Simulation of Health Time Series with Nonstationarity

Adedolapo Aishat Toye ATOYE@STEVENS.EDU
Louis Gomez LGOMEZ@STEVENS.EDU
Samantha Kleinberg SAMANTHA.KLEINBERG@STEVENS.EDU
Department of Computer Science, Stevens Institute of Technology, USA.

Abstract

Limited access to health data remains a challenge for developing machine learning (ML) models. Health data is difficult to share due to privacy concerns and often does not have ground truth. Simulated data is often used for evaluating algorithms, as it can be shared freely and generated with ground truth. However, for simulated data to be used as an alternative to real data, algorithmic performance must be similar to that of real data. Existing simulation approaches are either black boxes or rely solely on expert knowledge, which may be incomplete. These methods generate data that often overstates performance, as they do not simulate many of the properties that make real data challenging. Nonstationarity, where a system's properties or parameters change over time, is pervasive in health data with changing health status of patients, standards of care, and populations. This makes ML challenging and can lead to reduced model generalizability, yet there have not been ways to systematically simulate realistic nonstationary data. This paper introduces a modular approach for learning dataset-specific models of nonstationarity in real data and augmenting simulated data with these properties to generate realistic synthetic datasets. We show that our simulation approach brings performance closer to that of real data in stress classification and glucose forecasting in people with diabetes.

Data and Code Availability The OHIO (Marling and Bunescu, 2020) and OpenAPS (Melmer et al., 2019) datasets are available upon agreement with the dataset authors. The WESAD (Schmidt et al., 2018) dataset is publicly available.[1] The code is available at https://github.com/health-ai-lab/Nonstationarity-Simulation.

1. https://www.eti.uni-siegen.de/ubicomp/home/datasets/icmi18/

Institutional Review Board (IRB) This study was approved as exempt by the IRB at Stevens Institute of Technology.

1. Introduction

Large amounts of health data are being generated from medical records and patient generated sources, enabling the rapid advance of machine learning (ML) for healthcare. However, many researchers do not have access to these datasets, as they cannot be easily shared due to privacy concerns. While there are publicly available datasets such as MIMIC-III (Johnson et al., 2016) and eICU (Pollard et al., 2018), they are mainly limited to intensive care unit data collected from a single location and may not generalize to other populations. Additionally, since health data is primarily generated for patient care and billing rather than for research, these datasets rarely have the ground truth needed for evaluating algorithms.

Simulated data can address these challenges as it can be generated with ground truth and shared without privacy concerns. However, current simulation approaches either generate data that does not have the same performance on ML tasks as real data (e.g., in glucose forecasting (Zhu et al., 2020)), or use black-box models, preventing ablation studies of how data properties affect performance. While recent simulation approaches have incorporated properties such as missing data and error into simulated data (Gomez et al., 2023), many additional real data properties that make ML challenging, such as nonstationarity, have not yet been explored.

Nonstationarity occurs when the statistical properties of data or the data-generating process change over time, such as changing hospital treatment practices or patient health status. This property poses challenges for many ML and causal inference methods, which often assume that data is stationary (Jung and Shah, 2015). For example, models may not gen-

eralize well to future time periods if the data distribution changes (Sahiner et al., 2023; Mårtensson et al., 2020; Nestor et al., 2019), such as ML models trained on pre-COVID-19 data performing worse for predicting hospital admissions when applied to early COVID-19 pandemic data (Duckworth et al., 2021). Additionally, many causal inference methods assume stationarity to guarantee the correctness of inferences (Assaad et al., 2022), yet there are no datasets for systematically testing how violations of this assumption (which are common in healthcare) affect results. As a result, ML models often perform better on simulated data than in real-world applications. Augmenting simulated data with nonstationarity may help bring performance closer to real data, which is important for model development and translation.

Methods for simulating nonstationarity often introduce random changes in properties such as a variable's mean (Yu et al., 2023; Li et al., 2023a) or variance (Van den Burg and Williams, 2020). However, this does not capture the nonstationarity found in healthcare data, which is systematic and may follow temporal patterns. For example, the glucose profile of a person with diabetes can change due to events such as weight loss (Marsden et al., 2022) or menstruation (Lin et al., 2023). Additionally, changepoints have mainly been simulated as happening at a single time instant (Wambui et al., 2015; Yu et al., 2023), while in health data, changes may occur over durations (e.g., gradual shift in disease severity). While methods have been developed to detect gradual changes (Ebrahimzadeh et al., 2019), they were tested using changes simulated with random durations. In practice, this is unrealistic, and durations may be influenced by other variables, such as electrodermal activity (EDA) gradually rising as an event becomes stressful.

We now address the limitations of existing simulation approaches by learning models of nonstationarity and augmenting simulated data with this property to generate more realistic data with similar performance to real data. We assume a generative model and set of real data, model nonstationarity using the real data, and then add this property to the simulated data. As our use cases, we focus on two important health applications for which generative models exist yet do not capture the full complexity of real-world data: glucose forecasting for people with Type 1 Diabetes (T1D), and stress detection from EDA data. Our key contributions are (i) methods to learn models of nonstationarity from real data and (ii) showing that adding nonstationarity to the simulation brings ML performance on simulated data closer to real data.

2. Related Work

We discuss (i) simulation of health data, and (ii) simulation of nonstationarity, both within and outside of healthcare.

2.1. Synthetic Health Data Simulation

Health data has been simulated using three main approaches: knowledge-based, data-driven, or hybrid.

Knowledge-based methods generate synthetic data using mathematical models that simulate complex human physiology. This approach has been used in several application areas, such as glucose simulation for people with T1D (Man et al., 2014), EDA (Bach et al., 2010), and brain activity (Wakeland and Goldstein, 2008). Knowledge-based approaches rely on expert knowledge, which is a limitation as we may not have complete knowledge of complex biological processes. A second key limitation is that these models aim to simulate biological processes rather than the data that is recorded, resulting in overstated performance when compared to real data (Zhu et al., 2020). For example, in diabetes, models have been developed to simulate the dynamics between blood glucose (BG) and insulin (Man et al., 2014; Wilinska et al., 2010), and have been approved by the FDA for testing BG control algorithms (Kovatchev et al., 2008). However, performance on BG forecasting is often significantly better on simulated data than real data (Li et al., 2019; Zhu et al., 2020). One reason for this performance gap is that the simulation system does not include other factors that affect BG such as stress (Riazi et al., 2004) and menstruation (Milionis et al., 2023). These factors could be added to the models, but this still does not guarantee comparable performance to real data. A larger reason for this gap is that these models do not include real data properties (e.g., measurement error, nonstationarity) which are pervasive in health data. For example, a commonly used glucose simulator (Man et al., 2014) assumes some of the model parameters (e.g., bodyweight) are fixed, yet in reality these parameters can change over time resulting in nonstationarity in the data.

Data-driven methods address some of the limitations of knowledge-based methods by learning a model directly from real data without requiring ex-

pert knowledge of the data generation process. This approach has been used to generate electronic health record (EHR) data: EMERGE simulates data for specific disease outbreaks (Lombardo and Moniz, 2008; Buczak et al., 2010), OMOP simulates data of diseases and treatments using characteristics extracted from real data (Murray et al., 2011), and both Synthea (Walonoski et al., 2018) and CoMSER (McLachlan et al., 2016) simulate data using health statistics and clinical guidelines without relying on real EHR data. However, these works focus on recreating population characteristics, rather than evaluating performance on machine learning tasks. Based on the success of Generative Adversarial Networks (GANs) in simulating other data such as images, they have been used to simulate health data such as BG (Cichosz and Xylander, 2021), EDA (Ehrhart et al., 2022), and EHR data (Choi et al., 2017; Li et al., 2023b). Although GANs are considered state of the art, they are susceptible to privacy concerns as they can potentially memorize training data (Hitaj et al., 2017). They are also black box models and cannot be used for ablation studies (e.g., causal inference with and without nonstationarity). Data generation methods are usually evaluated by comparing statistical properties of the data with those of real data (Figueira and Vaz, 2022). However, this does not guarantee that performance on ML tasks will be similar to that of real data, which is important if researchers hope to use simulated data as an alternative to real data for the development of ML models in healthcare.

The hybrid method of generating simulated data is a mixture of the two aforementioned methods, where data generated from knowledge-based methods is augmented with properties learned from real data. This method has been explored in several health applications such as simulating glucose data (Gomez et al., 2023), cardiac images (Prakosa et al., 2012), and brain MRIs (Khanal et al., 2017). For example, Data Augmented Simulation (DAS) (Gomez et al., 2023) was developed to simulate glucose data by learning missing and error patterns from real data and augmenting simulated data with these learned properties. This method of learning individual data properties allows researchers to conduct ablation studies to identify which data property (e.g., error or missing data) is responsible for algorithm performance. While DAS shows that simulating features of real data rather than only biological processes can bring the performance on simulated data closer to that on real data for ML tasks, that work only examined missing data and error, which are not the only factors contributing to this performance gap. Data properties such as nonstationarity pose challenges for ML and can impact model performance (Jung and Shah, 2015; Duckworth et al., 2021; Rahmani et al., 2023).

2.2. Simulating Nonstationary

Nonstationarity can be characterized by the frequency, type (e.g., a change in mean), magnitude, and duration of the changes in the data. Many existing methods for simulating nonstationarity were developed to test changepoint detection algorithms, which might limit the ability of the simulation to capture the fluctuations in health data accurately. For example, the number of changepoints is often simulated randomly (Ebrahimzadeh et al., 2019; Van den Burg and Williams, 2020; Li et al., 2023a) or following a Poisson distribution (Shi et al., 2022) but this does not capture the non-random or varying changepoint frequencies across different patients' data. The changepoints are simulated to occur randomly (Cummings et al., 2020; Li et al., 2023a), at specific time intervals (Wambui et al., 2015), or within a specified range (Yu et al., 2023). Introducing changes at random times is not realistic for health data, where changes are usually due to an underlying factor. Many methods simulate abrupt changes as a shift in mean or variance (Yu et al., 2023; Cummings et al., 2020; Van den Burg and Williams, 2020; Wambui et al., 2015) where the size of the changes is completely random, while gradual changes are simulated with random durations. However, changes in health data may be gradual over specific durations of time (e.g., weight loss).

To be a viable complement to health data, simulated data needs to lead to similar algorithmic performance as real data. Nonstationarity is a key challenge for ML methods, yet there is not yet a way to simulate realistic nonstationary data. Instead, we propose learning dataset-specific models of nonstationarity and use them to augment generative models to emulate real-world health data better.

3. Methods

We introduce our method for generating more realistic simulated data by learning a model of nonstationarity from real data and then augmenting simulated

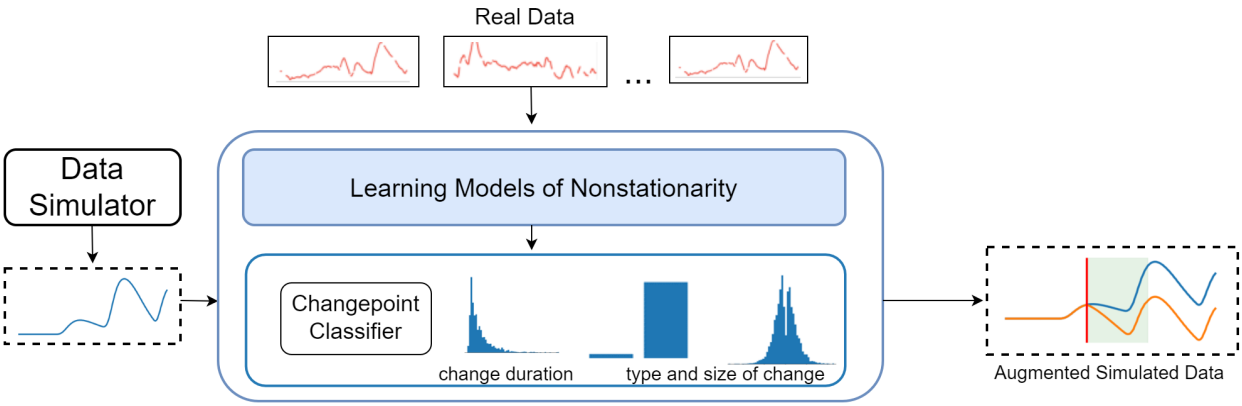


Figure 1: An overview of our method for generating simulated data with nonstationarity. We begin with simulated data, learn nonstationarity from real data, and add this property to the simulated data.

data with this property. We first describe how we learn a model of nonstationarity, then how we add these properties to the simulated data generated from an existing model. See Figure 1 for an overview of our method.

3.1. Preliminaries

We assume a generative model exists, but there is a gap in performance between the data it generates and real data on ML tasks. We begin with a univariate regularly sampled time series $V = \{v_1, v_2, ..., v_T\} \in \mathbb{R}^{1\times T}$ where $v_t \in \mathbb{R}$ is an observation at time t, and T is the length of the timeseries.

We aim to model the nonstationarity (i.e., when the changes occur; and the type, magnitude, and duration of the changes) for a single variable $v \in \mathbb{R}^{1\times T}$. For glucose and EDA data, the variable represents the CGM readings and EDA signal, respectively. Let $\mathcal{C} = \{c_1, c_2, ..., c_{|\mathcal{C}|}\}$ represent the set of changepoints we aim to identify in v. Each change c_i is represented as a tuple $(n_i, \tau_i, k_i, \alpha_i)$, where n_i is the time the changepoint occurs, τ_i is the duration of the change, k_i is the type of change, and α_i is the magnitude of the change.

3.2. Learning Nonstationarity From Real Data

In our approach, learning patterns of nonstationarity from real data involves learning when a change occurs; and the type, duration, and magnitude of the change. Given the absence of ground truth indicating when changepoints occur in the real data, we first identify the changepoints and the duration of each change $\tau_{1:|C|}$. To do this, we use a modification of the Trendet algorithm (Bartolome, 2020) as this algorithm detects changepoints and their duration. A sample signal with identified changepoints and durations is shown in Figure 2. Then, using the data along with the labels indicating the presence or absence of a change at each timepoint, we frame the task of predicting when a changepoint occurs as a time series classification problem. We train a classifier using statistical and temporal features (see Appendix A) computed over an extracted history window ($v_{t-w:t-1}$) of a fixed size w to predict if a changepoint occurs at time v_t.

Next, we determine the type of change k, which can be (i) a mean shift if there is a difference in the mean values, (ii) a standard deviation (SD) change if there is a difference in the SD (iii) a change in both if both properties differ. To do this, we compute the mean and SD of the values between the current and previous changepoint, and compare them with the mean and SD of the values between the current and the next changepoint. Similarly, to quantify the magnitude of change α_i at each changepoint, we compute the difference in the statistical property (mean and standard deviation) of the values before and after the changepoint. Subsequently, we determine the most suitable distribution that best fits the duration $\tau_{1:|C|}$, type $k_{1:|C|}$, and magnitude $\alpha_{1:|C|}$ of all changes in the data. We use maximum likelihood estimation (MLE) method to evaluate various distribution models, including exponential, bimodal, normal, and uniform

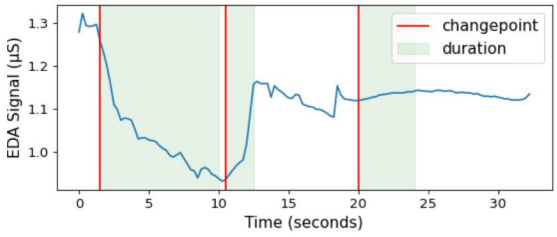


Figure 2: A sample EDA signal with identified changepoints and duration of chnage.

distributions. The MLE method is defined as:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \mathcal{L}(\theta \mid \text{data})$$

where, $\hat{\theta}_{MLE}$ represents the estimated parameters that maximize the likelihood function $\mathcal{L}(\theta \mid \text{data})$ and data is either $\tau_{1:|C|}$ or $\alpha_{1:|C|}$. For instance, if the data is τ, the likelihood function is:

$$\mathcal{L}(\theta \mid \tau) = \sum_{i=1}^{|C|} \log(f(\tau_i \mid \theta))$$

At the end of this stage, we have the changepoint classifier and the distributions of the change properties. We then use these learned models of nonstationarity to add changes to the simulated data.

3.3. Augmenting Simulated Data with Nonstationarity

To add the changes learned from real data to simulated data, we implement a post-processing step where the input is simulated data and the output is the augmented data with nonstationarity added to it. This post-processing step involves two processes: (i) first, we predict when changepoints occur and their properties (duration, type, and magnitude of the changes), (ii) second, we modify the simulated data based on these change properties to generate the augmented data.

To predict when the changepoints occur, we compute statistical and temporal features (see Appendix A) over a history window of length w and iteratively make predictions using the changepoint classifier described in Section 3.2. Once a changepoint is identified at t, we assign a duration of change τ_i by sampling from the change duration distribution derived in the learning phase. We move to the time point after the end of the change duration $t+\tau_i$ and repeat this prediction process until the end of the time series. Once we have predicted when the changepoints occur and the duration of each change, we then assign a corresponding type k_i and magnitude of change α_i for all the changepoints by generating values from their respective distributions derived in the learning phase. Then, for each changepoint, we adjust the values between the current changepoint and the beginning of the next changepoint using the type, duration and magnitude of the change.

If the predicted changepoint at t is a change in mean with magnitude, α_i^{mean} and duration, τ_i, we first compute the mean of values between (i) the current and previous changepoint (μ_{prior}) and (ii) the current and next changepoint (μ_{current}). Then, each value in $v_{t:t+\tau_i}$ is adjusted incrementally by a fraction of the total change based on the elapsed time since the changepoint started. After $t+\tau_i$, the values are shifted by the total change until the beginning of the next changepoint. The adjustment of the values can be represented as:

$$v'_{t'} = v_{t'} + (\mu_{\text{prior}} + \alpha_i^{mean} - \mu_{\text{current}}) \cdot \frac{\min(t'-t, \tau)}{\tau}$$

where:

- t' represents each time step between t and the next changepoint
- $v'_{t'}$ is the adjusted value at each time step between t and the next changepoint,
- $v_{t'}$ is the original value at each time step between t and the next changepoint.

If the predicted changepoint at t is a change in SD with magnitude α_i^{SD}, we first compute the SD of values between (i) the current and previous changepoint (σ_{prior}) and (ii) the current and next changepoint (σ_{current}). We then scale the values based on the ratio of the new SD ($\sigma_{\text{prior}}+\alpha_i^{SD}$) and the current SD σ_{current}, to reflect the change in SD. This can be represented as:

$$v'_{t'} = (v_{t'} - \mu_{\text{current}}) \cdot \frac{(\sigma_{\text{prior}} + \alpha_i^{SD})}{\sigma_{\text{current}}} + \mu_{\text{current}}$$

If the predicted changepoint at t is a change in both mean and SD, we implement the approaches sequentially, first scaling the values to reflect the change in SD, then adjusting the values to reflect the change in mean.

4. Experiments

We carry out experiments on two data types (glucose and EDA) to evaluate our method and show that adding nonstationarity brings the performance of simulated data closer to real data on glucose forecasting and stress detection. Glucose forecasting is important as it can help improve insulin dosing for people with T1D, hence ensuring that models trained using simulated data have similar performance to real data is vital for improving management of T1D. Stress detection is also important for improving stress management as prolonged stress can be detrimental to human health. We first describe the datasets used, then describe the baselines we compare against. Finally, we discuss our experiments on glucose forecasting and stress detection to show that simulated data augmented with nonstationarity has closer performance to real data.

4.1. Datasets

We describe the real and simulated datasets used in this work for both glucose and EDA experiments.

4.1.1. Glucose Datasets

OhioT1DM (Marling and Bunescu, 2020) dataset has been widely used for developing BG forecasting methods. The data includes CGM readings (recorded every 5 minutes), BG readings (recorded when taken), insulin, bolus doses (recorded when they are administered), physiological sensor readings (e.g, heart rate), and meal intake (self-reported) collected from 12 adults with Type 1 diabetes (T1D) over a period of 8 weeks. We use only the glucose data for our experiments.

Open source artificial pancreas system (OpenAPS) (Melmer et al., 2019) dataset is a patient-generated dataset from individuals with T1D who manage their diabetes using an open-source artificial pancreas system and donate their data voluntarily. It contains CGM readings (recorded every 5 minutes), basal rates, bolus doses (recorded when administered), and meal intake (size in grams and time of intake) for 86 people over an average of 308 days of data per participant.

Simulated Data was generated using the implementation (Xie, 2018) of the UVA/PADOVA T1D simulator (Man et al., 2014) to match the characteristics of each real dataset. We generate glucose readings, meals, and insulin values using a set of model and input parameters (e.g., weight and meal events). The meal and physical activity events were generated following the description in (Gomez et al., 2023). See Appendix B for more details. Data was generated for 10 adults with 54 and 308 days of data per subject to match the characteristics of OHIO and OpenAPS data respectively. We use only the simulated glucose data for our experiments.

4.1.2. EDA Dataset

WESAD (Schmidt et al., 2018) dataset has been widely used for stress detection. The data includes physiological data (such as EDA, body temperature, electrocardiogram, respiration) collected from 15 participants who were exposed to neutral, stress and amusement conditions. We focus on the EDA signals which represent changes in the electrical conductance of the skin in response to various physiological events such as stress. The EDA was recorded using the wrist device (Empatica E4) and sampled at 4Hz.

Simulated Data is generated to match the basic characteristics of the WESAD dataset. We use the Neurokit2 Python package (Makowski et al., 2021) implementation of an EDA simulator developed in previous work (Bach et al., 2010). We set the duration parameter for the baseline and stress states as 1174 seconds and 664 seconds respectively to match the average duration observed in the real data. The SCR peaks count parameter is determined using a uniform distribution $\mathcal{U}(1, 5)$ for the baseline state and $\mathcal{U}(6, 20)$ for the stress state (Vasile et al., 2023). We generate EDA signals for 10 individuals using these parameters. Each individual's data contains a combination of baseline and stress states.

4.2. Baselines

We compare our method against four baseline methods for adding nonstationarity to simulated data. The baselines were selected to evaluate current practices of simulating nonstationarity, namely by introducing random shifts in a data property with random magnitude and duration. The number of changepoints, magnitude and duration of the changes are selected randomly from the range of values of the change properties in the real data.

MeanShift-Constant We place a fixed number of changepoints at random times with a change in mean of random magnitude and duration of change. For

glucose data, we place 2 changepoints for each day in the dataset, with magnitude of 5 mg/dL, and duration of 30 minutes. For EDA, we place 20 changepoints across the data with magnitude of 1 µS and duration of 10 seconds.

MeanSDShift-Constant We add the same changes as *MeanShift-Constant*, and add a change in standard deviation (SD) of 0.5 mg/dL and 0.5 µS across the changepoints for glucose and EDA data respectively.

MeanShift-Varying Now, instead of using fixed values, we simulate the number of changepoints from $\mathcal{N}(3, 1)$. We generate only changes in mean with magnitude sampled from $\mathcal{N}(0, 10)$ with a fixed duration of 30 minutes. Similarly, for EDA, the number of changepoints is sampled from $\mathcal{N}(20, 10)$ with magnitude of $\mathcal{N}(0, 1)$ and duration of 10 seconds.

MeanSDShift-Varying We add changes in the same way as *MeanShift-Varying* with an additional change in SD of ($\mathcal{N}(0, 1)$) across the changepoints for OHIO and OpenAPS, and ($\mathcal{N}(0.1, 0.01)$) for EDA.

4.3. Learning Nonstationarity

We conduct experiments to test our method of learning models of nonstationarity from real data. We first divide each dataset into non-overlapping subsets for learning nonstationarity and ML tasks to avoid data leakage. We divide it into 70% (60 subjects) and 30% (26 subjects) for OpenAPS. For OHIO we divide into 6 weeks/2 weeks of data due to the number of subjects, while for WESAD we divide into 60/40% to have a sufficient number of subjects in both subsets.

To learn the changes in the real data, we use input sequences of CGM and EDA data. We compute statistical and temporal features (see Appendix A) over an extracted window of 60 minutes (for glucose data) and 5 seconds (for EDA data) using Tsfel time series feature extraction package (Barandas et al., 2020). For OpenAPS, we split the 60 subjects into a train and test sets of 80/20% and train an XGBoost classifier. We adjust the class weights based on the class frequency to address the data imbalance, and evaluate using AUROC. For OHIO and WESAD, due to the smaller number of subjects, we use leave one subject out cross validation (LOOCV) and report the average performance across the subjects. Thereafter, we determine the duration, type, and magnitude of each change and estimate their corresponding distributions following the procedures outlined in 2.2.

4.4. Experiments on ML Tasks

We aim to test whether our method of augmenting simulated data with nonstationarity leads to similar performance as real data on glucose forecasting and stress classification tasks when compared to the baseline methods.

Glucose Forecasting We perform forecasting using various algorithms previously used for glucose forecasting: Linear Regression (REG), Random Forest Regression (RF), Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) as defined in (Hameed and Kleinberg, 2020). We apply the same preprocessing steps used in that work. We use default parameters for LR and RF, while for RNN and LSTM, we use a hidden layer consisting of 32 units, a batch size of 248, a maximum of 50 epochs, and early stopping at 15 epochs. Each model was run through 10 iterations for each dataset, and the average Root Mean Squared Error (RMSE) across these iterations was reported.

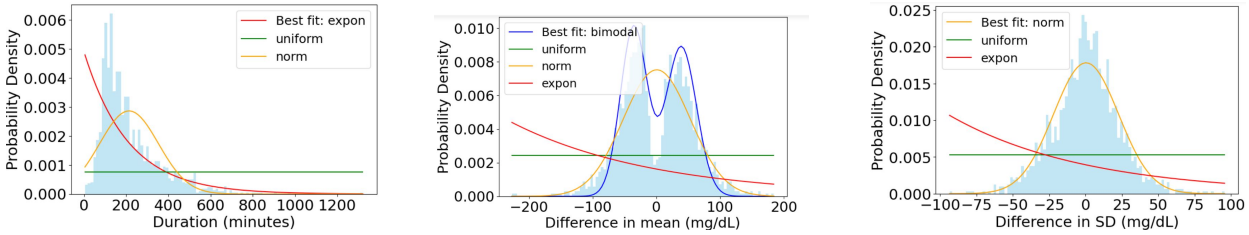
Stress Classification We apply the same preprocessing steps as outlined in (Garg et al., 2021). We segment the EDA signals into non-overlapping 10-second windows and extract statistical features including mean, standard deviation, minimum, and maximum values for each window. We train a variety of ML algorithms that have previously been used for this task: Logistic Regression (LR), k-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF) to predict stress and baseline states. We use default parameters for LR and SVM. For KNN, we set the number of neighbors to 100. For RF, we set the number of trees to 50 and the minimum number of samples required to split a node to 5. We evaluate using leave-one-participant-out approach and report the average accuracy and F1 score for each model.

5. Results

We now discuss the results of learning models of nonstationarity from real data, and the ML tasks used to evaluate our approach.

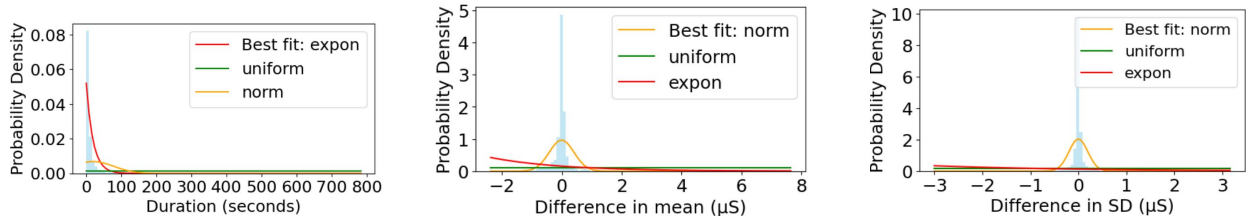
5.1. Results on Learning Nonstationarity

Table 2 shows our results for predicting changepoints on the three datasets. We consider a prediction as



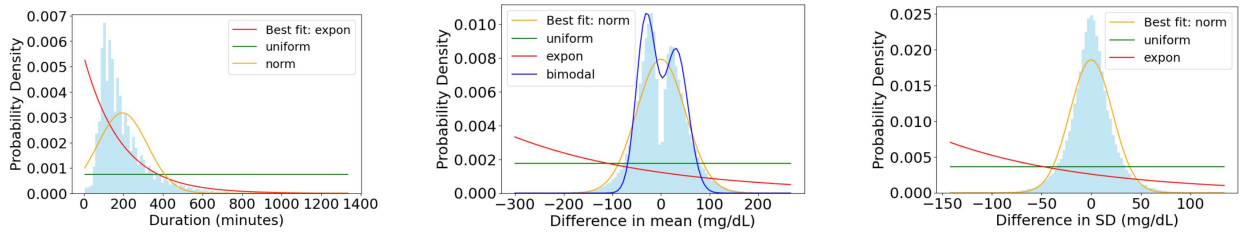
(a) Duration (b) Mean change (c) Standard deviation change

Figure 3: Distribution of the changepoint properties for OHIO dataset.



(a) Duration (b) Mean change (c) Standard deviation change

Figure 4: Distribution of the changepoint properties for WESAD dataset.



(a) Duration (b) Mean change (c) Standard deviation change

Figure 5: Distribution of the changepoint properties for OpenAPS dataset.

Table 1: Mean RMSE results when comparing performances between real and simulated data for glucose forecasting. Values in bold indicate the simulated dataset with the closest performance to real data

Simulated Dataset	**OHIO**				**OpenAPS**			
	REG	RF	RNN	LSTM	REG	RF	RNN	LSTM
Raw Simulated	3.27	6.27	6.66	10.09	3.14	4.15	5.60	12.71
MeanShift-Constant	10.44	9.25	10.81	12.79	9.80	7.90	8.14	10.14
MeanSDShift-Constant	14.66	12.20	13.22	14.53	13.10	9.78	10.79	13.00
MeanShift-Varying	12.02	11.43	12.51	13.41	11.73	9.41	10.16	12.55
MeanSDShift-Varying	15.13	13.51	13.50	14.16	14.86	11.81	12.20	13.30
Our Method	**22.03**	**21.83**	**21.79**	**22.94**	**17.53**	**15.86**	**16.19**	**19.35**
Real Data	24.56	23.06	24.30	23.48	20.29	19.82	19.41	20.05

Table 2: AUROC results for predicting the changepoints in the data. We use varying tolerance ranges η for how close a predicted change must be to be considered a true positive.

Dataset	AUROC		
	η=0	η=15min	η=30mins
OHIO	0.62	0.71	0.77
OpenAPS	0.63	0.70	0.76
		η=30sec	η=60sec
WESAD	0.67	0.71	0.76

correct if it is at the exact time of the true changepoint. As this is strict, we also evaluate results using tolerance ranges, where a prediction is considered correct if it is within η time units before or after the true changepoint. We use tolerance values of ± 15 and 30 minutes (for glucose data), and ± 10 and 20 seconds (for WESAD). As shown in Table 2 the AUROC improves significantly as the tolerance value increases. Thus while it is challenging to predict the exact time of a changepoint, our approaches predict changepoints close to those identified by the changepoint detection algorithm.

Next we examine the properties of the identified changepoints. The distributions of the changepoint properties derived from the datasets are shown in Figure 3, Figure 4 and Figure 5. Across all datasets, the duration of the changepoints is best captured by an exponential distribution with parameters $Exp(0.005)$, $Exp(0.0052)$ and $Exp(0.05)$ for OHIO, OpenAPS and WESAD respectively. The magnitudes of the change in mean for OHIO are best represented with a bimodal $\mathcal{B}(38.77, 23.94, -35.62, 20.92)$, while for OpenAPS and WESAD, they are represented with normal distributions with parameters $\mathcal{N}(-0.01, 50.26)$, and $\mathcal{N}(-0.01, 0.41)$. The change in SD across all the datasets are best captured with a normal distributions $\mathcal{N}(0.30, 22.40)$, $\mathcal{N}(0.01, 21.44)$, and $\mathcal{N}(0, 0.20)$ for OHIO, OpenAPS, and WESAD respectively.

5.2. Results on ML Tasks

We compare our approach to other baselines for adding nonstationarity to simulated data on glucose forecasting and stress classification tasks.

Glucose forecasting on simulated data As shown in Table 1 raw simulated data appears to have the best RMSE but its performance is the farthest from real data. Simulated data generated with our method has the overall closest performance to real data across both datasets. We use a t-test to compare results of our method to the best performing baseline, *MeanSDShift-Varying*. We find that the difference in RMSE between our method and *MeanSDShift-Varying* was statistically significant across all models (all $ps < 0.006$). While simulating both a change in mean and SD brings performance closer to real data compared to a change in mean only (as seen in *MeanSDShift-Varying* and *MeanSDShift-Constant*), learning dataset-specific patterns of nonstationarity better emulates real data performance.

Stress classification on simulated data Similar to glucose forecasting, the raw simulated data and the baselines overstate the performance of the classification models, while our method's performance is closest to the real data as shown in Table 3. When comparing our method to *MeanSDShift-Varying* which is the second closest to the real data, we see that the accuracy and F1 score are statistically significantly different across the classification models (all $ps < 0.005$). Our results further show that all the baselines perform almost the same as the raw simulated data. This performance is likely because our baseline methods simulate nonstationarity using random change properties which may not reflect the actual changes that occur in the real data.

6. Discussion

Simulated data is often used for evaluating ML algorithms. However, existing simulation methods sometimes do not guarantee similar performance to real data or are black boxes hindering our ability to identify the data properties responsible for performance. An important data property that poses challenges for many ML and causal models is nonstationarity, as these models are developed with stationary assumptions. Due to this, simulated data that lacks nonstationarity are likely to have a higher performance on models compared to real-world data where nonstationarity is pervasive. This overstated performance is detrimental in practice where inaccurate predictions may affect health decisions such as insulin dosing in diabetes management. It is important that performance closely mirrors that of real-world data, for sim-

Table 3: Performance on real and simulated data for stress prediction. Values in bold indicate the simulated dataset with the closest performance to real data.

Dataset	KNN		LR		RF		SVM	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
Raw Simulated	0.99	0.99	0.99	0.98	1.00	1.00	1.00	1.00
MeanShift-Constant	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.99
MeanSDShift-Constant	0.98	0.98	0.99	0.98	0.99	0.99	0.99	0.98
MeanShift-Varying	0.99	0.98	0.98	0.98	1.00	0.99	1.00	0.99
MeanSDShift-Varying	0.96	0.94	0.98	0.97	0.98	0.98	0.97	0.97
Our Method	**0.79**	**0.55**	**0.74**	**0.49**	**0.82**	**0.71**	**0.81**	**0.60**
Real Data	0.69	0.69	0.70	0.71	0.67	0.68	0.73	0.72

ulated data to be suitable for training ML models in critical domain areas like healthcare.

To address this, we introduced a method for adding nonstationarity to simulated data. Our approach outperformed other baselines in bringing performance closer to real data on glucose forecasting and stress detection tasks. This shows that learning models directly from real data and encoding them into simulated data may help replicate the varying fluctuations that are usually present in real-world data. This helps to ensure that algorithm behavior on simulated data is not so different from that of real data. The main limitation is our inability to generalize to other areas where a generative model does not exist. This is because our simulation method builds on an existing generative model which might not be available in all domain areas. In future, we aim to expand our method to cases where there is no model by simulating the underlying data generation process using the data only. Another limitation is the absence of ground truth to verify the identified changepoints because obtaining annotated data is challenging and time-consuming. This limitation shows the need for labeled data for learning reliable models of nonstationarity from real data. As obtaining labels is often difficult and labor intensive, innovative ways of learning these models with minimal data needs to be explored. Further, we focus only mean and SD and did not address additional statistical measures that may also contribute to complexity in real data. Simulating these additional variations may further bring the algorithm performance on simulated data closer to real data. Finally, our method of aggregating features such as mean may pose privacy risks. While these risks may be negligible in our models compared to deep learning models, future research is needed to explore the application of differential privacy to help reduce these risks.

7. Conclusion

We develop an approach for learning models of nonstationarity in real data and subsequently augmenting simulated data with this property. We test our approach on common health-related time series data, such as glucose and EDA data, and show that our approach brings simulated data performance closer to real data on glucose forecasting and stress classification tasks. This work shows that models of nonstationarity can be learned from real data to create more reliable synthetic data. This enables researchers to perform more reliable evaluations of their algorithms, and avoid overestimating their performance on simulated data. Additionally, our approach allows the opportunity for ablation studies where properties can be varied to see how they affect model performance. This provides an avenue for improving how algorithms handle these variations to make them more adaptable to the dynamic nature of real-world data. Future work is needed to incorporate additional statistical properties (e.g., skewness) while learning the models of nonstationarity, as this may better capture the full dynamic nature and complexity of real-world health data.

Acknowledgments

This work was supported in part by the NLM of the NIH under Award Number R01LM011826.

References

Charles K Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, 2022.

Dominik R. Bach, Guillaume Flandin, Karl J. Friston, and Raymond J. Dolan. Modelling event-related skin conductance responses. *International Journal of Psychophysiology*, 75(3):349–356, 2010. ISSN 0167-8760. doi: https://doi.org/10.1016/j.ijpsycho.2010.01.005.

Marília Barandas, Duarte Folgado, Letícia Fernandes, Sara Santos, Mariana Abreu, Patrícia Bota, Hui Liu, Tanja Schultz, and Hugo Gamboa. Tsfel: Time series feature extraction library. *SoftwareX*, 11:100456, 2020.

Alvaro Bartolome. Trendet. https://trendet.readthedocs.io/index.html, 2020.

Anna L Buczak, Steven Babin, and Linda Moniz. Data-driven approach for creating synthetic electronic medical records. *BMC medical informatics and decision making*, 10(1):1–28, 2010.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, page 286–305. PMLR, November 2017. URL https://proceedings.mlr.press/v68/choi17a.html.

Simon Lebech Cichosz and Alexander Arndt Xylander. A conditional generative adversarial network for synthesis of continuous glucose monitoring signals. *Journal of Diabetes Science and Technology*, 16(5):1220–1223, 2021. doi: 10.1177/19322968211014255.

Rachel Cummings, Sara Krehbiel, Yuliia Lut, and Wanrong Zhang. Privately detecting changes in unknown distributions. In *Proceedings of the 37th International Conference on Machine Learning*, page 2227–2237. PMLR, November 2020. URL https://proceedings.mlr.press/v119/cummings20a.html.

Christopher Duckworth, Francis P Chmiel, Dan K Burns, Zlatko D Zlatev, Neil M White, Thomas WV Daniels, Michael Kiuber, and Michael J Boniface. Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during covid-19. *Scientific reports*, 11 (1):23017, 2021.

Zahra Ebrahimzadeh, Min Zheng, Selcuk Karakas, and Samantha Kleinberg. Deep learning for multi-scale changepoint detection in multivariate time series. *arXiv preprint arXiv:1905.06913*, 2019.

Maximilian Ehrhart, Bernd Resch, Clemens Havas, and David Niederseer. A conditional gan for generating time series data for stress detection in wearable physiological sensor data. *Sensors*, 22(16), 2022. ISSN 1424-8220. doi: 10.3390/s22165969. URL https://www.mdpi.com/1424-8220/22/16/5969.

Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10(15), 2022. doi: 10.3390/math10152733. URL https://www.mdpi.com/2227-7390/10/15/2733.

Prerna Garg, Jayasankar Santhosh, Andreas Dengel, and Shoya Ishimaru. Stress detection by machine learning and wearable sensors. In *26th International Conference on Intelligent User Interfaces-Companion*, pages 43–45, 2021.

Louis A. Gomez, Adedolapo Aishat Toye, R. Stanley Hum, and Samantha Kleinberg. Simulating realistic continuous glucose monitor time series by data augmentation. *Journal of Diabetes Science and Technology*, page 19322968231181138, June 2023. ISSN 1932-2968. doi: 10.1177/19322968231181138.

Hadia Hameed and Samantha Kleinberg. Comparing machine learning techniques for blood glucose forecasting using free-living and patient generated data. In *Proceedings of the 5th Machine Learning for Healthcare Conference*, 2020.

Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. URL https://api.semanticscholar.org/CorpusID:5051282.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3 (1):1–9, 2016.

Kenneth Jung and Nigam H. Shah. Implications of non-stationarity on predictive modeling using ehrs. *Journal of Biomedical Informatics*, 58:168–174, 2015. doi: 10.1016/j.jbi.2015.10.006.

Bishesh Khanal, Nicholas Ayache, and Xavier Pennec. Simulating longitudinal brain mris with known volume changes and realistic variations in image intensity. *Frontiers in neuroscience*, 11:132, 2017.

BP Kovatchev, MD Breton, C Dalla Man, and C Cobelli. In silico model and computer simulation environment approximating the human glucose/insulin utilization. *Food and Drug Administration Master File MAF*, 1521:338–346, 2008.

Hanmo Li, Yuedong Wang, and Mengyang Gu. Sequential kalman filter for fast online changepoint detection in longitudinal health records. *arXiv preprint arXiv:2310.18611*, 2023a.

Jin Li, Benjamin J Cairns, Jingsong Li, and Tingting Zhu. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. *NPJ Digital Medicine*, 6(1):98, 2023b.

Kezhi Li, John Daniels, Chengyuan Liu, Pau Herrero, and Pantelis Georgiou. Convolutional recurrent neural networks for glucose prediction. *IEEE journal of biomedical and health informatics*, 24(2): 603–613, 2019.

Georgianna Lin, Rumsha Siddiqui, Zixiong Lin, Joanna M Blodgett, Shwetak N Patel, Khai N Truong, and Alex Mariakakis. Blood glucose variance measured by continuous glucose monitors across the menstrual cycle. *npj Digital Medicine*, 6 (1):140, 2023.

Joseph S Lombardo and Linda J Moniz. Ta method for generation and distribution. *Johns Hopkins APL Technical Digest*, 27(4):356, 2008.

Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and S. H. Annabel Chen. Neurokit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, 53(4):1689–1696, August 2021. ISSN 1554-3528. doi: 10.3758/s13428-020-01516-y.

Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator: New features. *Journal of diabetes science and technology*, Jan 2014. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4454102/.

Cindy Marling and Razvan Bunescu. The ohiot1dm dataset for blood glucose level prediction: Update 2020. In *CEUR Workshop Proc*, page 5, 2020.

Antonia M Marsden, Peter Bower, Elizabeth Howarth, Claudia Soiland-Reyes, Matt Sutton, and Sarah Cotterill. 'finishing the race'–a cohort study of weight and blood glucose change among the first 36,000 patients in a large-scale diabetes prevention programme. *International Journal of Behavioral Nutrition and Physical Activity*, 19(1):1–10, 2022.

Gustav Mårtensson, Daniel Ferreira, Tobias Granberg, Lena Cavallin, Ketil Oppedal, Alessandro Padovani, Irena Rektorova, Laura Bonanni, Matteo Pardini, Milica G Kramberger, et al. The reliability of a deep learning model in clinical out-of-distribution mri data: a multicohort study. *Medical Image Analysis*, 66:101714, 2020.

Scott McLachlan, Kudakwashe Dube, and Thomas Gallagher. Using the caremap with health incidents statistics for generating the realistic synthetic electronic healthcare record. In *2016 IEEE international conference on healthcare informatics (ICHI)*, pages 439–448. IEEE, 2016.

Andreas Melmer, Thomas Züger, Dana M. Lewis, Scott Leibrand, Christoph Stettler, and Markus Laimer. Glycaemic control in individuals with type 1 diabetes using an open source artificial pancreas system (OpenAPS). *Diabetes, Obesity and Metabolism*, 21(10):2333–2337, October 2019.

Charalampos Milionis, Ioannis Ilias, Evangelia Venaki, and Eftychia Koukkou. The effect of menstrual hormonal fluctuations on the glycaemic control in women with type 1 diabetes mellitus. *Practical Diabetes*, 40(4):35–39, 2023.

Richard E Murray, Patrick B Ryan, and Stephanie J Reisinger. Design and validation of a data simulation model for longitudinal healthcare data. In *AMIA Annual Symposium Proceedings*, volume

2011, page 1176. American Medical Informatics Association, 2011.

Bret Nestor, Matthew BA McDermott, Willie Boag, Gabriela Berner, Tristan Naumann, Michael C Hughes, Anna Goldenberg, and Marzyeh Ghassemi. Feature robustness in non-stationary health records: caveats to deployable model performance in common clinical machine learning tasks. In *Machine Learning for Healthcare Conference*, pages 381–405. PMLR, 2019.

Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1), 2018. doi: 10.1038/sdata.2018.178.

Adityo Prakosa, Maxime Sermesant, Hervé Delingette, Stéphanie Marchesseau, Eric Saloux, Pascal Allain, Nicolas Villain, and Nicholas Ayache. Generation of synthetic but visually realistic time series of cardiac images combining a biophysical model and clinical images. *IEEE transactions on medical imaging*, 32(1):99–109, 2012.

Keyvan Rahmani, Rahul Thapa, Peiling Tsou, Satish Casie Chetty, Gina Barnes, Carson Lam, and Chak Foon Tso. Assessing the effects of data drift on the performance of machine learning models used in clinical sepsis prediction. *International Journal of Medical Informatics*, 173:104930, 2023.

Afsane Riazi, John Pickup, and Clare Bradley. Daily stress and glycaemic control in type 1 diabetes: individual differences in magnitude, direction, and timing of stress-reactivity. *Diabetes Research and Clinical Practice*, 66(3):237–244, 2004.

Berkman Sahiner, Weijie Chen, Ravi K Samala, and Nicholas Petrick. Data drift in medical machine learning: implications and potential remedies. *The British Journal of Radiology*, page 20220878, 2023.

Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, page 400–408, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356923. doi: 10.1145/3242969.3242985. URL https://doi.org/10.1145/3242969.3242985.

Xuesheng Shi, Colin Gallagher, Robert Lund, and Rebecca Killick. A comparison of single and multiple changepoint techniques for time series data. *Computational Statistics & Data Analysis*, 170: 107433, 2022.

Gerrit JJ Van den Burg and Christopher KI Williams. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*, 2020.

Floriana Vasile, Anna Vizziello, Natascia Brondino, and Pietro Savazzi. Stress state classification based on deep neural network and electrodermal activity modeling. *Sensors (Basel, Switzerland)*, 23 (5):2504, February 2023. ISSN 1424-8220. doi: 10.3390/s23052504.

Wayne Wakeland and Brahm Goldstein. A review of physiological simulation models of intracranial pressure dynamics. *Computers in biology and medicine*, 38(9):1024–1041, 2008.

Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 2018.

G Dorcas Wambui, Gichuhi Anthony Waititu, and Anthony Wanjoya. The power of the pruned exact linear time (pelt) test in multiple changepoint detection. *American Journal of Theoretical and Applied Statistics*, 4(6):581, 2015.

Malgorzata E Wilinska, Ludovic J Chassin, Carlo L Acerini, Janet M Allen, David B Dunger, and Roman Hovorka. Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes. *Journal of diabetes science and technology*, 4(1):132–144, 2010.

Jinyu Xie. Simglucose v0.2.1. "https://github.com/jxx123/simglucose", 2018. "Accessed: 2021-05-01.

Jennifer Yu, Tina Behrouzi, Kopal Garg, Anna Goldenberg, and Sana Tonekaboni. Dynamic interpretable change point detection for physiological

data analysis. In *Machine Learning for Health (ML4H)*, pages 636–649. PMLR, 2023.

Taiyu Zhu, Kezhi Li, Jianwei Chen, Pau Herrero, and Pantelis Georgiou. Dilated recurrent neural networks for glucose forecasting in type 1 diabetes. *Journal of Healthcare Informatics Research*, 4(3): 308–324, 2020.

Appendix A. Additional Method Details

Table 4: Time series features

	Features
Statistical Features	interquartile range, kurtosis,minimum, maximum, mean, standard deviation, median, skew, variance, median absolute deviation,root mean square, mean absolute deviation, mean absolute deviation.
Temporal Features	area under the curve, mean absolute difference, median absolute difference, median difference, signal distance, negative turning points, neighbourhood peaks,peak to peak distance, centroid, positive turning points, entropy, sum absolute difference, autocorrelation, mean difference

A.1. Time series features

See Table 4 for the statistical and temporal time series features used for predicting the changepoints.

A.2. Overall Workflow

Figure 6 provides an overview of the overall design workflow. Frist we learn models of nonstationarity from real and augment simulated data with this property. We then carry out ML tasks using the resulting augmented simulated data, and evaluate the performance. The ML tasks are glucose forecasting and stress detection for glucose and EDA data respectively.

Appendix B. Additional Glucose Simulation Details

We use a set of input parameters to generate glucose data with the glucose simulator. The input parameters include daily meal and physical events which are generated based on the meal and physical events information in the dataset.

B.1. Generating daily meal events

To simulate the daily meal events, we sample from the distributions of the number of meals, meal sizes, mealtimes, and duration of meals for each dataset. We generate the daily number of meals m by sampling from the distribution of the daily number of meals for each dataset (OHIO, and OpenAPS). To generate the meal sizes for the m meals, we first extract the total carb intake for days where m number of meals was consumed in the data. Then, we fit this into a normal distribution and sample from it to get the total carb intake. After, we select m number of meal sizes from the distribution of the meal sizes in the data, and normalize them so they add up to 1. Finally, we multiply these adjusted meal sizes by the total carb intake to derive the final meal sizes.

For the meal times, If $m \leq 2$, we sample the meal hour from the distribution of the meal times, and the minute from $\mathcal{U}(0, 59)$. If $m > 2$, we assume that 3 of the meals were consumed at standard meal times i.e breakfast (6am - 10am), lunch (11am - 3pm), and dinner (4pm - 8pm), and sample from these time groups for each of the meal. The mealtimes for the remaining $m-3$ meals are samples using the approach used for when $m \leq 2$. For the duration of the meals, we sample from $\mathcal{N}(45, 15)$ and set the limit to 1 minute and 90 minutes for each meal.

B.2. Generating daily physical activity events.

We simulate the activity as a step increase in the heart rate for a duration of time as in (Man et al., 2014). We assume that there are three physical activity periods. We perform Bernoulli trials with a probability of occurrence of 0.5 to generate the number of activity periods for each day. For each period, we sample the hour of the activity, the minute, and the increase in heart rate from $\mathcal{U}(0, 23)$, $\mathcal{U}(0, 59)$, $\mathcal{N}(45, 100)$ respectively.

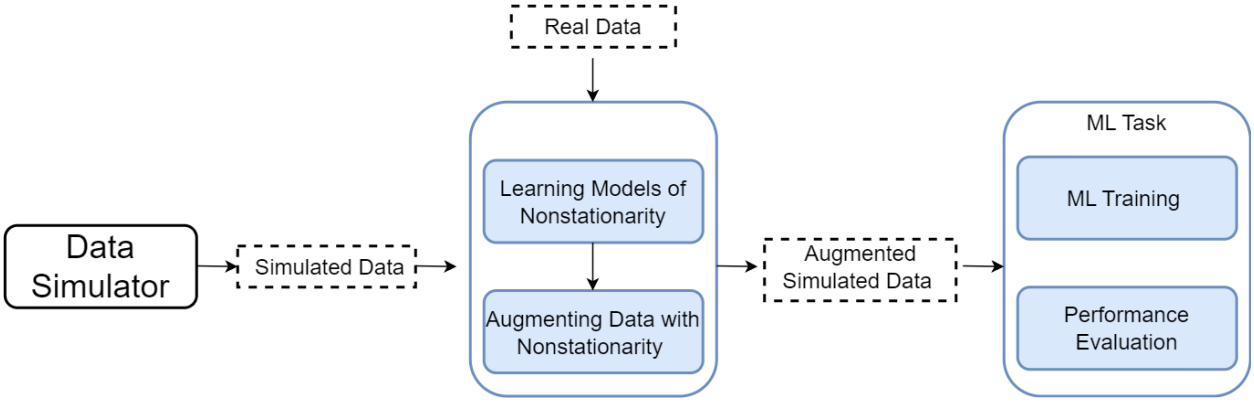


Figure 6: An overview of the overall workflow.

Appendix C. Additional Experimental Results

We conduct additional experiments to evaluate the performance of models trained on the various simulated datasets and tested on the real dataset. Table 5 and Table 6 shows the results for glucose forecasting and stress classification respectively.

Table 5: Mean RMSE results when training on the simulated dataset and testing on real dataset for glucose forecasting. The OHIO and OpenAPS are the test data.

	OHIO				**OpenAPS**			
Simulated Dataset (Train)	REG	RF	RNN	LSTM	REG	RF	RNN	LSTM
Raw Simulated	125.14	29.15	48.23	29.74	90.88	33.27	55.40	37.81
MeanShift-Constant	34.44	27.38	30.07	27.86	28.59	24.22	24.11	23.46
MeanSDShift-Constant	24.77	27.55	24.89	28.09	21.55	26.50	23.46	24.22
MeanShift-Varying	30.56	27.27	28.99	27.02	25.14	23.29	22.86	22.44
MeanSDShift-Varying	24.92	25.35	24.62	24.49	21.57	22.72	22.56	22.85
Our Method	25.87	27.59	25.50	27.55	26.04	24.53	26.22	24.09
Real Data	24.56	23.06	24.30	23.48	20.29	19.82	19.41	20.05

Table 6: Performance on real and simulated data for stress prediction when training on the simulated dataset and testing on real dataset.

	KNN		**LR**		**RF**		**SVM**	
Simulated Dataset (Train)	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
Raw Simulated	0.74	0.54	0.74	0.54	0.76	0.61	0.74	0.53
MeanShift-Constant	0.74	0.53	0.74	0.53	0.75	0.57	0.73	0.53
MeanSDShift-Constant	0.69	0.31	0.70	0.28	0.71	0.48	0.70	0.29
MeanShift-Varying	0.74	0.53	0.73	0.53	0.75	0.55	0.73	0.51
MeanSDShift-Varying	0.71	0.47	0.70	0.18	0.73	0.53	0.69	0.43
Our Method	0.74	0.57	0.80	0.72	0.64	0.48	0.64	0.14
Real Data	0.69	0.69	0.70	0.71	0.67	0.68	0.73	0.72