Classification and feature extraction for neonatal sepsis detection

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

02-03-2022 / 03-03-2022

CITATION

Honoré, Antoine; Forsberg, David; jost, kerstin; Adolphson, Katja; Stålhammar, Alexander; Herlenius, Eric; et al. (2022): Classification and feature extraction for neonatal sepsis detection. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.19290257.v1>

DOI

[10.36227/techrxiv.19290257.v1](https://doi.org/10.36227/techrxiv.19290257.v1)

Classification and feature extraction for neonatal sepsis detection

Antoine Honoré*[†], *Student, IEEE*, David Forsberg[†], Kerstin Jost[†], Katja Adolphson[†], Alexander Mildalen Stålhammar[†], Eric Herlenius[†], Saikat Chatterjee* *Member, IEEE*

Abstract—Machine learning based early detection of neonatal sepsis - 24 hours before clear symptoms - is a challenging binary classification problem. In neonatal intensive care units (NICU), the challenge arises due to lack of facilities to collect and process multi-modal data, lack of consensus for feature extraction from data, presence of class imbalance in data, choice of detection algorithms, and absence of a standardized dataset for evaluation of algorithms. In this article, we provide a comprehensive study of traditional and contemporary binary classification algorithms, in a wide gamut of complexity. We consider two major types of classifiers - generative-model based algorithms using maximum-likelihood principle and data-driven model based algorithms using discriminative learning principle. Our cohort is composed of 93 very low birth weight ($< 1500\text{g}$) infants, among which 18 experienced sepsis. In total we analyze data for 1847 hospitalization days. In our study, we include logistic regression (LR), naive Bayes (NB), multi-layer perceptrons (MLP), Gaussian mixture models (GMM), and normalizing flows (NF). We first show that feature extraction using a larger window length than existing state-of-the-art choice is better, almost across all the classifiers. Secondly, we show that in absence of high-resolution data, simple classifiers such as LR and NB are highly competitive compared to the contemporary complex classifier NF. Finally, we show that it could be beneficial to collect high-resolution data for the complex NF classifier. We achieve the classification (detection) performance around 0.87 using the measure area-under-receiver-operating-characteristics (AUROC).

I. INTRODUCTION

The early detection of neonatal sepsis is a challenging task that, if improved, could reduce mortality and morbidity in neonatal intensive care unit (NICU) patients [1]. As an inadequate treatment puts the patient at risk of organ failures, potentially leading to death or long-term co-morbidity, many

clinicians instead over-treat with antibiotics as the diagnostic aids to detect sepsis are limited [2]–[4]. Bacterial growth in blood cultures is the golden standard for sepsis diagnosis, but is slow, invasive and has a high false negative rate ([5]–[7]). Instead, C-reactive protein (CRP) is often used as marker for general inflammation. CRP is rapidly analyzed (minutes), but slow biological kinetics and low specificity and sensitivity limits its usefulness [5]–[7].

The non-invasive monitoring of cardio-respiratory patterns might be helpful in the early detection of neonatal sepsis [8]–[10]. However, sepsis characteristic patterns of vital signs are often subtle and might go unnoticed by the medical staff with current care routines. Thus, the development of algorithms continuously analyzing monitoring data might help the clinical staff to both identify adverse patterns and start adequate treatment early.

a) *Data storage impacts research:* Bedside monitors are able to record data such as electrocardiograms with a sampling frequency as high as 500Hz. Other signals such as the pulse oximetry waveform and chest impedance to detect respiratory rate are sampled at frequencies of 62Hz and 125Hz, respectively. The large volume of data generated by the monitors make the detection of trends and patient assessment difficult to derive in daily clinical practice. The monitors often implement pre-processing, such as noise removal or QRS complex detection, which helps present relevant physiological information, such as heart rate, to the medical staff. The pre-processed signals are sampled at frequencies of at most 1Hz. This to facilitate interpretation, transmission and temporary storage of data. The retrospective retrieval of monitor signals is a requirement for daily care routines, allowing clinicians to observe dynamics and trends from the recent history of the patients. It is also useful for researchers to design and validate new predictive scores for diseases such as sepsis in large cohorts of patients. Large volumes of data which are both securely stored and accessible for research are costly. The sampling frequency of the signals is often a key parameter to reduce the storage computational and financial burden. Hospital-wide database systems often sub-sample the data to frequencies of around $\frac{1}{60}$ Hz (1 sample per minute) for long term storage, drastically reducing the data volume without knowledge/insight of the consequence on research.

b) *Extracting features from time series:* Neonatal sepsis detection (NSD) algorithms based on monitor data often resort to a binary classifier. The binary classifier is used to determine

This work was supported in part by grants from Digital Futures, KTH Royal Institute of Technology, the Swedish Research Council (2019-01157), the Stockholm County Council (20190400 and 2019-0974), the Swedish Brain Foundation (FO2019-0087), the Swedish National Heart and Lung Foundation (20180505), the Swiss National Science Foundation (P400PM-194474), the Axel Tielmans, and the Freemasons Children's House foundations. This study was approved by the Swedish Ethical Review Authority (2020-02487).

* Division of Information Science and Engineering, School of Electrical Engineering and Computer Science, KTH Royal Institute of technology, Stockholm, Sweden.

[†] Department of Women's and Children's Health, Karolinska Institutet, Stockholm, Sweden.

Correspondence: Antoine Honoré (honore@kth.se).

whether a time window of data presents "septic" patterns or not. Classification is performed after features have been extracted from fixed-duration time windows of monitoring data. Features extracted from time windows often require the signal to be stationary within that period. However, sensor artefacts inducing arbitrary patterns in the monitoring data might lead to non-stationary signals over the course of a too long time. In fact, windows longer than 1 minute fail to pass a stationary test [11]. This indicates that features computed on windows longer than 1 minute might fail to consistently capture "septic"-like dynamics in the signal. Regardless, some features for NSD have been designed on time frames much longer than 1 minute. Features such as sample entropy have been designed to quantify the predictability of inter beat interval (IBI) signals [12]. Such features have been shown useful, even when the time window duration is between 20 and 30 minutes, i.e. largely exceeding 1 min [10]. Sample entropy [12] and sample asymmetry [13], two features extracted from a time window of IBI signal, have been shown to help predict sepsis in very low birth weight infants. These features are used in a commercialized sepsis prediction algorithm named HeRO (Heart Rate Observation) [14]. Motion and respiration related time domain features in [15] and Fourier domain features in [16], [17] have also been shown useful for NSD. A feature extraction method which does not depend on a time window length parameter has been proposed in [18] and uses hidden Markov models (HMM) to detect clinical events relevant for sepsis prediction. This shows that feature-based algorithms are still relevant although the stationarity assumption is not fulfilled. The window length choice should be made with the time vs frequency resolution tradeoff. A better resolution in a certain domain leads to more accurate features in that domain. Shorter window length lead to better time resolution while longer window length lead to a better frequency resolution. To the best of the authors' knowledge, no comprehensive study has been performed to unravel the impact of time window duration on feature based sepsis detection algorithms.

c) *Choosing appropriate statistical models:* In feature-based NSD, statistical models are used to describe the relationship between input data, including features computed from vital signs time series, and the patient condition. The statistical models are a key component of the prediction algorithm, enabling an informed binary classification given new input data. The interpretability of prediction algorithms is often required by clinicians to motivate their actions based upon a classification result [19]. This often leads to the choice of linear prediction algorithms to build feature based classifiers [20]. Linear prediction algorithms can yield an importance score for each input feature which directly explains the classification result to clinicians [14], [15]. Such algorithms might nonetheless have a limited ability to leverage complex relations between the input data and the clinical condition to be detected.

Machine learning based non-linear statistical models emerged as a way to obtain better performances on a variety of prediction tasks such as machine vision [21] or machine translation [22]. These models are theoretically promising given their complexity and adaptability to various data sources. However, their abilities have not been extensively explored on medical domain tasks

such as NSD. It is unclear whether more powerful prediction algorithms, which are not readily interpretable, would perform better and be more suitable with similar input features for a NSD task. In fact, [23] reported that random forest (RF) and multilayer perceptron (MLP) do not perform better than logistic regression (LR) on a feature based NSD task. Models such as Convolutional neural networks (CNN) take advantage of local correlation in the input data have a limited amount of learnable parameters. CNN are however, not applicable here because the input data does not have local correlations. Kernel based Support Vector Machines (SVM) have been widely used for binary classification task. Their training algorithm is nonetheless intractable when the total number of samples becomes large. MLP, RF and SVM might underperform due to the lack of positive cases in the training data and subsequent class imbalance, inherent to NSD tasks.

Class imbalance is an issue where some classes are over-represented and other classes are under-represented in a training dataset. When not accounted for, the algorithms lean toward classifying all the new samples in the over-represented class, falsely reporting a very high accuracy on the training dataset. In our NSD datasets, we observed that the samples from the negative class (non-sepsis) are roughly 200 times more prevalent than samples in the positive class. This means that classifying all the samples as negative would give 99.5% accuracy. To account for class imbalance, discriminative models such as MLP or RF, are often used in a cost-sensitive learning framework [24]. This consists in assigning a well chosen penalty on the misclassification of samples in the under-represented class during training. This method might lead to sub-optimal engineering choices. Generative models take class imbalance into account directly in the statistical model. Moreover, a Naïve Bayes classifier (a generative model) has been shown to reach it's maximal performances with fewer sample points than the logistic regression algorithm (a discriminative model) [25]. This makes the use of generative models promising in our scenario. Recently, generative models based on deep neural networks called normalizing flows have been proposed [26], [27]. Normalizing flows were used by [28], [29] as emission probability densities for hidden markov models in time series classification tasks including NSD. These models combine the potential benefits of being generative models and of being neural networks based, i.e. with potentially more modeling capacities than classical generative models such as Naïve Bayes classifiers or Gaussian Mixture models (GMM).

A. Contributions

The contributions of the study can be summarized as follows:

- 1) We compared the performances of five classes of algorithms for the sepsis detection task: logistic regression, Extreme learning machines, Gaussian Mixture model, Multilayer Perceptron, Normalizing flow models.
- 2) We studied the impact of the time window length on the performances of detection algorithms.
- 3) We evaluated the effective consequences of increasing the sampling period of signals on the performances of NSD algorithms.

- 4) We compared the performances of normalizing flow models when trained generatively and discriminatively. We also discuss the impact of the model sizes and of the strength of l_1 -regularization on the normalizing flow performances.

II. METHODS

A. Study population

We performed a longitudinal cohort study on a population of very low birth weight ($< 1500\text{g}$) infants, hospitalized in the Neonatal Intensive Care Unit (NICU) at Karolinska University Hospital, Stockholm, Sweden. For this manuscript, we studied a subgroup of 93 infants, with a median gestational age of $27 (\pm 3)$ weeks. Among the included infants, 45% were male, 55% were female, 18 experienced sepsis. Sepsis was defined as culture verified early onset sepsis, culture verified late onset sepsis and culture negative sepsis with more than 5 days of antibiotic treatment. In total we analyzed 1847 days of hospitalization. This study was approved by the Swedish Ethical Review Authority (2020-02487)

B. Input data

For each patient, we used prospectively collected high frequency data from Philips IntelliVue MX800 Patient Monitor (Philips Healthcare, Amsterdam, Netherlands). We retrospectively built an event timeline to identify the time of sepsis suspicion from the Electronic Health Records of the patients. We used the pre-processed monitor data sampled at 1Hz as the basis signals. The signals we used were the electrocardiogram-derived IBI signal, the level of arterial oxygen saturation measured by pulse oxymetry (SpO_2) and the respiratory frequency derived from chest impedance.

C. Algorithms

1) *Pre-processing*: The signals are sub-sampled to the desired sampling periods in seconds.

Reducing the sampling frequency of signals can be done for instance by keeping one sample during each sampling period, and discard all the others. This approach discards a lot of potentially useful information carried by the monitoring signals during the sampling period. Here we chose to sub-sample by averaging the signal over the sampling period. This means for instance that for a sampling period of 10 seconds, we keep the average of the signals over a 10 seconds window, rather than keeping only the first or last sample. A sampling period of 1 second corresponds to the highest frequency which can be obtained for the parameter data from the monitors.

For the sub-sampled data of sampling period 1 and 3 seconds, the missing data intervals smaller than 15 samples were linearly interpolated. For the sub-sampled data of sampling period 10 seconds and more, only the missing data intervals of 1 sample were linearly interpolated. The interpolated and sub-sampled data were then smoothed with a moving average filter of width 3 time samples.

2) *Filtering*: The IBI signal was further filtered due to the use of the more sensitive sample entropy feature. We intended to remove ectopic beats and strong non-physiological frequency content [30]. For this we used a composition of a moving median filter of width 3 samples and a Butterworth filter of order 6 with normalized low cut frequency of 0.0021 and high cut frequency 0.43 for the non-physiological trends removal [31]. This filtering strategy has the limitation of requiring the input time window to be at least 39 samples long.

3) *Feature extraction and labeling*: We built overlapping time windows of fixed duration. For instance, the 45 minutes windows are 45 samples long for a signal sub-sampled with a sampling period of 60 seconds, and 90 samples long for a signal sub-sampled with a sampling period of 30 seconds. A time window is labeled 1 if it starts up to 24h ahead of a suspicion of sepsis as described in the patients EHR and 0 otherwise. For each individual time window, we computed the minimum, maximum, mean, standard deviation, kurtosis and skewness of each signal. We also computed the sample entropy and the sample asymmetry of the IBI signal. This resulted in 20 features computed for each window. The time windows containing signals with missing data were discarded. We also included repetitive body weight measurements and postnatal age of the patients. Additionally, we included background patient information as static features: the birth weight and the sex.

For each time window, this resulted in 24 features in total, i.e. samples of dimension $d = 24$.

4) *Datasets*: The pre-processing and feature extraction steps described above lead to a list of features for each patient time window. In the rest of the text we call a list of features, a sample.

The collection of all the lists of features (the collection of all the samples) and their respective labels, constitute the training dataset. In our context, the training datasets were imbalanced. This means that there were a lot more samples labeled 0 than samples labeled 1. The prevalence of positive samples varied from 0.40% to 0.50% depending on window length and sampling period (Suppl. table IV). As was explained in II-C.2, time windows must be at least 39 samples long. The datasets corresponding to window length/sampling frequency pair leading to time window of less than 39 samples were removed.

5) *Binary classification*: For each time window, the feature extraction steps resulted in 24 features in total, i.e. samples of dimension $d = 24$, representing the patient state. Binary classification is a problem where input samples $\mathbf{x} \in \mathbb{R}^d$ (read: $\mathbf{x}$ being a list of 24 decimal numbers) are to be assigned a binary label y , either 1 or 0. This problem can be decomposed into three main sub-problems. (1) modeling the posterior probability $p(y = 1 | \mathbf{x})$ of label 1 given input data samples $\mathbf{x} \in \mathbb{R}^d$, i.e. deciding on a statistical model which represents accurately the dependency between x and its label. (2) evaluating the posterior probability for a sample $\mathbf{x}$, i.e. using our model on a sample to obtain a probability prediction from the statistical model. (3) classifying the sample with a decision function, i.e. deciding upon assigning label 1 or 0 based on the statistical model prediction.

The modeling step (1) typically consists in two subparts. First a parametric statistical model is chosen, then the parameters of the models are optimized so that the likelihood of a training dataset under the model is maximized. The final classification is then performed for example with a threshold function: $p(y = 1 | \mathbf{x}) > \frac{1}{2}$ to classify the sample $\mathbf{x}$ as positive, i.e. if the posterior probability of label 1 for sample $\mathbf{x}$ is larger than $\frac{1}{2}$.

We compare five classification algorithms for the a posteriori (or posterior) probability: logistic regression (LR), extreme learning machine (ELM), multilayer perceptron (MLP), Naïve Bayes (NB), Gaussian mixture models (GMM) and normalizing flow (NF) models. LR, ELM and MLP are data-driven model based algorithms which are trained using discriminative principle. They can handle class imbalance using suitable cost functions at the time of training and we implemented that in our experiments. NB, GMM and NF are generative-model based methods which use maximum-likelihood based model training. For NF, we also did a discriminative learning based training.

6) Binary classification algorithms: We describe the statistical models we used to represent the posterior probability $p(y = 1 | \mathbf{x})$, i.e. modeling the dependencies between $\mathbf{x}$ and its label y . A summary is available in table IV-A.

a) Logistic regression (LR): Logistic regression (LR) is a model that is often used for neonatal sepsis prediction algorithms [20]. LR models the relationship between $\mathbf{x}$ and its label using a sigmoid function and a weighted sum of the components in $\mathbf{x}$. This makes LR an easily interpretable linear algorithm. In a logistic regression model parameterized by a vector $\beta \in \mathbb{R}^d$, the posterior probability that the label of a given data sample $\mathbf{x}$ is positive is expressed as:

$$p(y = 1 | \mathbf{x}) = \sigma(\beta^T \mathbf{x}) = \frac{1}{1 + e^{-\beta^T \mathbf{x}}}. \quad (1)$$

b) Extreme Learning Machine: Extreme Learning Machine (ELM) [32] generalize LR models with the addition of a one hidden layer feed forward neural network. In an ELM, only the weights of the output layer are trained. The weights of the input layer are generated at random and are kept fixed. This produces a rich set of random features without adding significant computational complexity at training time. ELMs can then theoretically use the rich set of features to obtain better classification performances than using the input features directly (i.e. the logistic regression algorithm). In an ELM with $n \in \mathbb{N}$ hidden neurons and parameterized by a vector $\beta \in \mathbb{R}^n$, the posterior probability that a sample $\mathbf{x}$ is positive is expressed as:

$$p(y = 1 | \mathbf{x}) = \sigma(\beta^T a(R\mathbf{x})) = \frac{1}{1 + e^{-\beta^T a(R\mathbf{x})}}, \quad (2)$$

where $R \in \mathbb{R}^{n \times d}$ is a fixed matrix with components drawn from a normal distribution. $a : \mathbf{z} \mapsto \max(\mathbf{z}, 0)$ is the rectified linear unit (ReLU) activation function.

c) Multilayer perceptron (MLP): Multilayer perceptron (MLP) is a feed forward neural network with a fully connected architecture. As opposed to ELM with one hidden layer and untrained input weights, MLP has several fully connected hidden layers (i.e. the whole set of features at one hidden

layer is used to compute the set of features of the next hidden layer) and all the weights are trained. In a MLP composed of L layers, each with n hidden nodes, the posterior probability of y given $\mathbf{x}$ is expressed as:

$$p(y = 1 | \mathbf{x}) = \sigma(\theta_L^T f_L(f_{L-1}(\dots f_1(\mathbf{x}; \theta_0) \dots; \theta_{L-2}); \theta_{L-1})), \quad (3)$$

where $\theta_i \in \mathbb{R}^{n \times n}$ are the weights at layer $i \in \{1, \dots, L-1\}$, $\theta_0 \in \mathbb{R}^{d \times n}$ are the weights at the input layer, $\theta_L \in \mathbb{R}^n$ are the weights of the output layer. f_i is the mapping at layer i defined as $f_i(\mathbf{h}_{i-1}; \theta_{i-1}) = a(\theta_{i-1}^T \mathbf{h}_{i-1})$, with $a : \mathbf{z} \mapsto \max(\mathbf{z}, 0)$ the ReLU activation function, and $\mathbf{h}_i$ the input at layer i (i.e. the output of layer $i-1$) and $\mathbf{h}_0 = \mathbf{x}$.

d) Gaussian mixture models (GMM): A Gaussian mixture model (GMM) is used to model the likelihood function of each class. The samples labeled "1" (class 1 samples) and the samples labeled "0" (class 0 samples) are used independently to train two statistical models. The likelihood of an input sample $\mathbf{x} \in \mathbb{R}^d$ with respect to the GMM of a class c is expressed:

$$p(\mathbf{x} | y = c; \theta^c) = \sum_{i=1}^d w_i^c \mathcal{N}(\mathbf{x}; \mu_i^c, \Sigma_i^c), \quad (4)$$

where $c \in \{0, 1\}$, $\theta^c = \{(w_i^c), (\mu_i^c), (\Sigma_i^c)\}$ is the set of parameters for the Gaussian mixture model. $w_i^c \in \mathbb{R}^+$ are the positive mixing parameters which sum to 1, $\mu_i^c \in \mathbb{R}^d$ are the mean vectors, $\Sigma_i^c \in \mathbb{R}^{d \times d}$ the covariance matrices. $\mathcal{N}(\mathbf{x}; \mu_i^c, \Sigma_i^c)$ is the density function of a multivariate Gaussian distribution with mean vector μ_i^c and covariance matrix Σ_i^c . The covariance matrices of GMMs are often chosen diagonal for classification problems. For input features of dimension 24, diagonal covariance matrices reduce the number of parameters of a Gaussian distribution from 650 to 50 and help prevent overfitting. A mixture model with 1 diagonal Gaussian distribution is identical to the statistical model of the Naïve Bayes algorithm.

The posterior probability for class c is then expressed using Bayes rules:

$$p(y = c | \mathbf{x}) = \frac{p(\mathbf{x} | y = c)p(y = c)}{\sum_{j=1}^2 p(\mathbf{x} | y = j)p(y = j)}, \quad (5)$$

where $p(y = c)$ is class c prior probability, i.e. the probability of drawing a sample with class c from the training dataset.

For Gaussian mixture models, we use a generative learning approach, i.e. the models for each class are learnt independently. In other words, the GMM of each class are independent and not optimized to reject samples from other classes. This allows new classes to be added to the model by simply training new GMMs.

Once the models are learnt, the decision function we use to perform the binary classification is the maximum a posteriori rule:

$$\hat{c} = \arg \max_j p(\mathbf{x} | y = j)p(y = j), \quad (6)$$

where $\hat{c} \in \{0, 1\}$ is the predicted class for a given input sample $\mathbf{x}$.

In a maximum a posteriori classifier, the class imbalance is taken into account explicitly in the statistical model with the use

of class prior probability. In the logistic regression model the class imbalance is taken into account with a heuristic approach such as cost-sensitive learning [24]. This makes the use of maximum a posteriori classifiers with generatively learned statistical models more theoretically sound for data imbalanced problems.

e) Normalizing flows (NF): Normalizing flows are statistical models which can be trained generatively and used in a maximum a posteriori classifier. They offer more modeling capacities than GMM and offer more state of the art training algorithms and regularization methods.

We use a particular normalizing flow (NF) model called RealNVP [26] in our study. The RealNVP model uses a non-linear, neural network-based diffeomorphism (differentiable bijective map with a differentiable invert) to map the input data distribution to a fixed multivariate Normal distribution. The design of the transform makes the computation of the data likelihood tractable with the use of the change of variable formula.

Suppose that a latent variable $\mathbf{z} \in \mathbb{R}^d$ is sampled from a multivariate Normal distribution denoted $p_0(\mathbf{z})$. Suppose also that g_θ , parameterized by θ , is a diffeomorphism between a latent space and the input space. Then, using g_θ and $\mathbf{z}$, we can generate samples $\mathbf{x}$ in the input space: $\mathbf{x} = g_\theta(\mathbf{z})$. Using the inverse function, $f_\theta = g_\theta^{-1}$, we can normalize samples $\mathbf{x}$ in the latent space: $\mathbf{z} = f_\theta(\mathbf{x})$. The probability of a sample in the input space is thus computed according to the change of variable formula by evaluating (1) the probability of the normalized sample $f_\theta(\mathbf{x})$ in the latent space and (2) the Jacobian of the normalization transformation:

$$\log p(\mathbf{x}) = \log p_0(\mathbf{z}) + \log |\det \nabla g_\theta|^{-1}, \quad (7)$$

where $\nabla g_\theta = \frac{\partial g_\theta}{\partial \mathbf{z}}$ is the Jacobian of g_θ , and $p_0(\mathbf{z})$ is the density function of a multivariate Normal.

Similarly to the GMM case, we model each class likelihood function with one such normalizing flow.

We propose two training strategies for the normalizing flow models. The first strategy is similar to that used for the GMMs, i.e. the normalizing flow models are trained independently and the classification is performed using the maximum a posteriori decision rule (See equation (6)). The second approach consists in learning the class distributions jointly by maximizing the true (i.e. normalized) posterior probability distribution with respect to the model parameters. In this framework, the two class normalizing flow models are trained jointly using back-propagation on the negative posterior probability defined in (5).

In both frameworks, we use l_1 -regularization imposed on the loss defined in equation (5) to constrain the neural networks layers to sparse transformations and reduce overfitting.

The generative training of the normalizing flow models lead to the following loss function for each class j :

$$\text{loss}^j(\mathbf{x}) = -p(\mathbf{x}|y = j; \theta_j) + \alpha \|\theta_j\|_1. \quad (8)$$

The joint training of both classes lead to the following loss

function:

$$\text{loss}(\mathbf{x}, c) = -\frac{p(\mathbf{x}|y = c; \theta_c)p(y = c)}{\sum_{j=1}^2 p(\mathbf{x}|y = j; \theta_j)p(y = j)} + \alpha \sum_{j=1}^2 \|\theta_j\|_1, \quad (9)$$

where $c \in \{0, 1\}$ is the true label of a data sample $\mathbf{x}$, θ_j is the set of parameters for the realNVP model of class j , α controls the strength of the regularization. The loss is optimized for the normalizing flows of both classes jointly with mini-batch back-propagation. The computation of equation (9) requires the evaluation of the likelihood of both classes for all the samples which is rapidly too computationally expensive when the number of classes exceeds 2.

III. EXPERIMENTS

A. Evaluation of performances

The evaluation of the generalization ability of the classification algorithms consists in evaluating the classification performances on a patient population that has not been used during algorithm training. The datasets shown in supplementary table IV are split at the patient level into a training set and a testing set. We prevented samples from the same patient to be used both for training and testing by splitting at the patient level rather than at the samples level. We performed 20 random splits where 70% of the patients were used as training and 30% are used as testing.

Given our heterogeneous and limited patient population, we refrained from further splitting the training dataset to perform cross-validation. A fundamental assumption for cross-validation to be valid is that the performance metrics computed on the validation set are proxies for the same performance metrics computed on a test set. It translates in our context to the validation and test sets to be sufficiently similar in terms of patient background factors. The sufficient similarity is not guaranteed when the number of patients is limited. This in turn might lead to hyper-parameters optimized on a validation set not relevant for the test set. Instead, the hyper-parameters for the most complex Multilayer perceptron and normalizing flow models were optimized in a reference feature extraction case. They were kept fixed when the feature extraction parameters were varied (see section III-C).

B. Performance metrics

We report our results as median and inter-quartile range (IQR) values of performance metrics computed on the test sets. The main score we used was the AUROC score [33] that measures the discriminative ability of the algorithms for various decision thresholds. The AUROC is the probability that the risk prediction scores related to a true negative sample and a true positive sample will be ordered correctly. A high AUROC value indicates a high discriminative performance. Low IQR indicates small variations of the AUROC score when the training and testing sets are varied.

Other threshold dependent metrics are often used to measure the abilities of classification algorithms against type-I and type-II errors. These metrics are computed from the confusion matrix, i.e. from 4 classification scores: (TP) the number of

true positive, i.e. the number of samples correctly classified as positive; (FP) the number of false positives, i.e. the number of samples wrongly classified as positives, and symmetrically, (TN) the number of true and (FN) false negatives. Here we consider the specificity: $\frac{TN}{TN+FP}$, the sensitivity $\frac{TP}{TP+FN}$ and the balanced accuracy $\frac{\text{sensitivity}+\text{specificity}}{2}$ [34]. To compute these, a threshold must be chosen to make sure that the binary decision maximizes the sensitivity and specificity metrics. The optimal threshold might not be the same across neither algorithms, nor population splits. Thus, adjusting the decision threshold after every algorithm training raises the concern of the choice of the threshold when the algorithm is to be used in practice. Ideally, one optimal threshold would be determined after several runs based upon a validation set performance. As discussed in section III-A, our limited dataset does not allow such an approach. Therefore, we chose to set the threshold on the posterior probabilities to 0.5 for all the algorithms. This allows for a fair comparison between algorithms and prevents training set or testing set over-fitting.

AUROC measures the discriminative ability of prediction algorithms without the need for a threshold choice. This threshold agnostic metric is thus better suited for our case. We present all the metrics for the results of the reference feature extraction method case. The rest of the results are presented and discussed in terms of AUROC only.

C. Model training

We evaluated the performances of Naïve Bayes classifier, Logistic regression, Diagonal GMM, Full covariance GMM, ELM, MLP and normalizing flow models, both trained generatively and discriminatively.

LR models and ELM are trained with the limited memory Broyden–Fletcher–Goldfarb–Shanno (BGFS) algorithm [35]. We tried ELM with 200, 500 and 1000 hidden nodes. The models are denoted ELM(200), ELM(500) and ELM(1000).

The MLP models are trained using the ADAM [36] optimizer for mini-batch gradient descent on the cross-entropy loss function [37]. We trained 4 layers MLP with 8 and 32 hidden neurons (HN). The models are denoted MLP(8) and MLP(32) respectively. Given the dimension of our input data ($d = 24$), 8 hidden neurons corresponds to a highly compressed regime, while 32 hidden neurons corresponds to a network with more modeling capacities. Intermediate architectures with changing hidden size across layers can be obtained automatically with $l1$ -regularization where some connections might be assigned a 0 weight at training time.

For MLP, ELM and LR, the class imbalance was taken into account using cost sensitive learning [24]. More specifically, we added a large penalty to the loss of false negative predictions. The penalty weights on the loss for a sample of class c were computed as $\frac{n}{2n_c}$, where n_c is the number of samples in class c , and $n = n_1 + n_2$ is the total number of samples in the training dataset.

The GMMs were trained generatively (i.e. independently) for each class. The diagonal GMMs were chosen with 1, 2, 5 and 10 Gaussian distributions in the mixture model. We considered the Naïve Bayes (NB) classifier as being a GMM

with 1 Gaussian with diagonal covariance. We also tested a full covariance GMM with 1 Gaussian. The optimal parameters for all GMMs are learnt using the Expectation Maximization algorithm [38].

We tested all the algorithms described above with signals sampled with a sampling period of 1, 3, 10, 20, 30, 60 seconds. We also evaluated the performances with features computed from time windows of duration 5, 15, 25, $\dots$, 165 minutes with 50% overlap. We considered feature extraction with a window length of 25 minutes and a sample period of 1 second as the reference case, often used in the literature [10], [12], [15].

The normalizing flow models were trained both generatively (Eq (8)) and discriminatively (Eq (9)). In the reference feature extraction procedure, we tested normalizing flow models, with neural networks composed of 4 layers with hidden sizes 8 and 32 (denoted NF(8) and NF(32)), $l1$ -regularization parameter $\alpha = 0, 2e-5, 2e-4, 2e-3, 2e-2, 2e-1, 2e0$ and trained with a number of epochs $n_epochs=10, 20, 30$. Given the prohibitively long training time of normalizing flow models, we performed the rest of the experiments with models trained discriminatively, $\alpha = 200$ and $n_epochs=20$ for both NF(32) and NF(8), after experimenting on data sampled at 1Hz and a window size of 25 minutes (See table VII). The normalizing flow models were trained using mini-batch back-propagation on the losses defined in (Eq (8)) and (Eq (9)) with the ADAM algorithm [36] and a batch size of 128.

The experiments were carried out in Python (Python 3.7.3 [39]) using the scikit-learn library [40] for LR, GMMs and ELM. We used a Pytorch [41] implementation of the RealNVP normalizing flow model for our classifier. The experiments were carried out on a compute server running $2 \times \text{Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz}$ and $4 \times \text{Tesla V100-SXM2-16GB}$.

IV. RESULTS

A. Summary of models

We first show a summary of the models, their number of parameters and the training time in reference conditions (table I).

B. Reference conditions

The classification results are shown with features computed in reference conditions, i.e. from signals sampled at 1Hz, with a time window of 25 minutes and 50% overlap (table II). The results are reported in terms of median scores over 20 training/testing runs. The scores presented here are AUROC, sensitivity, specificity and balanced accuracy.

The best performing algorithm in this condition in terms of median AUROC was the normalizing flow model with neural networks of hidden size 32. The next 5 best performing algorithms LR, MLP(8), NF(8), NB and GMM(2) were within the IQR of the best performing algorithm, indicating a gain in using NF(32) that is only marginal. The algorithm with the smallest IQR among the top 6 performing algorithms is the MLP(8) with an IQR of 0.08, and a median AUROC of 0.8.

The lowest performing algorithms were ELM, GMMs and MLP(32). The highest performing ELM had 200 hidden neurons and a median AUROC of 0.62. The best performing

TABLE I: Details of the algorithms: number of trainable parameters, time required to fit in reference conditions, training scheme.

	Nb. params.	Fit time (s)
LR	25	11.38 (2.12)
NB	102	0.44 (0.32)
ELM(200)	200	9.5 (3.86)
GMM(2)	204	1.48 (0.6)
MLP(8)	425	9.33 (2.52)
ELM(500)	500	19.24 (7.91)
GMM(5)	510	5.23 (1.7)
GMM(1)	650	0.35 (0.05)
ELM(1000)	1000	26.45 (11.2)
GMM(10)	1020	11.24 (5.29)
MLP(32)	4001	210.39 (21.28)
NF(8)	13440	1210.38 (202.17)
NF(32)	76416	1895.61 (280.89)

Gen: Generative training, Disc: Discriminative training, Diag: Diagonal, Cov: Covariance, LR: Logistic regression, NB: Naïve Bayes, ELM: Extreme Learning Machine, MLP: Multi-layer Perceptron, GMM: Gaussian Mixture Model, NF: Normalizing flow model.

GMM had 2 diagonal Gaussians. Adding more Gaussians helped increase the training performances from 0.89 with 2 Gaussians to 0.93 and 0.98 with 5 and 10 Gaussians respectively, while decreasing the test performances to 0.6. This means that the GMM overfitted with more than 2 Diagonal Gaussians. GMM with 1 full covariance Gaussian overfitted with training performances reaching 0.96 and test performances as low as 0.55. This is our lowest performances, also reached by ELM(1000) and ELM(500). MLP(32) also poorly performed, with a median AUROC of 0.58.

NF(32) performed poorly according to the other metrics. The reported bAcc, sensitivity and specificity were 0.5, 0, 1, while AUROC was 0.82. We explain the discrepancy in the scoring by the fact that bAcc, sensitivity and specificity are threshold dependent metrics. The discrepancy between sensitivity and AUROC indicates that the model is poorly calibrated. In the rest of the experiments, we present and discuss only the AUROC score (see section III-B).

C. Variation in window length

We then evaluated the models while fixing the sampling period to 1 second and varying the window length (figure 1 and suppl table V). The best performances in terms of median AUROC is 0.86 and is achieved by NF(8) and NF(32), NB and MLP(8). Among these, the most reliable prediction is obtained by the MLP(8) algorithm with an IQR of 0.08, whereas NB and the normalizing flows have an IQR ≥ 0.11 . Varying the window length was particularly beneficial for NB with an increase in performances of +14.27% compared to the reference case discussed in IV-B. MLP(8), NF(32) and NF(8) have a marginal increase in median AUROC performances with +7.23%, +5% and +8% respectively.

The next best performing algorithm is LR with a median AUROC of 0.83 and a small marginal increase compared to the reference conditions of +3.86%. The IQR is moreover large (0.17) compared to that of MLP(8) (0.08).

GMM(2) and GMM(5) are outperformed by LR, with a median AUROC of 0.78 and 0.73 respectively. Varying the window length was largely beneficial to GMM(5) with an increase of +18.23% compared to the reference conditions case.

ELMs perform poorly with median AUROC of 0.68, and 0.65 for ELM(200), ELM(500) and ELM(1000) respectively. This is the case even though the change in window length was the most beneficial to ELM(1000) with an increase in performances of +19.17% compared to the reference conditions.

Overall, the algorithms benefited from an increase in window length for features computation when the sampling period was fixed to 1 second. GMM(5) however benefited from a decrease to 5 minutes, and GMM(10) did not benefit from changing the window length either way. The normalizing flows reached their best performances for a window length increased to 75 minutes and 115 minutes for NF(8) and NF(32), respectively. NB shows a large increase in performances relatively to the increase in window length. The performances increase by +14.27% in AUROC with a change of window length from 25 to 35 minutes. The rest of the algorithms benefit from increasing the window length from 25 to at least 75 minutes. ELM(1000), ELM(500), GMM Full Cov are best performing with 75 minutes windows; MLP(8) with 85 minutes; LR with 95 minutes; ELM(200) 105 minutes; MLP(32) and NF(32) 115 minutes; GMM(2) with 145 minutes. The algorithms attain their optimal performances for window lengths under 145 minutes. Figure 1 indicates that overall, the performances of the models decrease when the window length is increased over 145 minutes.

D. Optimal conditions per algorithms

We report the best results per algorithm when both the sampling period and the window length are varied (table III). The best performing algorithms were LR and NB with a median AUROC of 0.87, and NF(8), NF(32) and MLP(8) with nearly identical performances of 0.86. The lowest IQR were obtained with NB and NF(32) with 0.16. The largest IQR was NF(8) with 0.29. NB was the algorithm which benefited the most from varying the parameters with an increase in performances of +15.58%, then LR with an increase in performances of +9.65%. MLP(8), NF(8) and NF(32) still benefited from varying the parameters with +7.95%, +8% and +5% respectively.

LR, NB and MLP(8) benefited from slightly increasing the sampling period from 1 second to 3 seconds. The optimal window length was affected for NB which performed best in terms of median AUROC with a window length of 75 minutes (rather than 5 min with a sampling period of 1 second). This is overall in line with the other algorithms. NF(8) and NF(32) got their best performances with the smallest sampling period of 1 second.

Among the algorithms with performances under 0.86, GMM(2) was best with median AUROC 0.79. GMM(2) performed best with sampling period and window length of 3 seconds and 105 minutes respectively. Then GMM(5) performed slightly less with 0.71 median AUROC and increased its performances by +18.96% with sampling period and

TABLE II: Algorithms trained under reference conditions for feature extraction: window length of 25 min and signal sampling period of 1 second. The results are presented as median scores over 20 training/testing runs and with inter-quartile range in parenthesis

	AUROC (test)	bAcc (test)	Sen. (test)	Spec. (test)	AUROC (train)
NF(32)	0.82 (0.19)	0.5 (0.0)	0.0 (0.0)	1.0 (0.0)	0.89 (0.08)
LR	0.8 (0.15)	0.68 (0.2)	0.58 (0.49)	0.81 (0.06)	0.92 (0.04)
MLP(8)	0.8 (0.08)	0.63 (0.15)	0.35 (0.35)	0.93 (0.06)	0.93 (0.03)
NF(8)	0.79 (0.19)	0.5 (0.0)	0.0 (0.0)	1.0 (0.0)	0.89 (0.05)
NB	0.75 (0.19)	0.57 (0.04)	0.18 (0.1)	0.97 (0.01)	0.89 (0.03)
GMM(2)	0.73 (0.13)	0.61 (0.06)	0.27 (0.14)	0.96 (0.02)	0.89 (0.05)
ELM(200)	0.62 (0.12)	0.62 (0.11)	0.36 (0.23)	0.91 (0.03)	0.91 (0.03)
GMM(10)	0.6 (0.12)	0.51 (0.03)	0.05 (0.08)	0.99 (0.01)	0.98 (0.02)
GMM(5)	0.6 (0.12)	0.53 (0.05)	0.11 (0.12)	0.97 (0.02)	0.93 (0.04)
MLP(32)	0.58 (0.18)	0.51 (0.11)	0.06 (0.24)	0.95 (0.03)	0.95 (0.04)
GMM(1)	0.55 (0.19)	0.51 (0.03)	0.03 (0.05)	0.99 (0.01)	0.96 (0.02)
ELM(1000)	0.55 (0.07)	0.55 (0.07)	0.17 (0.14)	0.93 (0.06)	0.93 (0.03)
ELM(500)	0.55 (0.06)	0.55 (0.06)	0.14 (0.15)	0.94 (0.04)	0.92 (0.05)

AUROC: Area under the receiver operating characteristic, bAcc: Balanced accuracy score, Sen.: Sensitivity, Spec: Specificity, Diag: Diagonal, Cov: Covariance, LR: Logistic regression, NB: Naïve Bayes, ELM: Extreme Learning Machine, MLP: Multi-layer Perceptron, GMM: Gaussian Mixture Model, NF: Normalizing flow model.

Fig. 1: Different algorithms with varying window length and fixed sampling period to 1 second.

window length of 60 seconds and 135 minutes. This is the only model benefiting from increasing the sampling period to 60 seconds. GMM with full covariance benefited from increasing the sampling period to 30 seconds but reached only an AUROC of 0.63. Similarly, GMM(10) reached a median AUROC 0.61 with a sampling period of 10 seconds. Lastly, among ELMs, ELM(500) benefited from increasing the sampling period from 1 to 3 seconds and reached a median AUROC of 0.66. ELM(200) and ELM(1000) performed optimally with a sampling period of 1 second and reached median AUROC of 0.68 and 0.65 respectively.

1) Feature importance for logistic regression: We report the feature importance for LR algorithm in figure 3. The two most important features were IBI derived features, skewness and kurtosis. Respiratory features standard deviation, mean and kurtosis were also important and ranked 3rd, 6th, 7th, respectively. The most important feature extracted from oxygen saturation signal was ranked 16th. The repetitive weight measurement and post natal age also had an influence with

rank 4th and 5th. The first static feature was the birth weight ranked 9th.

V. DISCUSSION

a) Model calibration: Model calibration consists for instance in making sure the statistical model output probabilities are consistent with the intuitive notion of probabilities [42]. Concretely, making sure that among the samples with a posterior probability of label 1 = 0.9, 90% are true positives and 10% are false positives. This is often ensured by fitting a transform of the initial probabilities and requires an extra validation set. Given our limited and imbalanced dataset, we chose not to separate an extra validation set from our training dataset. This leads to scoring which is only relevant for threshold agnostic metrics, such as AUROC.

b) Influence of the feature extraction parameters: First the results presented in table III show that all prediction algorithms are sensitive to the choice of window length and sampling period for feature computation. When both parameters are

TABLE III: Algorithms evaluated at different window length and sampling periods

Models name	Ref AUROC	Best AUROC	Best wlen	Best Te	Gain
ELM(1000)	0.55 (0.07)	0.65 (0.16)	75 min	Ts=1	19.17%
ELM(200)	0.62 (0.12)	0.68 (0.15)	105 min	Ts=1	8.34%
ELM(500)	0.55 (0.06)	0.66 (0.11)	95 min	Ts=3	19.68%
GMM(1)	0.55 (0.19)	0.63 (0.11)	55 min	Ts=30	14.32%
GMM(10)	0.6 (0.12)	0.61 (0.1)	45 min	Ts=10	1.26%
GMM(2)	0.73 (0.13)	0.79 (0.2)	105 min	Ts=3	7.23%
GMM(5)	0.6 (0.12)	0.71 (0.21)	135 min	Ts=60	18.96%
LR	0.8 (0.16)	0.87 (0.18)	55 min	Ts=3	9.65%
MLP(32)	0.58 (0.18)	0.69 (0.18)	115 min	Ts=1	18.57%
MLP(8)	0.8 (0.08)	0.86 (0.19)	145 min	Ts=3	7.95%
NB	0.75 (0.19)	0.87 (0.16)	75 min	Ts=3	15.58%
NF(32)	0.82 (0.19)	0.86 (0.16)	115 min	Ts=1	5.0%
NF(8)	0.79 (0.2)	0.86 (0.29)	75 min	Ts=1	8.0%

Diag: Diagonal, Cov: Covariance, LR: Logistic regression, NB: Naïve Bayes, ELM: Extreme Learning Machine, MLP: Multi-layer Perceptron, GMM: Gaussian Mixture Model, NF: Normalizing flow model.

Fig. 2: AUROC performances of NF(32) and LR when the sampling frequency is varied and the window length kept fixed to 45 minutes.

varied, all the models benefit from an increased window length. This indicates that features based even indirectly on the frequency spectrum (e.g. standard deviation, skewness, sample entropy) are more predictive than time domain features (e.g. min, max, mean). This is also partly observed in figure 3. More specifically, NB, LR, NF and MLP(8) with median AUROC performances over 0.86 benefit from an increase in window length from 25 minutes to a range between 55 minutes and 145 minutes. These algorithms perform best when the sampling period is kept small, 1 second for NF(8) and NF(32) and 3 second for NB, MLP(8) and LR. Among these algorithms, NB benefits the most from tuning the parameters with a +15.58% increase in performances, followed by LR (+9.65%), NF(8) (+8%), MLP(8) (+7.95%) and NF(32) (+5%). These results indicate that, normalizing flow models can perform reasonably well even under sub-optimal feature extraction conditions. As opposed to NB or LR which require a careful tuning of the parameters to reach good performances.

c) *Influence of sampling period:* In our setup, NB and LR reached their best performances when the sampling period was increased from 1 to 3 seconds. On one hand, averaging the values in a given period has a filtering effect, allowing the computation of features more easily classifiable by linear

Fig. 3: Median feature importance from LR algorithm with a window length of 55 minutes and a sampling period of 3 seconds. The median is computed over 20 algorithm training.

classification models (NB and LR). On the other hand, averaging values in the sampling period might remove some useful information from the signals. Models with the largest number of parameters, in our case NF(8) and NF(32) are able to compensate for the information loss and keep high performances for sampling periods up to 10 seconds (figure 2). Sampling periods above 3 seconds are only advantageous for the lowest performing GMM algorithms, GMM(5), GMM(10) and GMM(1) with full covariance. However, these models still perform poorly (at most 0.71 for GMM(5)). Overall, maintaining a low sampling period helps the best performing algorithms to perform even better when other parameters such as the window length is varied.

d) *Normalizing flow based classifier*: In this study we proposed to use normalizing flows in combination with maximum a posteriori classification framework for the NSD task. The formulation of the training frameworks (Eq. (8) and (9)) allow the statistical models to be regularized. This allows us to overcome the overfitting problem encountered by GMM with diagonal covariance (see section IV-B). We proposed two training frameworks, a generative framework where two models are trained separately on each class (Eq. (8)) and a discriminative framework where both models are trained jointly (Eq. (9)). In table (VII) we showed that both frameworks can reach high performances (median AUROC > 0.8). The discriminative framework has the advantage over the generative framework of maintaining relatively high performances for different architectures and regularization parameters.

Overall however, the best performing algorithms had an IQR above 0.16, NF(8) had the highest IQR of 0.29 (see table III). NF(32) had an IQR of 0.16, identical to that of the best performing NB and better than LR (0.18). Moreover, NF(32) had the smallest increase in performances (+5%) among the best performing algorithms (median AUROC ≥ 0.86). This indicates that NF(32) predictions are more reliable, even when the feature extraction is not optimal.

In theory, the optimal window length for feature extraction changes as the vital sign signals statistics change over time. This means that ideally, the window length for computing features should be continuously tuned for each patients. The design of algorithms adapting the window length is difficult in practice. Here we showed that NF(32) leverages its modeling power to adapt to sub-optimal window length.

e) *Generative vs discriminative*: In our study, generative classifiers differed from discriminative classifiers in the way they were trained and how they handled class imbalance. Purely generative classifiers handled the class imbalance in the statistical model, and both classes of samples were used in independent statistical models. This includes GMMs which did not show good performances on our NSD task. We observed that with 1 full covariance Gaussian distribution and more than 2 diagonal covariance Gaussian distributions, the models overfit the training data. The Naïve Bayes algorithm, corresponding to a model with 1 diagonal covariance Gaussian, performed well with a median AUROC of 0.87. In the present study we used normalizing flows as hybrid models. On one hand, the two classes of data are used jointly for training which is similar to the way discriminative classifiers are trained. On the other hand the class imbalance is handled in the statistical model, rather than in a cost sensitive learning framework at training time. This hybrid model combines the benefits of a statistically sound class imbalance modeling and efficient use of the training data. Our results showed good performances of such models, even when the feature extraction algorithm is not optimal. The performances were not significantly better than simple LR and Naïve Bayes. Logistic regression is the best performing discriminative model with performances similar to Naïve Bayes and Normalizing flow based classifiers. Purely discriminative models such as MLP and ELM showed poor performances.

f) *Interpretability*: We have seen that simple linear classification algorithms such as LR obtained similar performances than complex non-linear models when used with optimal feature extraction. In addition, the LR model is readily interpretable when analyzing the weights attributed to each feature (Figure 3), as opposed to MLP or NF algorithms. We found IBI to be the most important vital sign in our analysis, followed by RF and SpO₂. A potential explanation could be that RF derived from chest impedance is less reliable than ECG-derived IBI. Also, SpO₂ features would perhaps be more relevant if information about respiratory support and administered oxygen were included as an input variables.

g) *Model validation*: The lack of standardized datasets raises an issue for external validations of machine learning models. In our work we showed that the prevalence of "septic" patterns in the monitoring data of very low birth weight infants in a NICU varies from 0.4% to 0.5%. The prevalence of "septic" patterns in monitoring data is hospital and practice dependent, which complicates the evaluation and deployment of predictive models. Sharing health care machine learning datasets however remains ethically and technically difficult, and current publicly available datasets are often not representative and this prevents external evaluations of models.

VI. CONCLUSION

In our study, advanced ML models such as multi-layer perceptrons or normalizing flows, performed as well as much simpler classifiers such as Naïve Bayes and logistic regression. Advanced ML models nonetheless showed better performances when the feature extraction pipeline is not optimal tuned. This makes ML classifiers relevant on heterogeneous populations, when tuning the optimal feature extraction parameters is difficult. The lack of interpretability of these models is nonetheless limiting their applicability and deployment in clinical practice.

Acknowledgement: The authors would like to acknowledge Daniel Huvehag at CMM IT department for the help in the management of the compute server. Lars Särnå and Ronny Grover at Stockholm Region IT for the help in monitoring data collection. Karen Coste, Hanna Olsson for help collecting the clinical timelines data.

REFERENCES

- [1] K. D. Fairchild, R. L. Schelonka, D. A. Kaufman, W. A. Carlo, J. Kattwinkel, P. J. Porcelli, C. T. Navarrete, E. Bancalari, J. L. Aschner, M. W. Walker, J. A. Perez, C. Palmer, D. E. Lake, T. M. O'Shea, and J. R. Moorman, "Septicemia mortality reduction in neonates in a heart rate characteristics monitoring trial," *Pediatr Res*, vol. 74, pp. 570–575, Nov. 2013.
- [2] C. Fleischmann-Struzek, A. Mikolajetz, D. Schwarzkopf, J. Cohen, C. S. Hartog, M. Pletz, P. Gastmeier, and K. Reinhart, "Challenges in assessing the burden of sepsis and understanding the inequalities of sepsis outcomes between National Health Systems: Secular trends in sepsis and infection incidence and mortality in Germany," *Intensive Care Med*, vol. 44, pp. 1826–1835, Nov. 2018.
- [3] J. V. E. Brown, N. Meader, K. Wright, J. Cleminson, and W. McGuire, "Assessment of C-Reactive Protein Diagnostic Test Accuracy for Late-Onset Infection in Newborn Infants: A Systematic Review and Meta-analysis," *JAMA Pediatr*, vol. 174, pp. 260–268, Mar. 2020.
- [4] J. B. Cantey and C. R. Bultmann, "C-Reactive Protein Testing in Late-Onset Neonatal Sepsis: Hazardous Waste," *JAMA Pediatr*, vol. 174, pp. 235–236, Mar. 2020.

- [5] T. G. Connell, M. Rele, D. Cowley, J. P. Buttery, and N. Curtis, "How reliable is a negative blood culture result? Volume of blood submitted for culture in routine practice in a children's hospital," *PEDIATRICS*, vol. 119, no. 5, 2007.
- [6] S. Huber, B. Hetzer, R. Crazzolara, and D. Orth-Höller, "The correct blood volume for paediatric blood cultures: A conundrum?," *Clin Microbiol Infect*, vol. 26, pp. 168–173, Feb. 2020.
- [7] R. L. Schelonka, M. K. Chai, B. A. Yoder, D. Hensley, R. M. Brockett, and D. P. Ascher, "Volume of blood required to detect common neonatal pathogens," *J Pediatr*, vol. 129, pp. 275–278, Aug. 1996.
- [8] S. Saigal and L. Doyle, "An overview of mortality and sequelae of preterm birth from infancy to adulthood," *The Lancet*, 2008.
- [9] J. L. Wynn, "Defining neonatal sepsis," *Curr Opin Pediatr*, vol. 28, pp. 135–140, Apr. 2016.
- [10] M. P. Griffin, T. M. O'Shea, E. A. Bissonette, F. E. Harrell, D. E. Lake, and J. R. Moorman, "Abnormal Heart Rate Characteristics Preceding Neonatal Sepsis and Sepsis-Like Illness," *Pediatric Research*, vol. 53, p. 920, June 2003.
- [11] H. Cao, D. E. Lake, M. P. Griffin, and J. R. Moorman, "Increased Nonstationarity of Neonatal Heart Rate Before the Clinical Diagnosis of Sepsis," *Annals of Biomedical Engineering*, vol. 32, pp. 233–244, Feb. 2004.
- [12] D. E. Lake, J. S. Richman, M. P. Griffin, and J. R. Moorman, "Sample entropy analysis of neonatal heart rate variability," *Am. J. Physiol. Regul. Integr. Comp. Physiol.*, vol. 283, pp. R789–797, Sept. 2002.
- [13] B. P. Kovatchev, L. S. Farhy, H. Cao, M. P. Griffin, D. E. Lake, and J. R. Moorman, "Sample asymmetry analysis of heart rate characteristics with application to neonatal sepsis and systemic inflammatory response syndrome," *Pediatr. Res.*, vol. 54, pp. 892–898, Dec. 2003.
- [14] J. F. Hicks and K. Fairchild, "Heart rate observation (HeRO) monitoring was developed for detection of sepsis in preterm infants.[...] The HeRO monitor is now in use in many NICUs in the USA and was approved in 2012 for use in Europe.," p. 5, 2013.
- [15] R. Joshi, D. Kommers, L. Oosterwijk, L. Feijs, C. Van Pul, and P. Andriessen, "Predicting Neonatal Sepsis Using Features of Heart Rate Variability, Respiratory Characteristics and ECG-Derived Estimates of Infant Motion," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2019.
- [16] E. Helander, N. Khodor, A. Kallonen, A. Värri, H. Patural, G. Carrault, and P. Pladys, "Comparison of linear and non-linear heart rate variability indices between preterm infants at their theoretical term age and full term newborns," in *EMBE & NBC 2017* (H. Eskola, O. Väisänen, J. Viik, and J. Hyttinen, eds.), vol. 65, pp. 153–156, Singapore: Springer Singapore, 2018.
- [17] A. Värri, A. Kallonen, E. Helander, A. Ledesma, and P. Pladys, "The Digi-NewB project for preterm infant sepsis risk and maturity analysis," *FinJeHeW*, vol. 10, May 2018.
- [18] I. Stanculescu, C. K. I. Williams, and Y. Freer, "Autoregressive Hidden Markov Models for the Early Detection of Neonatal Sepsis," *IEEE J. Biomed. Health Inform.*, vol. 18, pp. 1560–1570, Sept. 2014.
- [19] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing," *IEEE Acoustics, Speech, and Signal Processing Newsletter*, vol. 38, pp. 18–44, Mar. 2021.
- [20] E. Persad, K. Jost, A. Honoré, D. Forsberg, K. Coste, H. Olsson, S. Rautiainen, and E. Herlenius, "Neonatal sepsis prediction through clinical decision support algorithms – a systematic review," *Acta Paediatrica*, vol. n/a, no. n/a.
- [21] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, p. e7068349, Feb. 2018.
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019.
- [23] C. León, G. Carrault, P. Pladys, and A. Beuchée, "Early Detection of Late Onset Sepsis in Premature Infants Using Visibility Graph Analysis of Heart Rate Variability," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 1006–1017, Apr. 2021.
- [24] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Cham: Springer International Publishing, 2018.
- [25] A. Ng and M. Jordan, "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes," *Advances in Neural Information Processing Systems*, vol. 14, 2001.
- [26] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," *arXiv:1605.08803 [cs, stat]*, May 2016.
- [27] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing Flows: An Introduction and Review of Current Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
- [28] A. Honoré, D. Liu, D. Forsberg, K. Coste, E. Herlenius, S. Chatterjee, and M. Skoglund, "Hidden Markov Models for Sepsis Detection in Preterm Infants," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1130–1134, May 2020.
- [29] D. Liu, A. Honoré, S. Chatterjee, and L. K. Rasmussen, "Powering Hidden Markov Model by Neural Network based Generative Models," *arXiv:1910.05744 [cs, stat]*, Oct. 2019.
- [30] D. Nabil and F. Bereksi Reguig, "Ectopic beats detection and correction methods: A review," *Biomedical Signal Processing and Control*, vol. 18, pp. 228–244, Apr. 2015.
- [31] S. Akselrod, D. Gordon, F. A. Ubel, D. C. Shannon, A. C. Berger, and R. J. Cohen, "Power spectrum analysis of heart rate fluctuation: A quantitative probe of beat-to-beat cardiovascular control," *Science*, vol. 213, pp. 220–222, July 1981.
- [32] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, pp. 513–529, Apr. 2012.
- [33] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, June 2006.
- [34] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution," in *2010 20th International Conference on Pattern Recognition*, pp. 3121–3124, Aug. 2010.
- [35] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, Aug. 1989.
- [36] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.
- [37] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, "Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function," *Neural Process Lett*, vol. 50, pp. 1937–1949, Oct. 2019.
- [38] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, pp. 47–60, Nov. 1996.
- [39] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, "Scikit-learn: Machine Learning in Python," *MACHINE LEARNING IN PYTHON*, p. 6.
- [41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," p. 12.
- [42] On behalf of Topic Group 'Evaluating diagnostic tests and prediction models' of the STRATOS initiative, B. Van Calster, D. J. McLernon, M. van Smeden, L. Wynants, and E. W. Steyerberg, "Calibration: The Achilles heel of predictive analytics," *BMC Med*, vol. 17, p. 230, Dec. 2019.