Detecting Artifactual Events in Vital Signs Monitoring Data

Partha Lal, Christopher K. I. Williams, Konstantinos Georgatzis
School of Informatics, University of Edinburgh, Edinburgh EH8 9AB

Christopher Hawthorne
Academic Unit of Anaesthesia, Pain and Critical Care Medicine
University of Glasgow, Glasgow, G31 2ER

Paul McMonagle
Institute of Neurological Sciences
Queen Elizabeth University Hospital
Glasgow G51 4TF

Ian Piper, Martin Shaw
Department of Clinical Physics
NHS Greater Glasgow and Clyde

October 19, 2015

1 Introduction

The presence of artifact in intensive care monitoring data is a major problem. For example, maintaining blood pressure in critically ill patients is a key management goal, and yet it is the physiological variable most prone to error. In addition to real-time monitoring, artifact detection is necessary for the proper audit or trial of therapies.

In this study we collect and annotate data from 27 intensive care unit (ICU) patients from the Southern General Hospital in Glasgow. Two models are compared for the detection, removal and cleaning of artifact in the vital signs data, namely the Factorial Switching Linear Dynamical System (FSLDS) and the Discriminative Switching Linear Dynamical System (DSLDS). We also consider a combination of the two, called the α-mixture (as described in sec. 7.3). Three types of artifactual events are considered: blood sample, damped trace (in the arterial line), and suction events. The area under ROC curve (AUC) scores for the detection of these events are: blood sample 0.95, damped trace: 0.79, suction 0.64 (α-mixture), with similar results for the FSLDS and DSLDS. The system is able run in realtime, and we discuss issues that had to be addressed to achieve this.

The structure of the rest of the paper is as follows: we describe the data collection, data preprocessing and data annotation processes in sections 2, 3 and 4. Section 5 evaluates what effect data cleaning can have on data summaries. In sections 6 and 7 we describe the FSLDS and DSLDS models that are used to make predictions of artifactual events in the data. In order to use these models we need to identify a period of stability, when no artifact is present; the resulting stability detector is presented in section 8. Section 9 describes the issues that needed to be addressed to make a real-time system, and section 10 gives details of the software produced for the project. Experimental results are given in section 11, and conclusions and future work are discussed in section 12.

2 Data Collection

Data was captured in the Neuro Intensive Care Unit (ICU) of the Southern General Hospital (SGH) in Glasgow. Signals collected were arterial blood pressure (ABP), electrocardiogram (ECG), pulse oximetry pulse, intracranial pressure (ICP), end tidal CO2 (EtCO2) and the respiratory signal (Resp) from the patient bedside monitor. Data sampling rates are signal dependent; the ECG signal is sampled at 500 Hz and all other channels sampled at 125 Hz. This raw waveform data was captured from the bed side ICU Philips Intellivue Monitors.

The data were captured in two different ways. For the first set of eight patients (labelled BioTBI), the waveform data was recorded onto a laptop computer connected into the bedside monitor. This system had some reliability problems, so that the data for a patient can be broken into a number of intervals, with gaps in between. So, for example, patient BioTBI001 has two records BioTBI001_1 and BioTBI001_2 in

our database. In total there were 17 data intervals recorded for the eight BioTBI patients.

Due to the unreliability of the above system, the waveform capture software ixTrend was purchased from ixellence GmbH (2015). Their "Netserver" software sits as a service on each of the Intellivue Monitor's embedded PCs and captures data from the monitor via the Medical Interface Bus (MIB) serial interface. Each minute, raw waveform data from all waveform channels is captured, compressed and sent via the local area network to a SQL Server database hosted on a local ICU server. Waveform data from each of the eight SGH Neuro ICU beds is continuously captured from the moment a patient is admitted to an ICU bed space and a valid patient identifier is entered onto the local bedside electronic record system (Philips ICCA). A system batch file running as an ixellence service detects when new patients are admitted into an ICU bed. Data collected in this fashion is labelled as CSO_0001 onwards in our dataset.

Data for eighty four patients were captured between December 2012 and January 2015. Of these, 27 patients were selected as suitable for analysis. The remaining 57 patients were excluded for the following reasons: i) Inappropriate admission pathology: N=23 (as study admission criteria were focused upon patients with TBI or SAH), ii) Insufficient data or channel type within first 48 hours of admission: N=16 (some patients that are admitted over weekend or at unsociable hours can have several days before annotation can begin), iii) Network Hardware failure: N=10 or Software down-sampling failures: N=2, iv) TBI patients excluded to ensure study design balance between SAH/TBI cohorts: N=3, v) Noisy data or no Events of Interest: N=2 vi) Patient refused study consent: N=1. Of the 27 patients 15 were traumatic brain injury (TBI) and 12 subarachnoid haemorrhage (SAH) patients.

In addition to the raw waveform data capture, additional clinical data useful for event interpretation was captured and reviewed as required from the Philips Medical ICU eRecord system (ICCA) to supplement the waveform data. This included TBI/SAH status, age, gender, etc.

3 Data Preprocessing

Although the waveform data is recorded at 125 or 500Hz, second-by-second summary data is more than adequate for condition monitoring of patient status, as has been shown e.g. by the neo-natal ICU work of Quinn (2008).

C++ code was written to down-sample the waveform quality data to second-by-second summary measures. The approach used is fully described in Shaw (2013). In brief, for each signal mentioned in sec. 2, an *index* channel is identified (as specified in Table 2 in Shaw 2013). The purpose of the index signal is to identify a physiologically meaningful interval over which to measure the signal. For example, ECG is used as the index for the ABP signal, and the ECG is processed to identify the R-R interval (the interval between ventricle depolarizations in the heart). Once an interval has been identified the signal is processed as appropriate, and the results are then interpolated to 1 Hz. For example for the ABP signal, the mean, diastolic (minimum pressure) and systolic (maximum pressure) channels are obtained per interval and then interpolated.

4 Data Annotation

The BioTBI patients plus CSO_0001 and CSO_0002 were annotated by CH. The aim was to annotate specific types of events that can affect the data quality and interpretation of the channels. Later, an experienced ICU nurse (PMc) was hired for a six month period to carry out annotation of further data collected under the CSO project. To gain experience, PMc annotated patients CSO_0001 and CSO_0002 independently, and then CH and PMc discussed the annotations together to produce a consensus annotation.

Where possible PMc carried out "live" annotation of patients admitted to the Neuro ICU of the SGH. A total of 8 patients in the study were annotated in this manner (CSO_0083, CSO_0099, CSO_0107, CSO_0112, CSO_0113, CSO_0115, CSO_0129, CSO_0158). However, as the annotator cannot be present 24/7 and to ensure consistency, these patients were re-annotated using recorded data to produce the dataset.

46 different annotation labels were used, as can be seen on the column headings of Table 1. Each event was annotated with a date, a start and end time, the event type, an indication of which signals are affected by the event, and a field for free text comments. These events include taking blood samples, damped traces, patient turning and suctioning. Table 1 shows the number of events of each type for each patient, along with summary statistics at the bottom. Table 2 shows the total duration (in seconds) for each event

type for each patient.

In Figure 1, for each patient-annotation combination, the area of the rectangle indicates the fraction of recorded time that the particular annotation was present for that patient. The most notable feature is that the damped trace events occupy a large fraction of time, particularly for the CSO patients. We see from Table 2 that the mean duration of damped trace events is almost 30,000 seconds (over 8 hours). In contrast, the average duration of blood sampling is around 450s per patient which is a very small fraction of the total duration, and so is not distinguishable from the dotted grid points in Figure 1.

Figures 5 and 6 show examples of damped trace and blood sample events.

5 The Effect of Data Cleaning on Data Summary Measures

Given the annotations described above, we can assess what effect cleaning the raw data will have on summary measures. Say for example we wish to compute the average of the systolic blood pressure (BP.sys) over 30 minute intervals (which could be useful for assessing trends in the blood pressure of the patient). If there are periods labelled as artifact during the 30 minute period, these are removed, and the average is computed only over the "clean" data in the interval.

The results of doing this can be visualized with a Bland-Altman plot (Altman and Bland, 1983), where the clean value is plotted on the x-axis, and the difference between the clean and raw values on the y-axis. The results for the various channels are shown in Figure 2 for all 27 available patients. The fraction of entries for which the differences are non-zero are as follows: Heart Rate 55%, Respiratory Rate 59%, Blood Pressure (diastolic) 59%, Blood Pressure (systolic) 70%, Blood Pressure (mean) 84%, Intracranial Pressure (diastolic) 56%, Intracranial Pressure (systolic) 75%, Intracranial Pressure (mean) 75%, End Tidal CO2 75%, Pleth 64%, Central Venous Pressure 64%. Hence for all channels well over 50% of the 30-minute summaries are affected in some way by artifact. Of course in many cases the difference may be small, although e.g. for BP.sys we observe the extreme differences can be more than +20 and -10 mmHg, which would certainly be clinically significant.

6 Factorial Switching Linear Dynamical System

We have used two different models for make inferences from the the time-series data, the Factorial Switching Linear Dynamical System (FSLDS), and a newer variant called the Discriminative Switching Linear Dynamical System (DSLDS). The FSLDS is described below, and the DSLDS in section 7.

The FSLDS is a latent variable model for time-series data, where each time step represents an observation that covers one second of patient observations. At time step t the model has a hidden discrete state variable s_t, a hidden continuous state variable x_t and continuous observations y_t, as illustrated in Fig. 3(left). The discrete state is factorial in nature — it is the cross product of the factors. For each time step t, given M factors $f_t^{(1)}...f_t^{(M)}$ the state s_t is $f_t^{(1)} \otimes ... \otimes f_t^{(M)}$. Factors are assumed to be independent of each other in the prior and to have Markovian dependence, i.e.

$$p(s_t|s_{t-1}) = \prod_{m=1}^{M} p(f_t^{(m)}|f_{t-1}^{(m)}). \tag{1}$$

If each factor $f^{(m)}$ can adopt one of $L^{(m)}$ values then there are K possible states, where $K = \Pi_{m=1}^{M} L^{(m)}$.

The model is Markovian, so that s_t is independent of $s_{t-2}, s_{t-3}...$ given s_{t-1}, and the transition probabilities are specified by a matrix where the element (i, j) equals $P(s_t = j|s_{t-1} = i)$.

The continuous state x_t evolves in a linear Gaussian fashion so that

$$p(x_t|x_{t-1}, s_t) \sim \mathcal{N}(A^{s_t}(x_{t-1} - \mu^{s_t}) + \mu^{s_t} + d^{s_t}, Q^{s_t}), \tag{2}$$

where A^s is the system matrix for state s, μ^s is the mean value for the latent state, d^s is the drift term, and Q^s is the corresponding system noise covariance matrix.

The observations y_t are derived from the continuous state with some additive Gaussian noise so

$$p(y_t|x_t, s_t) \sim \mathcal{N}(H^{s_t}x_t + o^{s_t}, R^{s_t}), \tag{3}$$

where H^s is the observation matrix, o^s is an offset term, and and R^s is the corresponding observation noise covariance matrix.

The joint distribution of the model is therefore

$$p(s_{1:T}, x_{1:T}, y_{1:T}) = p(s_1)p(x_1)p(y_1|x_1, s_1)\prod_{t=2}^{T} p(s_t|s_{t-1})p(x_t|x_{t-1}, s_t)p(y_t|x_t, s_t). \quad (4)$$

6.1 Factors

Each of the discrete variables (or factors) represents an event that can affect the observations. In the domain of interest here, this could include taking blood samples, endo-tracheal suctioning, or a damped trace. A factor variable can either be inactive or in one of a number of possible discrete states. The discrete state of the system is obtained from the full specification of the values of all factor variables. Each factor affects a specific subset of channels. There is one further factor, the X-factor, which is there to catch all unusual events that aren't modelled by any of the other factors. The X-factor can be either active or inactive.

Each factor affects a specified set of channels and leaves others unaffected — for instance the blood sample factor *only* affects the arterial blood pressure (ABP) channels. When two factors both affect the same channel, one takes precedence over the other. The precedence rules are set using prior knowledge about how factors interact.

Details of the models used for each factor and the preference rules are given in Appendix A.

6.2 Channels

The continuous observation space in this work consists of the values of several physiological variables of interest such as heart rate, respiration rate, systolic and diastolic arterial blood pressure and mean intracranial pressure.

Data on any channel can be missing, perhaps because of a disconnected sensor. If a channel has not been present up to a given point in time (e.g. if a sensor has not yet been attached) then it is ignored. This is done by setting the appropriate rows of all H matrices to zero. In addition, those parts of the output are set to `NaN`, overwriting the value that was inferred. If the channel *was* present at some time before but is currently missing (e.g. the sensor has been disconnected) then it is ignored in the same way but the estimated values are outputted.

6.3 Inference

The inference module performs filtering (rather than smoothing or prediction). Thus at each time step t it infers the latent state (s_t and x_t) given the observation history up to and including t ($y_{1:t}$).

In a model with no discrete state, Kalman filtering would consist simply of alternating prediction and correction steps (as described in Appendix A.1 of Murphy (1998) or Section 18.3.1 of Murphy (2012)). However, there is also a discrete latent state. Exact inference in this case scales exponentially in t (Lerner and Parr, 2001). This problem is dealt with in the code by applying the Gaussian Sum Approximation, referred to in Murphy (1998) as the Generalized Pseudo Bayesian algorithm of order 1, or GBP(1). It works by collapsing the K different Gaussians at a given time down to a single Gaussian using moment matching, as detailed in Murphy (1998).

7 Discriminative Switching Linear Dynamical System

The DSLDS was developed by Georgatzis and Williams (2015). The key idea is to make the prediction of discrete state s_t be a *discriminative* task based on the observations, and then to make the inference of continuous state x_t be dependent on the observations and the inferred discrete state.

We start by modelling $p(s_t|y_{t-l:t+r})$ with a discriminative classifier, where (features of) observations from the previous l and future r time steps affect the belief of the model about s_t. The inclusion of r frames of future context is analogous to fixed-lag smoothing in an FSLDS (see e.g. Särkkä, 2013, sec. 10.5). Inclusion of future observations in the conditioning set means that the DSLDS will operate with a delay of r seconds, since an output of the model at time t can be produced only after time $t + r$. However, provided that r is small enough ($r \leq$10s in experiments), this delay is negligible compared to the increase in performance. The LDS component can also be regarded from a similar discriminative viewpoint which allows us to model $p(x_t|x_{t-1}, y_t)$. The main advantage of this discriminative view is that it allows for a

rich number of (potentially highly correlated) features to be used without having to explicitly model their distribution or the interactions between them, as is the case in a generative model. A combination of these two discriminative viewpoints gives rise to the DSLDS graphical model in Figure 3(right).

The DSLDS is summarized by the equation

$$p(s, x|y) = p(s_1|y_1)p(x_1|s_1, y_1)\prod_{t=2}^{T} p(s_t|y_{t-l:t+r})p(x_t|x_{t-1}, s_t, y_t). \tag{5}$$

We have used the simplest assumption for $p(s_t|y_{t-l:t+r})$ that it factorizes, so that $p(s_t|y_{t-l:t+r}) = \prod_{m=1}^{M} p(f_t^{(m)}|y_{t-l:t+r})$.

7.1 Predicting s_t

We model the conditional probability of each factor being active at time t given the observations with a probabilistic discriminative binary classifier, so that $p(f_t^{(i)} = 1|y_{t-l:t+r}) = G(\phi(y_{t-l:t+r}))$, where $G(\cdot)$ is a classifier-specific function, and $\phi(y_{t-l:t+r})$ is the feature vector that acts as input to our model at each time step. Following Georgatzis and Williams (2015) we use a random forest classifier (Breiman, 2001). The output of the random forest for a new test point is an average of the predictions produced by each tree, where the prediction of each tree is the proportion of the observations that belong to the positive class in the leaf node in which the test point belongs to.

We use a variety of features to capture interesting temporal structure between successive observations. At each time step, a sliding window of length $l + r + 1$ is computed. For some features we also divide the window into further sub-windows and extract additional features from them. More precisely, the full set of features that are being used are: (i) the observed, raw values of the previous l and future r time steps $(y_{t-l:t+r})$; (ii) the slopes (calculated by least squares fitting) of segments of that sliding window that are obtained by dividing it in segments of length $(l + r + 1)/k$; (iii) an exponentially weighted moving average of this window of raw values (with a kernel of width smaller than $l + r + 1$); (iv) the minimum, median and maximum of the same segments; (v) the first order differences of the original window; and (vi) differences of the raw values between different channels. The hyperparameters of the method (number of trees in the forest, l and r were set by nested cross-validation, as described in Georgatzis and Williams (2015, sec. 2.4).

7.2 Predicting x_t

The form of $p(x_t|x_{t-1}, s_t, y_t)$ is chosen as

$$\begin{aligned} p(x_t|x_{t-1}, s_t, y_t) \quad &\propto \quad \exp\{-\frac{1}{2}\left((x_t - \mu^{s_t}) - (A^{s_t}(x_{t-1} - \mu^{s_t}) + d^{s_t})\right)^T (Q^{s_t})^{-1} \left((x_t - \mu^{s_t}) - (A^{s_t}(x_{t-1} - \mu^{s_t}) + d^{s_t})\right)\} \\ &\times \quad \exp\{-\frac{1}{2}(C^{s_t}x_t + o^{s_t} - y_t)^T (R^{s_t})^{-1}(C^{s_t}x_t + o^{s_t} - y_t)\} \ . \end{aligned} \tag{6}$$

This closely mimics the structure of the FSLDS, but there are differences in C^{s_t}. In the DSLDS, C^{s_t} consists of 0/1 entries, which are set based on our knowledge of whether the observations y_t are artifactual or not under state s_t. In the FSLDS, the corresponding observation model encodes the belief that the generated y_t should be normally distributed around $x_t + o^{s_t}$ with covariance R^{s_t}, whereas in our discriminative version, the observation model encodes our belief that $x_t + o^{s_t}$ should be normally distributed around y_t with covariance R^{s_t}. The idea behind this model is that at each time step we update our belief about x_t conditioned on its previous value, x_{t-1}, and the current observation, y_t, under the current regime s_t. For example, under an artifactual process, the observed signals do not convey useful information about the underlying physiology of a patient. In that case, we drop the connection between y_t and x_t (for the artifact-affected channels) which translates into setting the respective entries of C^{s_t} to zero. Then, the latent state x_t evolves only under the influence of the appropriate system dynamics parameters $(A^{s_t}, Q^{s_t}, \mu^{s_t}, d^{s_t})$. Conversely, operation under a non-artifactual regime incorporates the information from the observed signals, effectively transforming the inferential process for x_t into a product of two "experts", one propagating probabilities from x_{t-1} and one from the current observations. The A^s, Q^s, μ^s, d^s, o^{s_t} and R^s parameters are estimated as in the FSLDS.

For inference, similarly to the FSLDS we wish to compute $p(s_t, x_t|y_{1:t+r})$. According to our proposed model, $p(s_t|y_{t-l:t+r})$ can be inferred at each time step via a classifier as described in Section 7.1. However, exact inference for x_t is still intractable; as with the FSLDS we make use of the Gaussian Sum Approximation.

7.3 Combining the FSLDS and DSLDS predictions for s_t

The FSLDS and DSLDS can be run independently and in parallel. One way to combine their predictions for s_t is via an α-mixture (see Amari 2007), with

$$p_\alpha(s_t) = c\left(p_g(s_t)^{(1-\alpha)/2} + p_d(s_t)^{(1-\alpha)/2}\right)^{2/(1-\alpha)}, \tag{7}$$

where $p_g(s_t)$ and $p_d(s_t)$ are the outputs for the switch variable at time t from FSLDS and the DSLDS respectively, and c is a normalization constant. For $\alpha = -1$ we obtain a mixture of experts (with equally weighted experts), while for $\alpha \to 1$, the formula yields a product of experts. $\alpha \to \infty$ yields the minimum of the two probabilities, while $\alpha \to -\infty$ gives the maximum.

8 Stability Detection

One of the main purposes of the FSLDS is to detect artifact in observed data. For this to work we need some idea of what non-artifactual data looks like—the stability detector is trained to automatically label periods of non-artifactual data. This idea was introduced in Williams and Stanculescu (2011).

We first need to separate the idea of a channel and a signal—systolic arterial blood pressure (ABP.sys) and heart rate (HR) are examples of channels, ABP and HR are examples of signals. Thus a number of channels can be derived from the same underlying signal measurement. It is signals that are labelled as stable or not, using a selection of channels to make that decision.

This problem is set up as an artifact/non-artifact classification task, where an interval is labelled as artifactual if it overlaps with any artifactual event. Williams and Stanculescu (2011) found that a logistic regression model operating on a number of hand-crafted features was effective for this task, and this is the model used here. In the current work the mean, median, standard deviation, minimum and maximum of each signal channel in the interval are extracted for use as features.

The classifier is trained to minimize log loss, and its performance can be assessed with a ROC curve. However, the real mode of operation is rather different – intervals are considered one by one as they come in, and once a non-artifactual interval is identified it is used to train the stability model for the patient.[1]

The operation of this process given a trained detector depends on the threshold applied on the classifier; if it is too low one would expect that artifactual intervals would be accepted as “clean” ones, and if it is too high then the system waits forever and has no notion of stability, and therefore cannot produce useful output. To address this issue the accuracy and waiting times were assessed as a function of the threshold in a cross-validation procedure on the training data, and thresholds were chosen on a per signal basis to be as high as possible while minimizing the waiting time and obtaining good classification accuracy. The ultimate evaluation would be to ask about the quality of the inferences made by the FSLDS depending on the stability interval selected, but this is too hard to optimize directly.

In the experiments reported below the stability detector is trained in a leave-one-patient-out (LOPO) fashion—predictions for the stability of one patient make use of the data for all of the other patients in the dataset for training.

The work by Fawcett and Provost (1999) on the Activity Monitoring Operating Characteristic (AMOC) curve is somewhat related to this problem. However, in their work one is considering a rare event which may occur zero or one times for a particular patient in their monitoring record. In contrast, our data show that over 75% of intervals are classified as non-artifactual for at least one of the signals, so we are not in this rare-event regime.

9 Realtime Implementation

To run the system in real time, the four stages of (i) data extraction from the Neuro ICU database, (ii) data pre-processing, (iii) stability detection and (iv) FSLDS[2] operation all need to connect up and work together in real time. See Figure 4 for a graphic of this pipeline.

[1] To account for the changing condition of a patient in intensive care, a stability model expires after a certain period (the reset interval). At this point the model behaves as if no stability period is defined — this continues until a new stable period is received from the stability detector. The reset interval can be configured according to the problem domain or in light of expert clinical input.

[2] Currently the DSLDS is only implemented in Matlab and is not available for realtime use.

Operating on live data introduces new issues that weren't present in prior work, which used historical data. We briefly discuss those issues below.

Computational efficiency For the system to used live, inference needs to be performed at least as fast as real-time. Given that there will be other concurrent demands on the server, e.g. database access and signal preprocessing, the inference implementation needs to ideally be several times faster than real-time. The final implementation ran at approximately 10x real-time. In addition to common methods for speeding up code (minimizing disc access, passing by reference to avoid copying large amounts of data) this was achieved using:

Parallelization Elements of the inference are performed in parallel (using OpenMP). This was used wherever possible but most notably in the Gaussian Sum Approximation. Here the latent state is inferred in light of a new observation. The previous state is required for predicting the new current state but unknown, and so we perform the inference for all possible previous states and collapse the resulting Gaussian states together (Murphy, 1998). Those inferences can be performed independently and so were done in parallel, distributed across multiple cores.

Fast matrix libraries We perform a large number of matrix operations and so rather than implement them from scratch we used an existing library `eigen`[3]. This has been shown to compare favourably in performance benchmarks to other matrix libraries, see `http://eigen.tuxfamily.org/index.php?title=Benchmark`.

Stability model estimation When using historical data, the period of stability can be selected from the entire patient stay and then used to train a model that is used from the start of the stay. If, however, the system is used in a live setting we can only select from the data we have seen so far. The FSLDS model cannot be used until a stability model has been learnt and so one should be found as soon as possible, as discussed in section 8 above.

The stability model differs from artifactual models in that its parameters are estimated from the patient upon which we are performing inference; in contrast the parameters of the artifactual models are estimated from the training data. In a live scenario we are provided with artifactual models that were trained offline and a stability model that has been trained on-the-fly. Combining the two models requires care since for some artifactual models, parameters from the stability models should be used when a given channel is unaffected by the artifact — for instance heart rate channels are unaffected during a blood sample event.

Since a patient's condition changes over the course of their stay, the system allows for the stability model to be re-estimated periodically. Once a stability period has been identified then it used until a given period of time has passed[4]. Once that period has passed, the system invalidates the existing stability period and starts detecting a new one.

10 Software

Software implementing the methods described here is available via `http://dx.doi.org/10.7488/ds/300`.

Matlab code

Matlab FSLDS Matlab code for training and inference using the FLSDS model, as well as some utility scripts for examining data and inference outputs.

Stability detection Matlab code for training a logistic regression classifier for stability detection, as well as methods to extract model parameters for use in the realtime system

Demos Scripts are included that demonstrate the application of the FSLDS on example blood sample, damped trace and suction events. This provides any easy entry point to using the codebase.

Realtime code

Preprocessing A tool for extracting a 1Hz signal from high frequency clinical waveform data. C++ source code and documentation is available

[3] See `http://eigen.tuxfamily.org/`.

[4] This is configurable and defaults to twelve hours

Stability detection This component accepts the 1Hz signal provided by the preprocessor and, for each channel, determines whether the signal on that channel is free from artifact or not. It uses the model trained on the Matlab side above.

FSLDS C++ code for performing inference in the FSLDS model.

Tests The code is covered by a suite of unit tests, these are included with the code

Data storage Once the waveform data has left the database all derived data is stored on the filesystem as CSV files. This has the advantage of making the files portable and easy to understand but is inefficient in terms of disk space.

Communication The various components of the system need to communicate with each other, passing on information about, for example, new observations or detected stability periods. This is done using the filesystem — a file is shared between the source and target process of any message and serialised JSON objects are appended to that file.

The above methods may be sufficient for a prototype but a more robust system would use a database instance for storing data and not rely on the filesystem for inter-process communication.

11 Experiments

We ran the FSLDS and DSLDS models on the data collected from the 27 patients. They were set up with factors to model blood sample, damped trace, suction and X. Ground truth for the X-factor is obtained from the full annotations—if there are annotations present at a given time which do not correspond to blood sample, damped trace or suction, then the X-factor is deemed to be active at that time. The evaluation was done in a leave-one-patient-out (LOPO) fashion, so predictions for a given patient can make use of the data for all of the other patients as training data.

At each second the models output posterior probabilities for each factor $p(f_t^{(m)}|y_{1:t}),\ m = 1, \ldots, M$, and the estimate of the state $p(x_t|y_{1:t})$. $p(x_t|y_{1:t})$ is a mixture of Gaussians — when visualizing outputs we shsow the weighted mean of the components and the overall variance of the mixture, which can be easily displayed with, for example, a line graph and error bars.

Examples of inferences are shown in Figures 5 and 6 for damped trace and blood sample events respectively. On the damped trace example the FSLDS nicely detects the first part of the event (where the systolic and diastolic blood pressures are very close), but erroneously detects a blood sample (instead of a damped trace) in the latter part of the event. It also erroneously detects a suction event throughout the trace. The X-factor fires correctly at the end of the trace, but also erroneously at the beginning. Notice how beliefs about the systolic and diastolic BP are maintained during the time that the damped trace and blood sample factors are active, as shown by the lighter coloured traces. In contrast the DSLDS correctly detects a damped trace throughout the event. The blood sample factor is correctly off the whole time, and the suction factor is correctly near to zero. The X-factor is quite active correctly near the end of the trace, but also erroneously at the beginning.

Looking at the blood sample example in Fig. 6 we see that the FSLDS model divides this event up between the blood sample and damped trace factors being active. In addition the X-factor is active for most of the time. Again notice how inference for the continuous variables (channels) works in the artifactual ramp, zeroing and flushing stages of the blood sample. For the DSLDS, the blood sample factor is active for the majority of the time the event is happening, but we also see that the X-factor is incorrectly active for most of the time.

As well as example inferences, we can also produce summaries of the performance. We plot a Receiver Operating Characteristic (ROC) curve for each factor, aggregating information over all times and all patients. Each ROC curve can be summarized by the area under the ROC curve (AUC). Figure 7 shows the ROC curves for the FSLDS, DSLDS and α-mixture for each of the four factors. Table 5 shows overall results for each factor. The best results (obtained from the α-mixture) are AUC scores are 0.95 for blood sample, 0.79 for damped trace, 0.64 for suction and 0.61 for the X factor.

The performance obtained for blood sample is very good, suggesting that this can be detected with high confidence. This is potentially useful for silencing false alarms. Even though the nurse is present at the

bedside during a blood sample procedure and hence knows that the alarm is false, reducing unnecessary alarms would help reduce “alarm fatigue”.

The damped trace performance is good. This is a particularly interesting case, as it is not an event caused by nursing interventions, and therefore it is particularly helpful to flag up. It would be very useful to identify such events automatically in order to prompt the nursing staff to correct the problem. Also, assessing the quality of the blood pressure data being recorded would be very important if automatic charting is in use.

For suction and X-factor the performance is not much better than random (which has an AUC of 0.5). Suction events are complex and have a variable time course, which may explain the difficulty in predicting them. Also note that suction and position change events can have similar effects on the patient, due to movement of the endo-tracheal tube, and that position change was not modelled with a factor in our experiments. Thus it may not be surprising if these two event types are confused, which may explain the poorer performance for suction events.

As well as looking at the results aggregated over patients, we can also perform a more detailed analysis, as shown in Tables 3 and 4 for the FSLDS and DSLDS respectively. For the blood sample event by comparing the tables line by line we see that the DSLDS performance is generally much better, giving a higher AUC on 21 out of 27 of the cases, and avoiding the low scores obtained with the FSLDS. For the other factors the results are generally quite similar between the two, in line with Table 5.

12 Conclusions and Future Work

In this project we have collected and annotated a valuable dataset of neuro ICU data, which can be made available to *bona fide* researchers on request, subject to regulatory approval. We were successful in implementing a real-time system carrying out FSLDS analysis on the raw data coming from the ICU, as described in section 9. We are making available the code for stability detection and the FSLDS in matlab and C++, and the preprocessing code in C++.

The Bland-Altman plots in section 5 show that for all channels in over 50% of the time there is a difference between the raw and cleaned averages obtained over a 30 minute interval. This illustrates that artifact contamination is an important problem.

We have evaluated the FSLDS and DSLDS models for the task of predicting blood sample, damped trace, suction and X-factor events. The AUC scores for the α-mixture are very high for blood samples (0.95), good for damped trace (0.79), and poor for suction (0.64) and X-factor (0.61) events. This combination method slightly outperforms the individual DSLDS or FSLDS models. The damped trace is a particularly interesting case, as this is not an event caused by nursing interventions, and therefore it is particularly helpful to flag up. We have also seen that it is the event class that dominates in terms of time (on average over 8 hours per patient).

Of course these results have been obtained from one specific ICU, and it will be important to assess the model’s performance in other patient populations and different centres to determine its robustness.

In terms of displaying the results to clinicians, we believe that plots like Figures 5 and 6 will be useful. It would be very dangerous to delete the raw data, but we can display the imputed data with error bars during artifactual periods, and show in greyscale the probability of artifactual factors being active.

In this paper we have evaluated the performance on a second-by-second basis using ROC curves. However, it would be useful to look at evaluation in an episode-based fashion (how well did we pick up a given event that lasted say 5 minutes?), as has been studied in Stanculescu *et al.* (2014, sec. IV.C).

We have focussed on using the FSLDS/DSLDS for detecting artifact, but note that it can be used more generally; for example Stanculescu *et al.* (2014) used an extended FSLDS model to detect sepsis in neonates, and more generally one can model changes in the patient’s state of health.

A Appendix: Models for each Factor

In this section we provide further details of the models used for stability, and for the blood sample, damped trace, suction, patient handling and X-factor events.

A.1 Stability

When none of the factors are active we are in a period of stability. Pulsatile channels are modelled with a relative AR model (as described in section 9.4.1 of Quinn and Williams 2011) consisting of an $AR(2)$ baseline and an $AR(2)$ signal. The filter that separate the baseline and signal components is a moving average filter with a window of width 3. Respiration rate, pleth rate and end-tidal CO_2 are instead modelled with a simple $AR(2)$ process.

Model parameters for each channel are estimated from the annotated stability period. Initial values are derived using the Yule-Walker equations and then updated using three iterations of expectation-maximisation (Ghahramani and Hinton, 1996). If EM results in a value for the system matrix A which is non-stationary[5] then we revert to the initial value. Observation noise variance is an exception here, it set to a fixed value of 10.

A.2 Blood Sample Events

The blood sample factor consists of four stages: ramp, zero, flush, and a fourth stage for periods within a blood sample event that appear the same as stability. A 5×5 transition matrix between these four stages and stability is estimated from training data.

The ramp model is detailed in section 9.4.2 of Quinn and Williams (2011).

During the zeroing stage the pressure transducer is being recalibrated through exposure to air. ABP drops to approximately zero and no trace of the patient's true ABP is visible. This is implemented by decoupling the state from the observations with H set to zero for the rows corresponding to the ABP channel, and setting the offset term to be the mean value observed during zeros. System noise covariance is unchanged but observation noise variance for each channel is set to the observed variance of the channel, as measured during zeroing events (this is multiplied by a scaling factor of 0.05). Since the boundaries of zeroing events aren't precise, the initial and final 20% of the event is excluded, both for observation variance and offset estimation purposes.

Typically towards the end of a blood sample, the arterial line is flushed. ABP rises to approximately 250–300 mmHg and no trace of the patient's true ABP is visible. This is implemented by decoupling the state from the observations with H set to zero for the rows corresponding to the ABP channel and setting the offset term to be the mean value observed during flushes. System noise covariance is unchanged but observation noise variance for each channel is set to the scaled variance of the channel, as measured during flush events (this is multiplied by a scaling factor of 0.05). Since the boundaries of flush events aren't precise, the initial and final 20% of the event is excluded, both for observation variance and offset estimation purposes.

For the "stability within a blood sample" stage the parameters are simply copied from the stability model. Transitions to that stage from stability are prohibited by setting zeros in the transition matrix.

A.3 Damped Trace Events

During a damped trace event there is an occlusion in the arterial line, typically causing the pulse pressure (the difference between the systolic and diastolic pressures) to drop to near zero. The systolic and diastolic pressures converge to the value that the mean pressure held before the event. The mean ABP signal is also somewhat damped in comparison to stability, and so the parameters of the AR model are re-estimated.

The $AR(2)$ model for mean ABP is estimated from the labelled damped trace events. The observation noise variance R for systolic ABP is the median variance of the difference between systolic and mean ABP — the diastolic value is computed similarly. The observation model H is such that elements for systolic and diastolic ABP are set to zero and only mean ABP is taken from the continuous state variable. The system model A for mean ABP is as learnt from the labelled events (using the Yule-Walker equations and EM) but remains unchanged for systolic and diastolic ABP channels.

[5]The system matrix A is non-stationary if the absolute value of any of its eigenvalues is greater than or equal to 1

A.4 Suction Events

For these purposes “suction - endo-tracheal” and “coughing” are both treated as suction events. Based on an analysis of the annotation files, only heart rate, respiration rate, pleth rate and end tidal CO_2 are understood to be affected during suction events.

Model parameters are re-estimated using the labelled suction events, Yule-Walker equations are used to produce an initial value for three iterations of EM. If EM results in a system matrix A that is non-stationary then we revert to the initial value. Observation noise variance is, as for stability, fixed to the value of 10.

A.5 X-factor

The X-factor (see Section III.A of Quinn, 2008) is used to account for all unusual observations that aren't already explained by one of the existing factors. The model parameters for the X-factor consist of the model for stability but with an inflated system noise covariance Q. The amount by which to inflate Q is the parameter ξ. Its value is learned by the Matlab code using equation 10 in Quinn (2008).

A.6 Overwriting Order of Factors

When multiple factors are active at a given time, we use the notion of an ordering of the factors to determine which one affects each signal, as in Quinn (2008).

The ordering used here is

$$\mathrm{X-factor} < \mathrm{handling} < \mathrm{suction} < \mathrm{blood\ sample} < \mathrm{damped\ trace} \tag{8}$$

where $f^{(a)} < f^{(b)}$ means that the parameters from $f^{(b)}$ can overwrite those set by $f^{(a)}$.

Acknowledgements

This work was funded by grant number CHZ/4/801 from the Chief Scientist Office (Scotland): Improving Decision Support for Treating Arterial Hypotension in Adult Patients During their Management in Intensive Care, May 2013-Apr 2015. The work of Konstantinos Georgatzis was supported by the Scottish Informatics and Computer Science Alliance (SICSA). Chris Hawthorne and Ian Piper recieved funding from AAGBI/Anaesthesia which allowed collection of pilot data for this project. We thank the Intensivists Prof Peter Andrews, Mr Laurence Dunn and Prof John Kinsella for their feedback in our review meetings, which helped to keep the project focussed on the important questions.

References

Altman, D. G. and Bland, J. M. (1983). Measurement in Medicine: The Analysis of Method Comparison Studies. *Journal of the Royal Statistical Society. Series D (The Statistician)*, **32(3)**, 307–317.

Amari, S.-i. (2007). Integration of Stochastic Models by Minimizing α-Divergence. *Neural Computation*, **19**(10), 2780–2796.

Breiman, L. (2001). Random Forests. *Machine Learning*, **45**(1), 5–32.

Fawcett, T. and Provost, F. (1999). Activity Monitoring: Noticing interesting changes in behavior. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 53–62.

Georgatzis, K. and Williams, C. K. I. (2015). Discriminative Switching Linear Dynamical Systems applied to Physiological Condition Monitoring. In M. Meila and T. Heskes, editors, *Proceedings of the 31st Annual Conference on Uncertainty in Artificial Intelligence*.

Ghahramani, Z. and Hinton, G. E. (1996). Parameter estimation for linear dynamical systems. Technical report, University of Toronto.

ixellence GmbH (2015). ixTrend. `https://www.ixellence.com/index.php/en/products/ixtrend`.

Lerner, U. and Parr, R. (2001). Inference in Hybrid Networks: Theoretical Limits and Practical Algorithms. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*, pages 310–318.

Murphy, K. P. (1998). Switching Kalman Filters. Technical report. http://www.cs.ubc.ca/ murphyk/Papers/skf.pdf.

Murphy, K. P. (2012). *Machine Learning: a Probabilistic Perspective*. MIT Press.

Quinn, J. A. (2008). *Bayesian Condition Monitoring in Neonatal Intensive Care*. Ph.D. thesis, University of Edinburgh.

Quinn, J. A. and Williams, C. K. I. (2011). Physiological monitoring with factorial switching linear dynamical

systems. In D. Barber, A. Cemgil, and S. Chiappa, editors, *Bayesian Time Series Models*, pages 182–204. Cambridge University Press.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge University Press.

Shaw, M. (2013). A concisise description of the clinical waveform pre-processing workflow. Manuscript available on request.

Stanculescu, I. A., Williams, C. K. I., and Freer, Y. (2014). Autoregressive Hidden Markov Models for the Early Detection of Neonatal Sepsis. *IEEE Journal of Biomedical and Health Informatics*, **18(5)**, 1560–1570.

Williams, C. K. I. and Stanculescu, I. (2011). Automating the calibration of a neonatal condition monitoring system. In M. Peleg, N. Lavrac, and C. Combi, editors, *AIME*, volume 6747 of *Lecture Notes in Computer Science*, pages 240–249. Springer.

Table 1: Table showing the number of events of each type for each patient, along with summary statistics at the bottom. The total column excludes dysrhythmia, since it is not an artifact.

Patient	admission to ICU	aspiration ngt	blood sample - abp	blood sample - cvp	central line insertion	change closed suction system	coughing	damped trace - abp	dysrhythmia	extubation	fluid bolus	fluid/drugs bolus via cvc	flush - abp	flush - cvp	flush - ngt	intubation	medical examination	medication administration - iv bolus	medication administration - iv infusion	medication administration - topical	mimor - eye care	mimor - mouth care	mimor - other	neuro obs	noisy - abp	noisy - cvp	noisy - ecg	noisy - pleth	noisy - resp	other	physiotherapy - chest	physiotherapy - other	position change - log rolling	position change - other	position change - turning	preparation for transfer	ramp - abp	return from transfer	suction - endo-tracheal	suction - oral	ventilation mode change	ventilator change - circuit	ventilator change - machine	zero - abp	zero - co2	zero - cvp	Total
BioTBI001_1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
BioTBI001_2	0	0	4	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	13	0	0	0	0	5	0	0	0	9	0	0	0	0	2	0	0	38
BioTBI003_1	0	0	2	0	0	0	0	1	1	0	0	0	3	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	33
BioTBI003_2	0	0	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	14
BioTBI003_3	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	4	0	0	0	0	0	0	0	16
BioTBI004_1	0	0	4	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	9	2	0	0	0	0	1	11	0	0	0	0	4	0	0	0	5	0	0	1	0	3	0	0	41
BioTBI004_2	0	0	3	0	0	0	0	1	1	0	3	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	13
BioTBI005_1	0	0	3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	4	0	0	0	0	2	0	2	23
BioTBI005_2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	11
BioTBI007	0	0	1	0	0	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	8	0	0	0	0	1	0	0	0	11	0	0	0	0	2	0	0	30
BioTBI009_1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
BioTBI009_2	0	0	2	0	0	0	7	20	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	33
BioTBI009_3	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
BioTBI010_1	0	0	5	0	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	1	0	0	11	0	0	0	0	2	0	0	0	5	0	0	1	0	2	2	1	37
BioTBI010_2	0	0	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	2	0	0	0	0	0	1	0	0	5	0	0	0	0	1	5	2	23
BioTBI010_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
BioTBI018	0	0	15	1	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	12	1	0	0	1	0	0	27	0	0	2	0	0	1	0	0	15	0	1	0	0	0	2	0	93
cso_0001	0	0	6	0	0	0	0	7	0	0	0	0	0	0	0	0	1	0	0	0	0	5	0	0	0	0	0	0	33	0	0	0	0	4	3	0	0	7	0	1	0	0	3	3	0	73
cso_0007	0	0	15	0	0	0	0	27	1	0	0	0	20	0	0	0	1	0	0	0	0	0	22	0	0	0	0	0	29	1	0	0	4	8	1	0	2	12	0	1	0	0	6	8	0	157
cso_0008	0	0	11	0	0	0	17	6	2	1	0	0	5	0	0	0	0	0	0	0	0	1	18	3	0	0	5	0	3	31	2	0	0	6	13	1	0	1	10	0	0	0	0	2	2	0	138
cso_0020	0	0	6	0	0	0	4	7	1	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	4	0	1	0	5	0	0	0	0	0	4	1	1	0	17	0	0	0	0	2	4	0	62
cso_0021	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	2	0	0	0	0	0	0	3	0	3	0	0	0	0	1	0	0	15
cso_0027	0	0	3	0	0	0	0	16	3	0	0	0	30	0	0	0	0	0	0	0	0	13	0	42	0	3	46	1	18	3	0	0	1	7	5	15	7	24	0	0	0	0	18	4	0	256
cso_0029	1	0	4	0	0	0	4	8	1	0	0	0	25	0	0	0	0	0	0	4	3	0	0	36	0	11	45	0	1	1	0	0	0	11	1	25	1	11	0	0	0	0	20	5	0	217
cso_0036	1	0	4	0	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	10	0	5	0	4	0	0	0	0	0	2	2	0	1	12	0	0	0	0	2	2	0	50
cso_0041	0	0	12	0	0	0	0	2	0	0	0	0	7	0	0	0	0	0	0	0	0	2	0	3	0	1	3	13	4	1	0	0	0	0	2	1	2	14	0	0	0	0	4	4	0	75
cso_0064	0	0	3	0	0	0	0	12	1	0	0	0	3	1	0	0	1	0	0	0	0	0	0	2	2	7	7	38	0	2	0	0	0	1	4	1	0	0	0	0	0	0	3	0	2	89
cso_0083	0	0	7	0	0	0	2	25	0	2	0	0	14	0	0	1	3	0	0	0	0	0	2	3	23	0	11	83	57	11	0	0	0	4	8	2	4	2	1	0	0	0	8	1	0	274
cso_0086	0	0	1	0	0	0	4	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	2	4	1	0	0	1	2	1	0	1	6	0	0	0	0	2	0	0	29
cso_0099	0	0	7	0	0	0	4	10	0	0	0	0	8	0	0	0	4	0	0	0	0	0	3	1	6	0	3	1	3	10	0	1	0	2	4	0	1	0	6	0	0	0	0	0	0	0	74
cso_0107	0	0	3	0	0	0	2	7	0	1	0	0	2	0	0	0	3	0	0	0	0	1	2	4	3	0	0	6	11	18	2	0	0	6	2	0	0	0	1	0	0	0	1	0	0	75
cso_0112	1	0	2	0	0	0	0	4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	18	0	2	4	3	0	0	0	0	0	1	1	1	1	1	0	0	0	1	1	0	0	45
cso_0113	0	0	8	0	0	0	2	19	0	0	0	0	6	0	0	0	2	0	0	2	2	1	0	3	5	0	3	0	6	23	1	0	0	3	2	2	0	2	25	1	0	0	0	5	0	0	123
cso_0115	1	3	14	0	1	3	8	6	0	0	1	1	2	1	0	0	13	3	1	0	6	8	6	13	15	5	3	4	2	68	3	0	4	10	7	1	0	1	28	6	0	0	0	4	2	1	255
cso_0123	1	0	10	0	0	0	0	46	1	1	0	0	9	0	0	2	0	0	0	0	0	0	2	0	39	0	12	19	12	3	0	0	0	1	5	2	4	2	9	0	0	0	0	9	2	0	190
cso_0129	0	0	9	0	0	1	9	8	0	1	1	0	3	0	0	0	8	0	0	0	6	5	1	8	17	0	15	11	2	20	3	1	0	5	10	2	0	2	27	5	1	0	0	4	3	0	188
cso_0158	0	3	13	0	0	0	0	9	0	0	0	0	5	0	1	0	5	0	3	2	1	1	1	6	10	0	0	19	0	18	1	0	1	1	0	3	0	2	0	1	0	0	0	6	2	0	114
cso_0172	0	0	4	0	0	0	40	23	0	0	0	0	13	0	0	0	0	0	0	0	0	1	0	0	14	0	1	12	0	4	1	0	0	1	10	2	0	2	25	0	0	0	0	4	3	0	160
min	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
mean	0.2	0.2	5.1	0.0	0.1	0.1	3.2	7.6	0.4	0.2	0.2	0.0	4.5	0.1	0.0	0.1	1.1	0.1	0.1	0.1	0.5	0.6	3.3	1.2	6.5	0.3	2.3	7.7	3.3	11.2	0.5	0.1	0.2	1.2	3.1	0.9	1.4	0.8	8.1	0.3	0.1	0.1	0.0	3.1	1.4	0.2	81.2
median	0.0	0.0	4.0	0.0	0.0	0.0	0.0	3.5	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	0.0	0.0	0.0	0.0	2.0	1.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	47.5
max	1	3	15	1	1	3	40	46	3	2	3	1	30	1	1	2	13	3	3	2	6	8	22	13	42	7	15	83	57	68	3	1	4	10	13	5	25	7	28	6	1	1	1	20	8	2	274
std.dev.	0.4	0.7	4.4	0.2	0.2	0.5	7.1	10.1	0.7	0.4	0.5	0.2	7.2	0.3	0.2	0.4	2.6	0.5	0.5	0.5	1.5	1.6	5.2	2.7	11.4	1.4	3.9	17.5	9.6	13.4	0.9	0.2	0.7	2.3	3.7	1.1	4.7	1.3	8.4	1.3	0.3	0.2	0.2	4.5	2.0	0.6	78.6

Table 2: Table showing the total duration in seconds for each event type for each patient, along with summary statistics at the bottom. The total column indicates the percentage of the patient stay affected by artifact, excluding dysrhythmia as it is not an artifact.

Patient	admission to ICU	blood sample - abp	blood sample - cvp	central line insertion	change closed suction system	coughing	damped trace - abp	dysrhythmia	extubation	fluid bolus	fluid/drugs bolus via cvc	flush - abp	flush - cvp	intubation	medical examination	medication administration - topical	minor - eye care	minor - mouth care	minor - other	neuro obs	noisy - abp	noisy - cvp	noisy - ecg	noisy - pleth	noisy - resp	other	physiotherapy - chest	physiotherapy - other	position change - log rolling	position change - other	position change - turning	preparation for transfer	ramp - abp	return from transfer	suction - endo-tracheal	suction - oral	ventilation mode change	ventilator change - circuit	ventilator change - machine	zero - abp	zero - co2	zero - cvp	Total
BioTBI001_1	0	96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.4
BioTBI001_2	0	510	0	0	0	53	765	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	555	0	0	0	0	496	0	0	0	277	0	0	0	0	150	0	0	0.0
BioTBI003_1	0	110	0	0	0	0	3292	6	0	0	0	60	0	0	0	0	0	0	1008	0	0	0	0	0	0	1790	0	0	0	0	0	0	0	0	231	0	0	0	0	0	0	0	23.2
BioTBI003_2	0	202	0	0	0	74	0	0	0	0	0	0	0	0	0	0	0	0	61	0	0	0	0	0	0	91	0	0	0	0	0	0	0	0	310	0	0	0	0	0	0	0	1.6
BioTBI003_3	0	0	0	0	0	371	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	478	0	0	0	273	0	0	0	0	0	0	0	3.2
BioTBI004_1	0	275	0	0	0	0	0	3	0	0	0	0	0	0	362	0	0	0	805	4694	0	0	0	0	985	3598	0	0	0	0	1819	0	0	0	580	0	0	46	0	155	0	0	19.4
BioTBI004_2	0	244	0	0	0	0	29	16	0	1022	0	0	0	0	0	0	0	0	776	0	0	0	0	0	0	776	0	0	0	0	0	0	0	0	98	0	0	0	0	0	0	0	6.1
BioTBI005_1	0	238	0	0	0	0	0	0	0	0	0	70	70	0	0	0	0	0	93	0	0	0	0	0	0	577	0	0	0	0	0	0	0	0	456	0	0	0	0	87	0	87	1.9
BioTBI005_2	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	615	0	0	0	0	0	0	0	0	349	0	0	0	0	0	0	0	2.8
BioTBI007	0	253	0	0	0	90	591	0	0	0	0	0	0	0	0	0	0	0	1666	0	0	0	0	0	0	9867	0	0	0	0	499	0	0	0	3857	0	0	0	0	39	0	0	21.6
BioTBI009_1	0	0	0	0	0	0	0	6144	0	0	0	0	0	0	0	0	0	0	67	0	0	0	0	0	0	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
BioTBI009_2	0	117	0	0	0	1501	2836	0	0	0	0	12	0	0	0	0	0	0	79	0	0	0	0	0	0	79	0	0	0	0	0	0	0	0	724	0	0	0	0	0	0	0	30.8
BioTBI009_3	0	42	0	0	0	0	0	2314	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
BioTBI010_1	0	706	0	1982	0	69	326	0	0	0	0	0	0	0	0	0	0	143	145	0	0	0	0	1195	0	3379	0	0	0	0	1544	0	0	0	2467	0	0	272	0	52	62	34	15.5
BioTBI010_2	0	410	0	0	0	0	0	0	0	380	0	0	0	0	0	0	35	0	232	0	0	0	0	0	0	393	0	0	0	0	0	4935	0	0	1308	0	0	0	0	38	46	77	45.5
BioTBI010_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9
BioTBI018	0	1090	214	0	0	0	15169	0	0	0	0	0	0	0	0	0	0	0	13890	356	0	0	12080	0	0	24227	0	0	1887	0	0	2410	0	0	8190	0	0	0	0	0	23	0	32.7
cso_0001	0	613	0	0	0	0	7912	0	0	0	0	0	0	0	37	0	0	0	1361	0	0	0	0	0	0	7676	0	0	0	0	1625	16326	0	0	2280	0	1347	0	0	628	56	0	47.5
cso_0007	0	991	0	0	0	0	62909	17	0	0	0	532	0	0	211	0	0	0	1737	0	0	0	0	0	0	12298	850	0	0	2162	2165	5250	0	244	197	0	10	0	0	101	84	0	48.9
cso_0008	0	739	0	0	0	1334	2406	8	66	0	0	121	0	0	0	0	0	14	3826	193	0	0	2593	0	3210	7434	1019	0	0	2348	3565	12758	0	1209	2116	0	0	0	0	42	21	0	73.5
cso_0020	0	866	0	0	0	29	73925	8	0	0	0	15	0	0	0	0	0	0	0	0	1051	0	696	0	815	0	0	0	0	0	1023	34	98	0	1412	0	0	0	0	65	49	0	77.7
cso_0021	0	141	0	0	0	10	0	0	0	0	0	7	0	0	0	0	0	0	5	0	0	0	3039	0	0	330	0	0	0	0	0	0	57	0	130	0	0	0	0	2	0	0	44.1
cso_0027	0	425	0	0	0	0	8727	24	0	0	0	160	0	0	0	0	0	0	326	0	2530	0	241	2991	285	1472	919	0	0	150	764	2733	242	25888	1539	0	0	0	0	2017	115	0	75.9
cso_0029	0	333	0	0	0	83	126384	184	0	0	0	90	0	0	0	0	1163	1207	0	0	1799	0	11038	6531	0	286	286	0	0	0	3764	2564	327	641	1386	0	0	0	0	202	80	0	86.0
cso_0036	0	300	0	0	0	0	100	0	0	0	0	29	0	0	0	0	0	0	0	0	1007	0	494	0	541	0	0	0	0	0	394	138	0	729	582	0	0	0	0	56	54	0	64.8
cso_0041	0	878	0	0	0	0	171976	0	0	0	0	20	0	0	0	0	0	0	73	0	308	0	14	190	1157	710	569	0	0	0	0	926	97	1841	1098	0	0	0	0	107	3825	0	99.6
cso_0064	0	288	0	0	0	0	50621	4	0	0	0	9	8	0	483	0	0	0	0	203	50	24711	253	4897	0	3056	0	0	0	56	1072	5	0	0	0	0	0	0	0	45	0	43	75.7
cso_0083	0	728	0	0	0	50	161923	0	6	0	0	54	0	180	120	0	0	0	0	85	2983	0	1090	5946	16390	3178	0	0	0	3178	1527	0	280	0	72	0	0	0	0	995	20	0	98.4
cso_0086	0	87	0	0	0	368	6600	204	0	0	0	0	0	0	235	0	0	0	0	0	33	0	0	44	82	7260	897	0	0	1467	830	19	0	291	188	0	0	0	0	57	0	0	62.9
cso_0099	0	883	0	0	0	70	104705	0	0	0	0	38	0	0	240	0	0	0	0	0	365	0	637	175	678	3920	0	0	0	473	1284	0	4	0	267	0	0	0	0	0	0	0	89.9
cso_0107	0	200	0	0	0	373	3760	0	0	0	0	28	0	0	60	0	0	33	38	228	22	0	0	348	1170	2148	1895	0	0	150	476	0	0	0	39	0	0	0	0	14	0	0	25.1
cso_0112	0	166	0	0	0	0	2037	0	0	0	0	25	0	0	0	0	0	0	0	0	600	0	30	243	173	0	0	0	0	0	522	13	23	0	50	0	0	0	14	9	0	0	57.5
cso_0113	0	913	0	0	0	70	42186	0	0	0	0	103	0	0	420	47	0	0	0	62	355	0	94	0	1559	3271	480	0	0	1391	156	1037	0	0	1241	0	0	0	0	118	0	0	33.9
cso_0115	0	964	0	660	57	175	3139	0	0	0	237	15	27	0	1920	0	71	71	14	0	680	1677	48	197	44	10373	2760	0	120	95	2218	306	0	0	4332	34	0	0	0	58	161	6	13.9
cso_0123	30	950	0	0	0	0	27229	21	275	0	0	61	0	1830	0	0	0	0	251	0	4553	0	1875	2913	2382	1830	0	0	0	1579	1469	6538	281	336	1146	0	0	0	0	282	2493	0	38.4
cso_0129	0	927	0	0	15	411	107340	0	0	0	0	58	0	0	1080	0	0	0	0	127	787	0	395	549	217	10172	1260	16	0	571	2568	22	0	0	1409	133	123	0	0	111	240	0	55.8
cso_0158	0	1194	0	0	0	0	13230	0	0	0	0	32	0	0	0	0	0	0	0	0	510	0	0	994	0	37362	720	0	600	1080	0	44	0	0	0	120	0	0	0	148	68	0	86.1
cso_0172	0	455	0	0	0	2093	125308	0	0	0	0	133	0	0	0	0	0	73	0	0	783	0	35	755	0	796	242	0	0	12	2684	0	0	0	2128	0	0	0	0	82	559	0	80.3
min	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mean	0.8	458.8	5.6	69.5	1.9	190.1	29616.4	235.6	9.1	36.9	6.2	44.0	2.8	52.9	136.0	1.2	33.4	40.6	696.9	156.5	484.6	694.4	943.3	704.6	781.3	4230.1	313.1	0.4	68.6	387.2	866.9	1475.2	37.1	820.5	1071.9	7.6	38.9	8.4	0.4	148.9	209.4	6.5	40.6
median	0.0	316.5	0.0	0.0	0.0	0.0	2987.5	0.0	0.0	0.0	0.0	13.5	0.0	0.0	0.0	0.0	0.0	0.0	21.5	0.0	0.0	0.0	0.0	0.0	0.0	1294.0	0.0	0.0	0.0	0.0	487.0	0.0	0.0	0.0	402.5	0.0	0.0	0.0	0.0	48.5	0.0	0.0	36.1
max	30	1194	214	1982	57	2093	171976	6144	275	1022	237	532	70	1830	1920	47	1163	1207	13890	4694	4553	24711	12080	6531	16390	37362	2760	16	1887	3178	3764	16326	327	25888	8190	133	1347	272	14	2017	3825	87	99.6
std.dev.	4.9	364.3	34.7	336.1	9.5	456.0	49802.6	1053.6	45.6	175.3	38.4	91.6	12.1	297.5	362.7	7.6	188.6	196.3	2320.9	760.3	974.4	4010.5	2637.9	1666.1	2696.9	7383.4	608.4	2.6	318.6	787.5	1053.5	3538.1	89.1	4193.5	1588.8	29.0	218.8	44.6	2.3	363.3	728.3	20.1	32.7

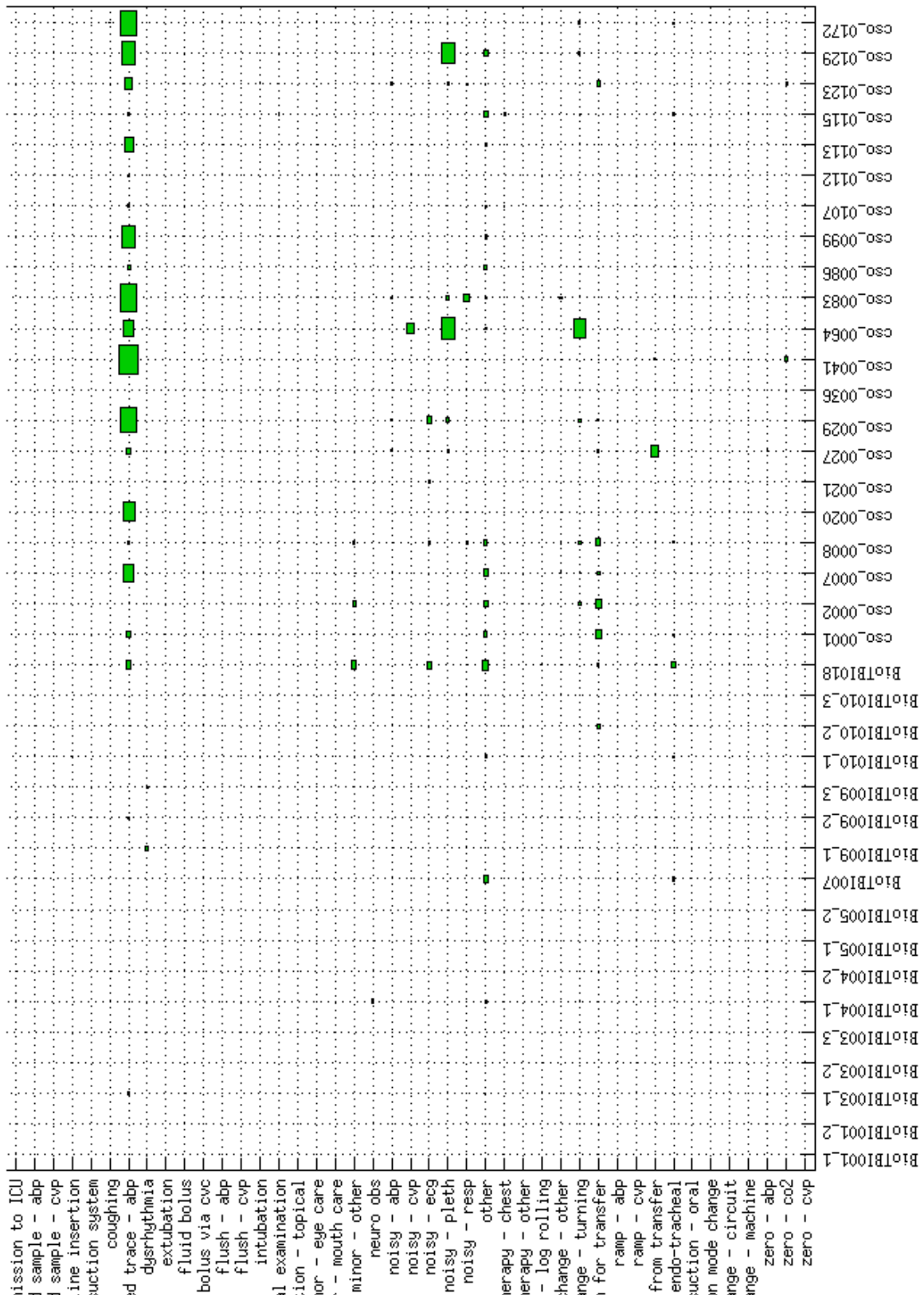

Figure 1: Graphical representations of annotation durations per patient. For each patient-annotation combination, the size of the rectangle indicates the fraction of recorded time that the particular annotation was present for that patient.

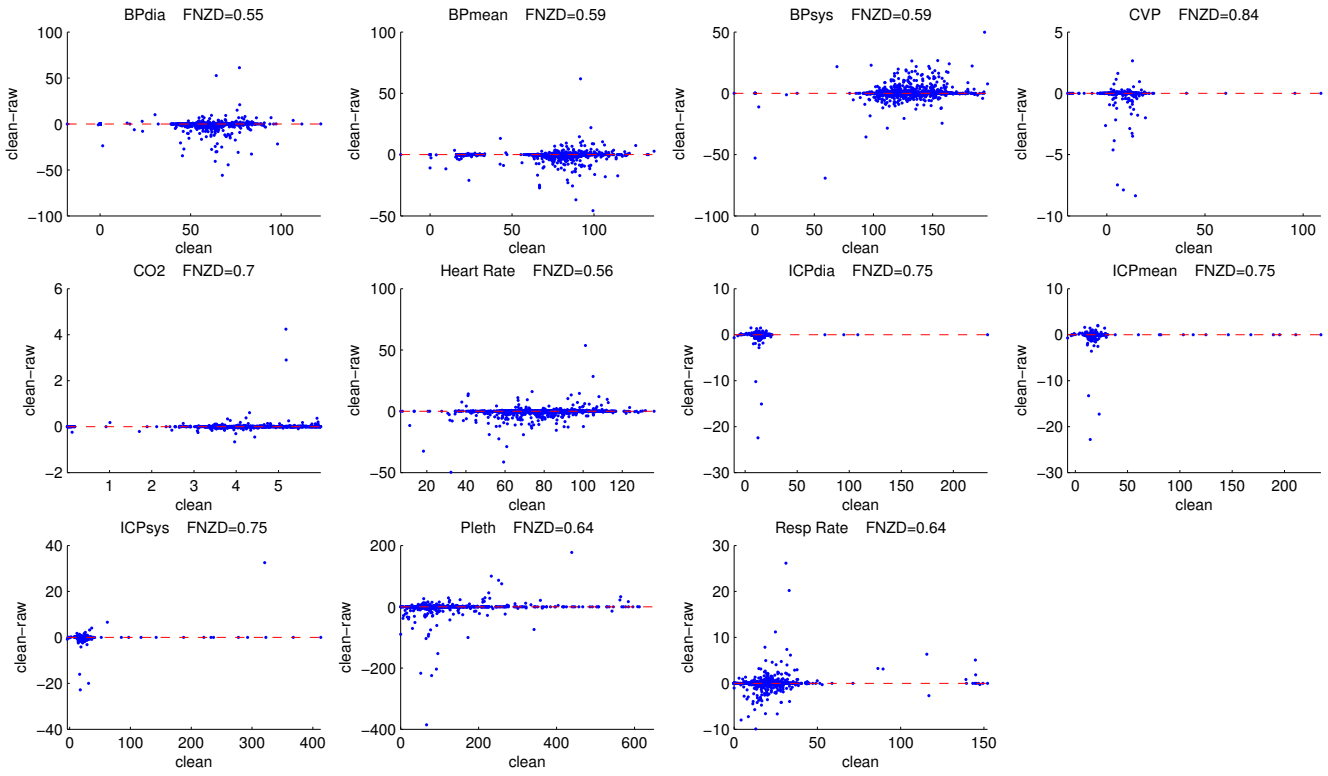

Figure 2: Bland-Altman plots for various channels computed over 30 minutes. The x-axis shows the clean value, and the y-axis the difference between the clean and raw values. FNZD denotes the fraction of non-zero differences.

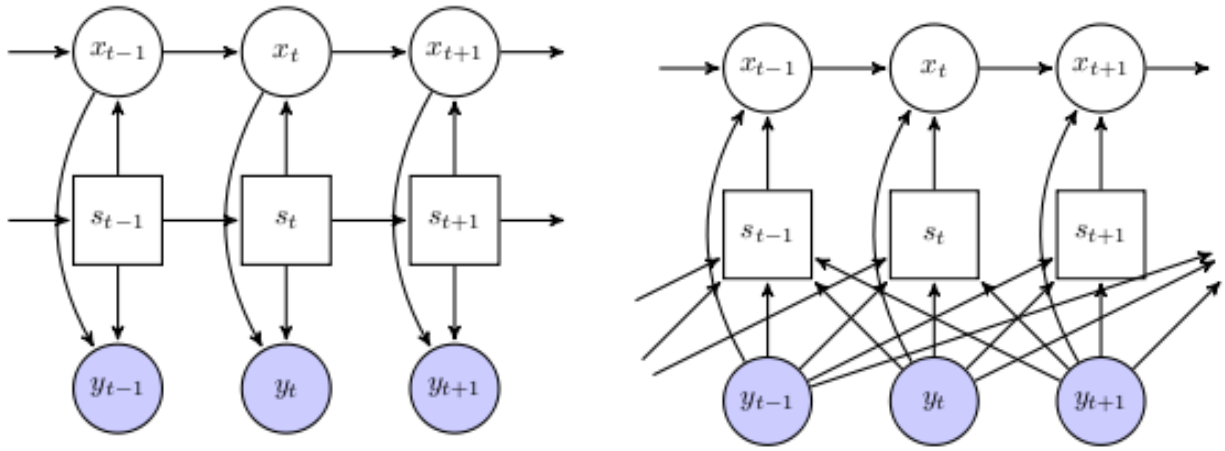

Figure 3: A graphical model representation of the FSLDS (left) and DSLDS (right).

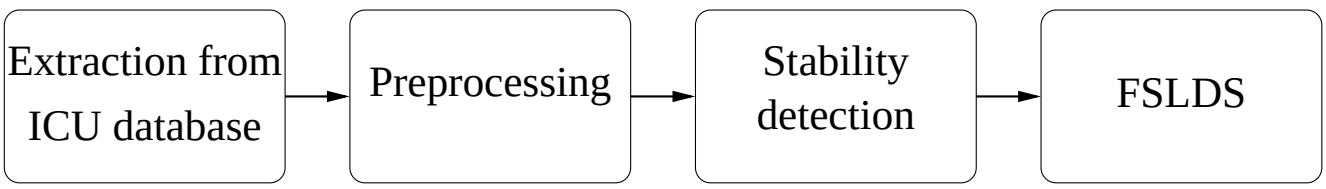

Figure 4: The realtime pipeline.

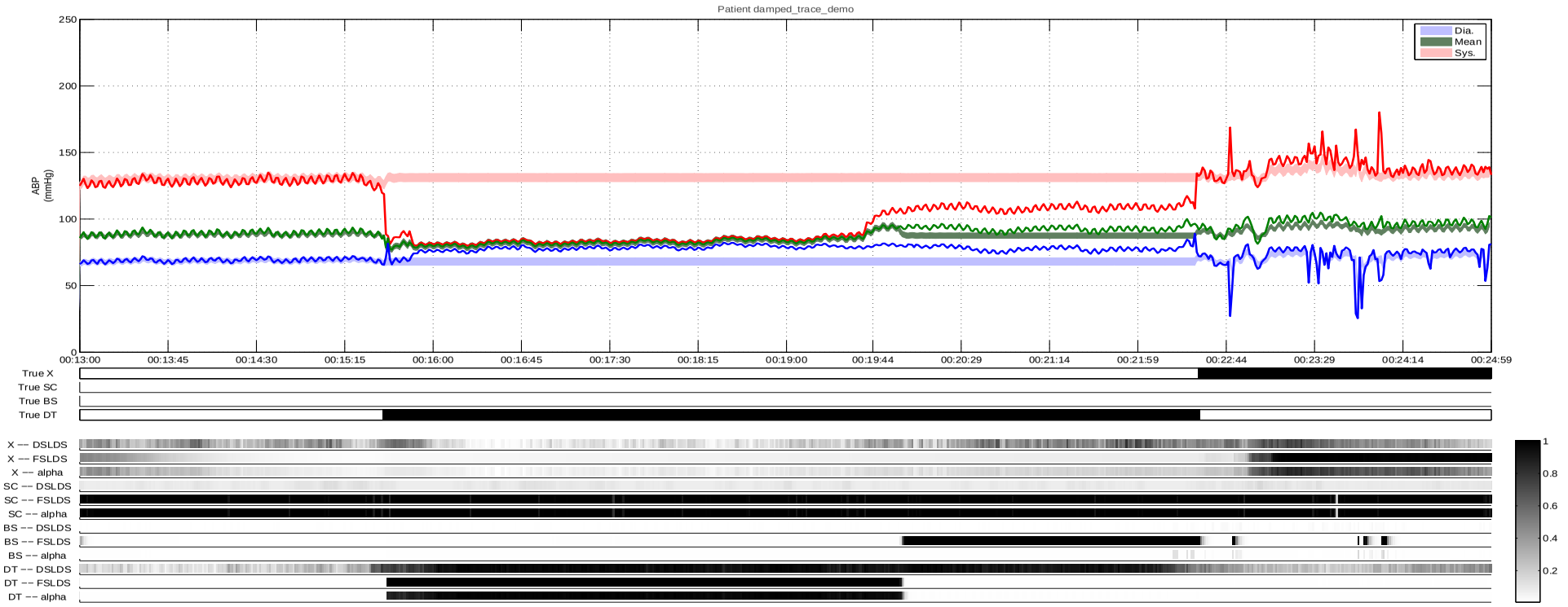

Figure 5: An example of FSLDS and DSLDS inferences for a damped trace event. Note that the active X-factor at the end of the plot is due to a “noisy ABP” annotation. The data is plotted in bright colour, with the FSLDS inferences shown as a lighter colour and with a one standard deviation confidence interval. For each factor the DSLDS, FSLDS and α-mixture inferences are shown. Posterior and ground truth probabilities are denoted by greyscale intensity as shown in the colour bar.

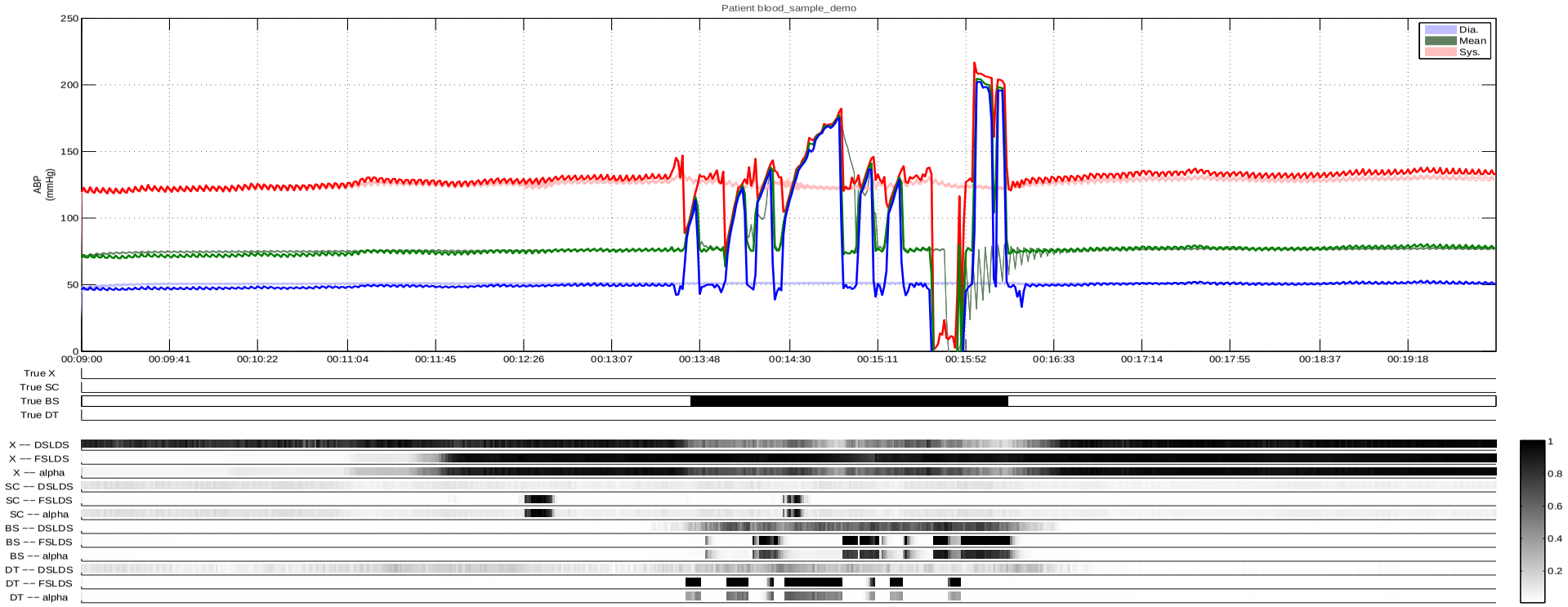

Figure 6: An example of FSLDS and DSLDS inferences for a blood sample event. The data is plotted in bright colour, with the FSLDS inferences shown as a lighter colour and with a one standard deviation confidence interval. For each factor the DSLDS, FSLDS and α-mixture inferences are shown. Posterior and ground truth probabilities are denoted by greyscale intensity as shown in the colour bar.

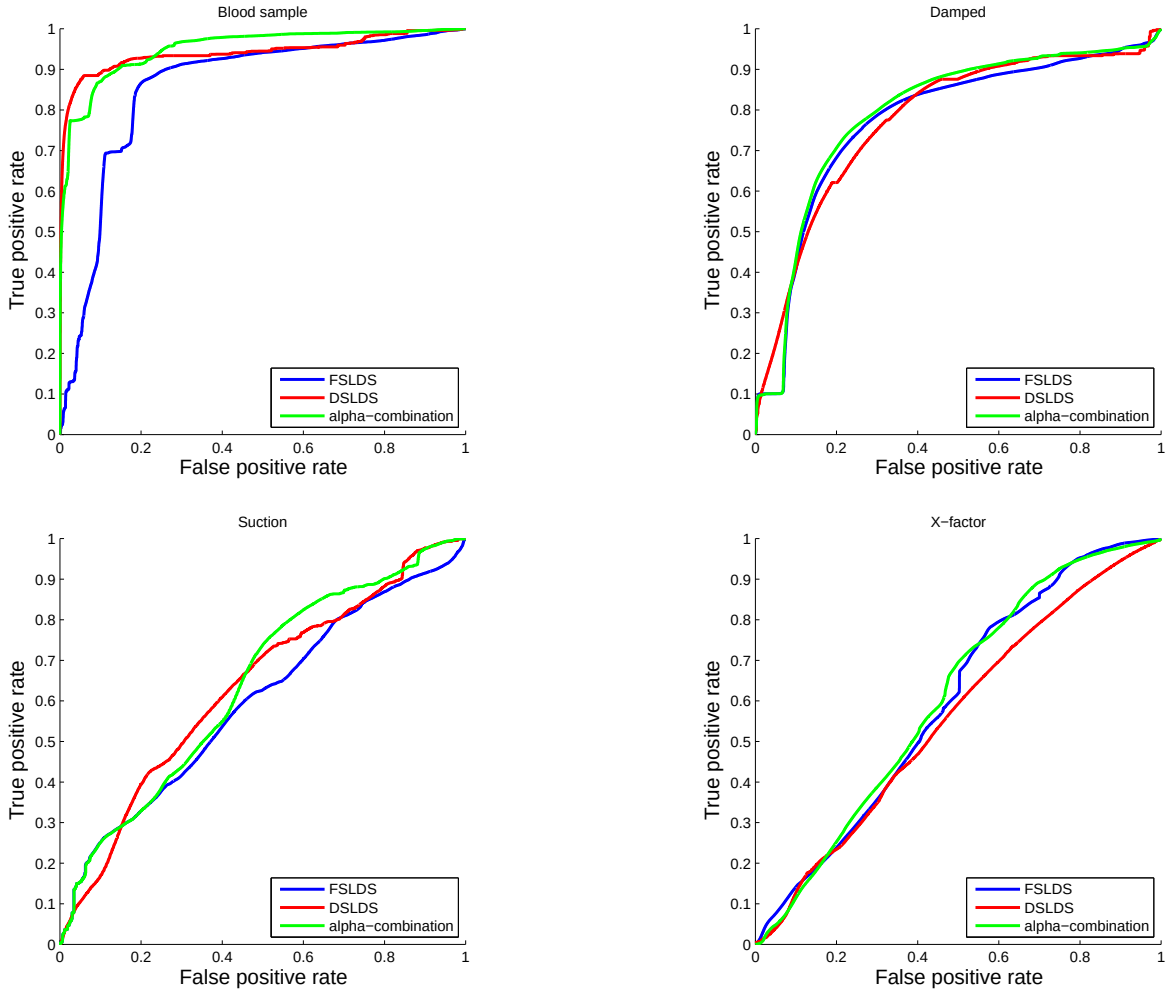

Figure 7: ROC curves for the blood sample, damped trace, suction and X factor computed from the FSLDS, DSLDS and α-mixture outputs.

AUC	BS	DT	SC	X
BioTBI001	0.90	0.99	0.74	0.64
BioTBI003	0.90	0.98	0.77	0.81
BioTBI004	0.73	0.83	0.65	0.61
BioTBI005	1.00	*NA*	0.81	0.44
BioTBI007	0.99	0.82	0.49	0.60
BioTBI009	0.69	0.21	0.28	0.25
BioTBI010	0.98	0.97	0.68	0.60
BioTBI018	0.96	0.90	0.54	0.61
CSO_0002	0.96	0.24	0.32	0.73
CSO_0007	0.47	0.89	0.47	0.57
CSO_0008	0.91	0.83	0.50	0.84
CSO_0020	0.84	0.70	0.53	0.74
CSO_0027	0.62	0.70	0.70	0.46
CSO_0029	0.69	0.90	0.51	0.80
CSO_0036	0.85	0.90	0.73	0.77
CSO_0041	0.95	0.64	0.58	0.35
CSO_0064	0.96	0.72	*NA*	0.33
CSO_0083	0.60	0.82	0.41	0.60
CSO_0099	0.98	0.87	0.57	0.75
CSO_0107	0.92	0.88	0.35	0.55
CSO_0112	0.43	0.80	0.41	0.65
CSO_0113	0.89	0.46	0.69	0.36
CSO_0115	0.97	0.71	0.13	0.33
CSO_0123	0.91	0.70	0.66	0.51
CSO_0129	0.91	0.96	0.77	0.39
CSO_0158	0.87	0.76	*NA*	0.58
CSO_0172	0.98	0.71	0.61	0.73
Total	0.86	0.77	0.60	0.60

Table 3: Table showing the AUC scores per factor per patient-interval for the FSLDS. NA indicates that the AUC score is not available because no events of the specified type occurred for the given patient.

AUC	BS	DT	SC	X
BioTBI001	0.97	0.95	0.68	0.37
BioTBI003	0.98	0.99	0.48	0.44
BioTBI004	1.00	0.96	0.59	0.65
BioTBI005	1.00	*NA*	0.85	0.36
BioTBI007	1.00	1.00	0.44	0.41
BioTBI009	1.00	0.90	0.49	0.22
BioTBI010	0.96	1.00	0.83	0.56
BioTBI018	0.98	0.83	0.64	0.45
CSO_0002	0.96	0.76	0.64	0.48
CSO_0007	0.95	0.55	0.38	0.87
CSO_0008	0.99	0.88	0.73	0.61
CSO_0020	0.94	0.75	0.61	0.69
CSO_0027	0.97	0.75	0.56	0.63
CSO_0029	0.98	0.47	0.59	0.53
CSO_0036	0.96	0.72	0.47	0.48
CSO_0041	0.98	0.41	0.53	0.48
CSO_0064	1.00	0.66	*NA*	0.57
CSO_0083	0.85	0.76	0.38	0.46
CSO_0099	0.97	0.87	0.60	0.62
CSO_0107	0.94	0.73	0.62	0.61
CSO_0112	0.92	0.71	0.14	0.60
CSO_0113	0.83	0.34	0.64	0.44
CSO_0115	0.94	0.67	0.84	0.44
CSO_0123	0.98	0.69	0.72	0.33
CSO_0129	0.94	0.86	0.68	0.57
CSO_0158	0.99	0.62	*NA*	0.75
CSO_0172	1.00	0.75	0.46	0.64
Total	0.94	0.78	0.64	0.56

Table 4: Table showing the AUC scores per factor per patient-interval for the DSLDS. NA indicates that the AUC score is not available because no events of the specified type occurred for the given patient.

AUC	BS	DT	SC	X
DSLDS	0.94	0.78	0.64	0.56
FSLDS	0.86	0.77	0.60	0.60
α-mixture	$0.95^{(0.9)}$	$0.79^{(0.9)}$	$0.64^{(-\infty)}$	$0.61^{(1.4)}$

Table 5: Table showing the over AUC scores per factor for the FSLDS, DSLDS and α-mixture. The optimal value of the α parameter per factor is shown inside parentheses.