that are dictated by the specific application. For our ICU case, for example, we could use \( s = \$4,000 \) value per unit ICU time, and \( M = \$500,000 \) expected cost per lawsuit. Note that \( s \) and \( M \) are domain-specific explainable quantities. Second, the benefit model is most likely to differ from application to application. For example in predictive system maintenance, savings and cost would have different semantics, assuming that early prediction of failure implies a renewal of all equipment. In that case, an early and accurate failure prediction would incur savings from costs of a complete system halt, but also long of equipment lifetime value due to early replacement plus the replacement costs. On the other hand, early but inaccurate failure prediction (i.e., a false alarm) would simply incur unnecessary replacement costs plus the loss of equipment lifetime value due to early replacement.

Our goal is to set up a general prediction framework that explicitly models benefit \( b \) based on incurred savings and costs associated with individual decisions, whereas the scope of specifying those savings and costs are left to the practitioner. We argue that each real-world task should strive to explicitly model benefit, where earliness and accuracy of predictions translate to real-world value. In cases where the prediction task is isolated from its real-world use (e.g., benchmark datasets), one could set both \( s = M = 1 \) for unit savings per unit time earliness and unit misclassification cost per incorrect decision. In those cases where \( M \) is not tied to a specific real-world value, it can be used as a “knob” (i.e., hyperparameter) for trading off accuracy with earliness; where, fixing \( s = 1 \), a larger \( M \) nudges BeneFitter to avoid misclassifications toward higher accuracy at the expense of delayed predictions and vice versa.

### 4.1.3 Benefit Modeling for Type Classification

Compared to outcome prediction where observations give rise to the labels, in type classification problems the labels give rise to the observations. Without a default class, predictions come with associated savings and cost for each class.

### Table 2: Benefit model for ICU outcome prediction.

| Actual \( l_i \) | Predicted \( l_i \) | \( S(t) = (L_i - t)s - M \) |
|-----------------|-----------------|------------------|
| survival        | 0               | \( (L_i - t)s - M \) |
| death           | 0               | \( (L_i - t)s \) |

Consider the 2-class setting of predicting an insect’s type from wingbeat frequencies. An example benefit model is illustrated in Table 3, \( s \) capturing the value of battery-life savings per unit time and \( M \) depicting the cost of misclassifying one insect as the other. Note that in general, misclassification cost need not be symmetric among the classes.

For type classification problems, we train a total of \( C \) benefit models, one for each class. Since misclassification costs are already incorporated into benefit, we train each (regression) model independently which allows for full parallelism.

\(^3\)Note that BeneFitter is flexible enough to accommodate any other function of time, including nonlinear ones, as the savings function \( S(t) \).
4.2 Online Decision-making using Benefit

Next we present how to employ BeneFitter in decision making in real time. Suppose we have trained our model that produces benefit estimates per class for a new time-series instance in an online fashion. How and when should we output predictions?

Thanks to our benefit modeling, the decision-making is quite natural and intuitive: BeneFitter makes a prediction only when the estimated benefit becomes positive for a certain class and outputs the label of that class as its prediction i.e. for our ICU scenario the predicted label \( \hat{l} \) is given as

\[
\hat{l} = \begin{cases} 
\text{unfavorable}, & \text{if } \text{benefit} > 0 \\
\text{favorable}, \text{ i.e. no action}, & \text{otherwise.}
\end{cases}
\]

For illustration, in Fig. 2 we show benefit estimates over time for an input series where \( t = 15 \) corresponds to decision time.

Note that in some cases BeneFitter may refrain from making any prediction for the entire duration \( L \) of a test instance, that is when estimated benefit never goes above zero. For outcome classification tasks, such a case is simply registered as default-class prediction and its prediction time is recorded as \( L \). For the ICU scenario, a non-prediction is where no death flag is raised, suggesting survival and regular course of action. For type classification scenario, a non-prediction is where no \( \text{de} \) \( \text{at} \) \( \text{h} \) \( \text{e} \) \( \text{a} \) \( \text{t} \) flag is raised, suggesting favorable, i.e. no action.

\[ \text{For } l \text{ instances may be of independent interest to domain experts in the } \]

\[ \text{natural and intuitive: How } \]

\[ \text{should we output predictions?} \]

\[ \text{Attention. The recurrent networks usually find it hard to focus on relevant information in long input sequences. For example, an EEG pattern in the beginning of a sequence may contain useful information about the patient’s outcome, however the lossy representation of LSTM would forget it. This issue is mostly encountered in longer input sequences [12]. The underlying idea of attention [20] is to learn a context that captures the relevant information from the parts of the sequence to help predict the target. For a sequence of length } L \text{, given the hidden state } h_L \text{ from LSTM and the context vector } c \text{, the attention step in BeneFitter combines the information from both vectors to produce a final attention based hidden state as described below:} \]

\[
\alpha_t = \frac{\exp(c_L \cdot h_t)}{\sum_t \exp(c_L \cdot h_t)}; \quad c = \sum_t \alpha_t h_t \quad (3)
\]

\[
h_{\text{attn}} = \sigma(W_d[\text{concat}(c, c_L)]) \quad (4)
\]

where \( c_L \) is the memory state of the cell at the last time step \( L \), \( h_t \) is the hidden state output of the LSTM at time \( t \), \( h_{\text{attn}} \) is the attention based hidden state, \( \sigma(\cdot) \) is the non-linear transformation, and \( W_d \) is the parameter. Intuitively, the attention weights \( \alpha_t \) allows the model to learn to focus on specific parts of the input sequence for the task of regression. The benefit prediction is given by a single layer neural network such that \( \hat{b}_L = h_{\text{attn}}w + w_0 \) where \( w \) and \( w_0 \) are parameters of the linear layer.

BeneFitter is used in real life decision making, where the attention mechanism could help an expert by highlighting the relevant information that guided the model to output a decision. We present model implementation details and list of tunable parameters in §5.

5 EXPERIMENTS

We evaluate our method through extensive experiments on a set of benchmark datasets and on a set of datasets from real-world use cases. We next provide the details of the datasets and the experimental setup, followed by results.

5.1 Dataset Description

We apply BeneFitter on our EEG-ICU datasets (see §2.1), and on 11 public benchmark datasets from diverse domains with varying dimensionality, length and scale. Table 4 provides a summary of the datasets used in evaluation. Note that EEG-ICU datasets are variable-length, but benchmarks often used in the literature are not. Detailed description of public datasets are included in Appx. A.1.

5.2 Experimental Setup

Baselines. We compare BeneFitter to the following five early time-series classification methods (also see Table 1):

\[ \text{Figure 2: Benefit estimate over time for a patient from EEG dataset with true } l = 1 \text{ (i.e. death). We show two out of all 107 signals used by BeneFitter: amplitude of EEG (aEEG) and suppression ratio (i.e. fraction of flat EEG epochs).} \]
We design our experiments to answer the following questions:

1. **Effectiveness**: How effective is BeneFitter at early prediction on time series compared to the baselines? What is the trade-off with respect to accuracy and earliness? How does the accuracy-earliness trade-off vary with respect to model parameters?

2. **Efficiency**: How does the running-time of BeneFitter scale w.r.t. the number of training instances? How fast is the online response time of BeneFitter?

3. **Discoveries**: Does BeneFitter lead to interesting discoveries on real-world case studies?

4. **Patient Outcome Prediction**: We compare BeneFitter with the baseline E2EL on two competing criteria: performance (e.g., precision, accuracy) and earliness (tardiness – lower is better) of the decision. We report precision, recall, F1 score, accuracy, tardiness and the total benefit using each method when applied to the test set. EEG dataset is a high dimensional variable-length dataset for which most of the baselines are not applicable. In our experiments, we set $M/s = (100, 200, 600)$ as misclassification cost for each of the dataset variants – sampled at an hour, 30 minutes, 10 minutes – respectively based on average daily cost of ICU care and the lawsuit cost. For the baseline methods, we report the best results for the earliness-accuracy trade-off parameters. For the baseline methods, we select the best value of accuracy and earliness based on their Euclidean distance to ideal accuracy = 1 and ideal tardiness = 0.

Table 5 reports the evaluation against different performance metrics. Note that predicting ‘default state’ for a patient does not change in such a decision setting, it is critical for the classifier to exhibit high precision. Our results indicate that BeneFitter achieves a significantly higher precision (according to the micro-sign test [24]) when compared to the baselines. On the other hand, a comparatively lower tardiness indicates that BeneFitter requires considerably fewer observations on average to output a decision (no statistical test conducted for tardiness). We also compare the total benefit accrued for each method on the test set where BeneFitter outperforms the competition. The results are consistent across the three datasets of varying granularity from hourly sampled data to 10 minute sampled data.

| Dataset          | Train | Test | Classes | Length | Dimension |
|------------------|-------|------|---------|--------|-----------|
| EEG-ICU Hour     | 507   | 218  | 2       | 24–96  | 107       |
| EEG-ICU 30 Min   | 507   | 218  | 2       | 48–192 | 107       |
| EEG-ICU 10 Min   | 507   | 218  | 2       | 144–376| 107       |
| ECG200           | 100   | 100  | 2       | 96     | 1         |
| ItalyPowerDemand | 67    | 1029 | 2       | 24     |           |
| GunPoint         | 50    | 150  | 2       | 150    | 1         |
| TwoLeadECG       | 23    | 1139 | 2       | 82     | 1         |
| Wafer            | 1000  | 6062 | 2       | 152    | 1         |
| ECGFiveDays      | 23    | 861  | 2       | 136    | 1         |
| MotelStrain      | 20    | 1252 | 2       | 84     | 1         |
| Coffee           | 28    | 28   | 2       | 286    | 1         |
| Yoga             | 300   | 3000 | 2       | 426    | 1         |
| SonyAIBO         | 20    | 601  | 2       | 70     | 1         |
| Endomondo        | 99754 | 42751| 2       | 450    | 2         |

Table 4: Summary of the datasets used in this work.

Table 5: Effectiveness of BeneFitter on EEG datasets. * indicates significance at $p$-value $\leq 0.05$ based on the micro-sign test [24] for the performance metrics. No statistical test conducted for tardiness and total benefit.

| Dataset | Precision | Recall | F1 | Accuracy | Tardiness | Benefit |
|---------|-----------|--------|----|----------|-----------|---------|
| E2EL    | 0.70      | 0.68   | 0.69| 0.79     | 1.0       | -2600   |
| EEG Hour| M-EDSC    | 0.69   | 0.65| 0.67     | 0.78      | 0.52    | 2497    |
|         | BeneFitter| 0.80*  | 0.68| 0.73*    | 0.83*     | 0.64    | 2737    |
| EEG 30Min| E2EL     | 0.64   | 0.67| 0.65     | 0.78      | 1.0     | -4800   |
|         | BeneFitter| 0.68*  | 0.66| 0.67*    | 0.79*     | 0.63    | 5962    |
| EEG 10Min| E2EL     | 0.73   | 0.69| 0.71     | 0.82      | 0.86    | -736    |
|         | BeneFitter| 0.76*  | 0.69| 0.72     | 0.83*     | 0.48    | 18722   |

For hourly sampled set, we also compare our method to multivariate EDSC baseline (for the 30 min and 10 min EEG dataset M-EDSC does not scale). Though M-EDSC provides better earliness trade-off compared to other two methods, the precision of the
outcomes is lowest which is not desirable in this decision setup. In Table 5, we indicate the significant results using \( ^* \) that is based on the comparison between BeneFitter and E2EL.

**Benchmark Prediction Tasks.** To jointly evaluate the accuracy and earliness (tardiness – lower is better), we plot accuracy against the tardiness to compare the Pareto frontier for each of the competing methods over 10 different benchmark datasets. In Fig. 1 and Fig. 3, we show the accuracy and tardiness trade-off for 10 benchmark UCR datasets. Each point on the plot represents the model evaluation for a choice of trade-off parameters reported in Table A1 (§A.2). Note that BeneFitter dominates the frontiers of all the baselines in accuracy vs tardiness on five of the datasets. Moreover, our method appears on the Pareto frontier for four out of the remaining five for at least one set of parameters.

To further assess the methods, we report quantitative results in Table 6 in terms of accuracy at a given tolerance of tardiness. We indicate the significant results using \( ^* \) that is based on the comparison between BeneFitter and E2EL (other baselines do not scale) for one set of parameters. Notice that the two methods are comparable in terms of the prediction performance while using less than half the length of a sequence for outputting a decision. The quantitative results suggest a way to choose the best classifier for a specified tolerance level for an application. In critical domains such as medical, or predictive maintenance a lower tolerance would be preferred to save cost. In such domains, BeneFitter provides a clear choice for early decision making based on the benchmark dataset evaluation.

**[Q2] Efficiency.** Fig. 4 shows the scalability of BeneFitter with the number of training time-series and number of observations per time series at test time. We use ECG200 dataset from UCR benchmark to report results on runtime.

**Linear training time:** We create ten datasets from the ECG200 dataset by retaining a fraction \( \in \{0.1, 0.2, \ldots, 1.0\} \) of total number of training instances. For a fixed set of parameters, we train our model individually for each of the created datasets. The wall-clock running time is reported against the fraction of training sequences in Fig. 4 (left). The points in the plot align in a straight line indicating that BeneFitter scales linearly with number of sequences.

**Constant test time:** We now evaluate BeneFitter runtime by varying the number of observations over time. For this experiment, we retain the hidden state of an input test sequence up to time \( t \) (the position in the plot align in a straight line indicating that BeneFitter scales linearly with number of sequences. We run the experiments on full Endomondo dataset (a large scale dataset) to compare BeneFitter with baseline E2EL (other baselines do not scale) for one set of earliness-accuracy trade-off parameters. We, first, compare the two methods on a sampled dataset – with 1000 time series instances – evaluated for a choice of trade-off parameters (see Fig. 6 in Appendix A.3). We select the parameters that yields a performance closest to ideal indicated. With the selected parameters, comparison of two methods on large-scale Endomondo activity prediction dataset are reported in Table 6 (last row). We report the accuracy of the two methods by fixing their tardiness at \( \leq 0.5 \). Notice that the two methods are comparable in terms of the prediction performance while using less than half the length of a sequence for outputting a decision.

The quantitative results suggest a way to choose the best classifier for a specified tolerance level for an application. In critical domains such as medical, or predictive maintenance a lower tolerance would be preferred to save cost. In such domains, BeneFitter provides a clear choice for early decision making based on the benchmark dataset evaluation.

**Endomondo Activity Prediction**. We run the experiments on full Endomondo dataset (a large scale dataset) to compare BeneFitter with baseline E2EL (other baselines do not scale) for one set of earliness-accuracy trade-off parameters. We, first, compare the two methods on a sampled dataset – with 1000 time series instances – evaluated for a choice of trade-off parameters (see Fig. 6 in Appendix A.3). We select the parameters that yields a performance closest to ideal indicated. With the selected parameters, comparison of two methods on large-scale Endomondo activity prediction dataset are reported in Table 6 (last row). We report the accuracy of the two methods by fixing their tardiness at \( \leq 0.5 \). Notice that the two methods are comparable in terms of the prediction performance while using less than half the length of a sequence for outputting a decision.

The quantitative results suggest a way to choose the best classifier for a specified tolerance level for an application. In critical domains such as medical, or predictive maintenance a lower tolerance would be preferred to save cost. In such domains, BeneFitter provides a clear choice for early decision making based on the benchmark dataset evaluation.

**Table 6:** BeneFitter wins most of the times. Accuracy on benchmark datasets against mean tardiness \( \in \{0.5, 0.75\} \). Bold represents best accuracy within a given tardiness tolerance, and the underline represents the next best accuracy. \( ^* \) indicates that on average method requires more observations than the given tardiness tolerance. \( ^\prime \) specifies non-applicability of a method on the dataset, and \( ^\prime\prime \) shows that a method does not scale for the dataset.

| Dataset       | Tardiness (s) | ECTS | EDNS | C-ECTS | RefClass | E2EL | BeneFitter |
|---------------|---------------|------|------|--------|----------|------|------------|
| ECGFiveDays   | 0.50          | 0.84 | 0.83 | 0.88   | 0.87     | 0.91 |            |
|               | 0.75          | 0.84 | 0.83 | 0.89   | 0.87     | 0.91 |            |
| TwoLeadECG    | 0.50          | 0.88 | 0.89 | -      | 0.79     | 0.98 |            |
|               | 0.75          | 0.89 | 0.84 | 0.91   | 0.96     | 0.97 |            |
| MoteStrain    | 0.50          | 0.90 | 0.96 | 1.0    | 0.99     | 0.99 |            |
|               | 0.75          | 1.0  | 0.96 | 1.0    | 0.99     | 0.99 |            |
| Coffee        | 0.50          | 0.72 | 0.95 | 0.57   | 0.64     | 0.87 |            |
|               | 0.75          | 0.72 | 0.95 | 0.59   | 0.77     | 0.87 |            |
| Yota          | 0.50          | 0.71 | 0.85 | -      | -        | 0.85 |            |
|               | 0.75          | 0.75 | 0.85 | -      | -        | 0.85 |            |
| SenseAIO      | 0.50          | 0.80 | 0.98 | 0.89   | 0.53     | 0.93 |            |
|               | 0.75          | 0.80 | 0.98 | 0.89   | 0.53     | 0.93 |            |
| Endomondo     | 0.50          | 0.50 | 0.50 | 0.68   | 0.66     |      |            |
[Q3] Discoveries. In this section, we present an analysis of BeneFitter highlighting some of the salient aspects of our proposed framework on ICU patient outcome task. In particular, we discuss how our method explains the benefit estimation corresponding to each hidden state \( \tau \), where \( \tau = t \). The model outputs the attention weights corresponding to each time-dimension of the inputs shown in Fig. 5 as a heatmap (dark colors indicate lower weights, lighter colors indicate higher weights). BeneFitter outputs a decision at \( t = 4 \), however we evaluate the model at further time steps. Note that the each row of heat map represents evaluation of input at \( t = 0, \ldots, 23 \). For each evaluation, we obtain a weight signifying the importance of a time dimension which are plotted as heat map. X-axis of heat map corresponds to time dimension, and y-axis of the heat map corresponds to the evaluation time step of the input. Observe that the attention places higher weights towards the beginning of the time series where we observe the crests and troughs of the signal. The visualization of importance weights is useful in critical applications where each decision involves a high cost. The estimated benefit along with the importance information can assist a clinician with the decision making.

Assisting with Model Evaluation In the clinical setting with comatose patients, there is a natural cost associated with an inaccurate
In this paper, we consider the benefit-aware early prediction of health outcomes on ICU-EEG data where BeneFitter provides up to 2× time-savings as compared to competitors while achieving equal or better accuracy. BeneFitter also outperformed or tied with top competitors on other real-world benchmarks.

### 6 CONCLUSIONS

In this paper, we consider the benefit-aware early prediction of health outcomes for ICU patients and proposed BeneFitter that is designed to effectively handle multi-variate and variable-length signals such as EEG recordings. We made multiple contributions.

1. **Novel, cost-aware problem formulation**: BeneFitter infuses the incurred savings from an early prediction as well as the cost from misclassification into a unified target called benefit. Unifying these two quantities allows us to directly estimate a single target, i.e., benefit, and importantly dictates BeneFitter exactly when to output a prediction: whenever estimated benefit becomes positive.

2. **Efficiency and speed**: The training time for BeneFitter is linear in the number of input sequences, and it can operate under a streaming setting to update its decision based on incoming observations.

### Table 7: Training and evaluation of BeneFitter for $M/s = \{100, 200, 300, 400\}$ on EEG hourly sampled data.

| $M/s$ | Precision | Recall | F1 score | Accuracy | Tardiness | Benefit |
|-------|-----------|--------|----------|----------|-----------|---------|
| 100   | 0.80      | 0.68   | 0.73     | 0.83     | 0.64      | 2737    |
| 200   | 0.80      | 0.67   | 0.71     | 0.82     | 0.68      | 1032    |
| 300   | 0.82      | 0.67   | 0.71     | 0.84     | 0.68      | 156     |
| 400   | 0.82      | 0.69   | 0.75     | 0.84     | 0.70      | -1326   |

### References

[1] Gustavo E. Batista, Eamonn J. Keogh, Agenor Mafra-Neto, and Edgar Rowton. 2011. SIGKDD Demo: Sensors and Software to Allow Computational Entomology, an Emerging Application of Data Mining. In SIGKDD. ACM, New York, NY, USA.

[2] Aníbal Bregón, M Aránzazu Simón, Juan José Rodríguez, Carlos Alonso, Belarmino Pulido, and Isaac Moro. 2005. Early fault classification in dynamic systems using case-based reasoning. In CAEPIA. Springer, 211–220.

[3] Yaping Chen, Eamonn Keogh, Bing Hu, Nurjanah Begum, Anthony Bagnall, Abdullah Muree, and Gustavo Batista. 2015. The UCR Time Series Classification Archive. www.cs.ucr.edu/~eamonn/time_series_data/.

[4] Asma Dachroui, Alexis Bondu, and Antoine Cornuéjols. 2015. Early classification of time series as a non myopic sequential decision making problem. In ECML/PKDD. Springer, 433–447.

[5] Mohamed F Ghalwash and Zoran Obradovic. 2012. Early classification of multivariate temporal observations by extraction of interpretable shapelets. BMC bioinformatics 13, 1 (2012), 195.

[6] Thomas Hartvigsen, Canou Sen, Xiangnan Kong, and Elke Rundensteiner. 2019. Adaptive-halting policy network for early classification. In SIGKDD. 101–110.

[7] Nima Hatami and Camelia Chira. 2013. Classifiers with a reject option for early time-series classification. In CIEE, IEEE, 9–16.

[8] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.

[9] BI Justusson. 1981. Median filtering: Statistical properties. In Two-Dimensional Digital Signal Processing II. Springer, 161–196.

[10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[11] Alexander VLotov, Vladimir A Bushenkov, and Georgy K Kamenev. 2013. Interactive decision maps: Approximation and visualization of Pareto frontier. Vol. 89. Springer Science & Business Media.

[12] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015).

[13] Juan J Rodríguez, Carlos Alonso, and Henrik Boström. 2001. Boosting interval classification of time series as a non myopic sequential decision making problem. In SIGKDD. ACM, 1343–1353.

[14] Jianmo Ni, Larry Muhlstein, and Julian McAuley. 2019. Modeling Heart Rate and Activity Data for Personalized Fitness Recommendation. In SIGIR. 31, 1 (2012), 105–127.

[15] Romain Tavenard and Simon Malinowski. 2019. End-to-end Learning for Early Classification of Time Series. arXiv preprint arXiv:1901.10681 (2019).

[16] Patrick Schäfer and Ulf Leser. 2020. TEASER: early and accurate time series classification. Data mining and knowledge discovery (2020), 1336–1362.

[17] Romain Tavenard and Simon Malinowski. 2016. Cost-aware early classification of time series. In ECML/PKDD. Springer, 633–642.

[18] Asish Varwani, Noam Shazeer, Nikó Palmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NeurIPS. 5998–6008.

[19] Zhengzheng Xing, Jian Pei, Guozhu Dong, and Philip S Yu. 2008. Mining sequence classification based literals. Intelligent Data Analysis 5, 3 (2001), 245–262.

[20] Marc Ruvuirm, Sébastien Lefèvre, Nicolas Courtay, Rémi Emonet, Marco Kirner, and Romain Tavenard. 2019. Early classification on time series. In SIGIR. 31, 1 (2012), 105–127.

[21] Zhengzheng Xing, Jian Pei, Guozhu Dong, and Philip S Yu. 2008. Mining sequence classification based literals. Intelligent Data Analysis 5, 3 (2001), 245–262.

[22] Zhengzheng Xing, Jian Pei, and S Yu Philip. 2012. Early classification on time series. Knowledge and information systems 31, 1 (2012). 105–127.

[23] Zhengzheng Xing, Jian Pei, Philip S Yu, and Ke Wang. 2011. Extracting interpretable features for early classification on time series. In SIAM SDM. 247–258.

[24] Yuning Yang, Xin Liu, et al. 1999. A re-examination of test categorization methods. In SIGIR.

[25] Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In SIGKDD. ACM, 947–956.
A APPENDIX

In this section, we first give a detailed description of the datasets used in the experiments. We then provide model training details for BeneFitter, and present the hyper-parameter configurations used in BeneFitter as well as the competing methods for the benchmark experiments.

A.1 Dataset Description

- **Benchmark Datasets.** Our benchmark datasets consist of 10 two-class time-series classification datasets from the UCR repository [3]. The datasets cover diverse domains and have diverse range of series length. The UCR archive provides the train/test split for each of these datasets, which we retain in our experiments.

- **Endomondo Dataset.** Endomondo is a social fitness app that tracks numerous fitness attributes of the users. We use the web-scale Endomondo dataset [14] (See Table 4) for the early activity prediction task. The data includes various measurements such as heart rate, altitude and speed, along with contextual data such as user id and activity type. For the task of early activity prediction, we use heart rate and altitude signals for early prediction of the type of activity, specifically biking vs. running. (Note that we leave out signals like speed and its derivatives which make the classification task too easy.)

A.2 Experimental Settings

A.2.1 Model Training Details. We define the outcome prediction problem as a regression task on the benefit, as presented in §4. The training examples represent the sequences observed up to time \( t \) along with their corresponding expected benefit at time \( t \). We then split the training examples to use 90% of the sequences for training the RNN model and remaining 10% for validation. We select our model parameters based on the evaluation on validation set. We have two sets of hyper-parameters: one corresponding to our benefit formulation that are \( s \) and \( M \), and the other for the RNN model. The hyper-parameter grid for the RNN model includes the dimension of the hidden representation \( \in \{16, 32\} \) and the learning rate \( \eta \in \{0.01, 0.001\} \). For BeneFitter, we fix \( s \) at 1 and vary \( M/s \) for model selection. The hyper-parameter grid for our benefit formulation is \( M/s \in \{0.5, 1.0, 1.5, 2.0\} \times \max(L_i) \), where \( L_i \) is the length of training series \( i \). For the general multi-class problem we tune for an additional hyper-parameter \( \Delta \) to predict the class label. We set \( \Delta \in \{0.4, 0.5, 0.6, 0.7\} \) of the maximum difference between the expected benefit for the two-class labels for a given training series. BeneFitter outputs a decision when the predicted benefit is positive and at least \( \Delta \) higher compared to the predicted benefit of other labels.

For the learning task, we use the mean squared error loss function and Adam optimizer [10] for learning the parameters. The loss corresponding to class \( l \) is given by

\[
J = \frac{1}{|N_{train}|} \sum_{b_t \in B_l} \sum_{l_t=1}^{L_t} (\hat{b}_{lt} - b_{lt})^2
\]

where \( |N_{train}| \) is the number of total count of time steps for all the sequences in the training set, \( L_t \) is the length of each sequence, and \( \hat{b}_{lt} \) denotes the expected benefit for the class \( l \). We use Keras and Pytorch to implement our models.

A.2.2 Benchmark Experiments - Hyper-parameters. We compared the performance of BeneFitter against six competing methods on benchmark datasets. In Table A1, we report the different hyper-parameter configurations for each method that provides a trade-off between accuracy and earliness.

| Method       | Model Training Hyper-parameters       |
|--------------|----------------------------------------|
| ECTS         | support \( \in \{0.1, 0.2, 0.4, 0.8\} \)  |
| EDSC         | Chebyshev parameter \( \in \{2.5, 3.0, 3.5\} \)  |
| C-ECTS       | delay cost \( \in \{0.0005, 0.001, 0.005, 0.01\} \)  |
| RelClass     | reliability \( \in \{0.001, 0.1, 0.5, 0.9\} \)  |
| E2EL         | earliness trade-off \( \alpha \in \{0.6, 0.7, 0.8, 0.9\} \)  |
| BeneFitter   | \( M/s \in \{0.5, 1.0, 1.5, 2.0\} \times \max(\text{series length}) \)  |

A.3 Results

![Figure 6: Accuracy versus tardiness for sampled Endomondo.](image-url)