The University of Auckland
Department of Engineering Science
Part IV Project Report

# Understanding the Spatiotemporal Organisation of GI Bioelectrical Activity at the Microscale

*Author:*
Kartikey Vyas

*Supervisors:*
Dr. Peng Du
Dr. Andreas W. Kempa-Liehr

October 2020

# Declaration of Authorship

I, Kartikey Vyas, declare that this report titled "Understanding the Spatiotemporal Organisation of GI Bioelectrical Activity at the Microscale" and the work presented in it are my own.

This project was carried out under the guidance of my supervisors Dr Peng Du and Dr Andreas Kempa-Liehr at the University of Auckland during the 2020 academic year. This project was completed as part of the ENGSCI 700 A/B course, and under the Department of Engineering Science in the Faculty of Engineering.

Published material that I have consulted has been referenced where appropriate. The work of my partner, Louis Lin, was almost completely independent of that of my own except for one figure, for which he been credited within this report. All other work presented has been my own contribution.

Signed:

$30^{th}$ October, 2020.

# Abstract

The treatment and diagnosis of gastrointestinal (GI) motility disorders remains a significant and costly problem for current clinical practice. GI motility is driven by the rhythmic contraction and relaxation of smooth muscles. This phenomenon is governed by bioelectrical activity known as slow waves. Dysrhythmia and abnormalities in slow waves are often the cause behind GI motility disorders and are treated with certain drugs. The effect of these drugs on the spatiotemporal propagation of slow waves is not well characterised. The main aim of this project was to explore the applications of machine learing (ML) on this problem. The available data consisted of GI slow wave potentials recorded from a 60 electrode Micro-electrode Array (MEA). Automated times series feature extraction and selection, implemented through the FRESH (FeaturRe Extraction based on Scalable Hypothesis tests) algorithm, was conducted. It was found that classifiers could reliably predict whether a subject was administered Atropine (AT) or Hexamethonium (Hex) subsequent to the administering of ACh (Acetylcholine). The most important features were explored. GI slow waves from subjects under the effect of AT consistently had higher values for each of the most important features indicating that AT has an excitatory effect on pacemaker potentials when administered after ACh, while Hex does not. For example, AT influenced slow waves had higher frequency and noise levels, on average containing 4 to 10 peaks for every 6 second recording, while Hex influenced slow waves had only up to 3 peaks for the same length of recording. Overall, a systematic framework has been established for characterising GI slow waves through machine learning that avoids laborious feature engineering.

# Acknowledgements

Firstly, I would like to express my gratitude to my exceptional supervisors Dr. Peng Du and Dr. Andreas Kempa-Liehr for their continued support and guidance over the course of this project. In particular, I would like to thank Dr. Andreas Kempa-Liehr for his patience and wisdom when guiding me through the process of making an open source contribution. I also greatly appreciated both supervisors for their flexibility and willingness to adapt to what was a tumultuous year for everyone. They both continue to inspire me to strive for excellence. To my friend and project partner Louis, your support throughout this project has been invaluable. I have enjoyed spending this year with you immensely and I am in awe of the meticulous work you produce.

# Contents

# List of Figures

# List of Tables

# Symbols and Acronyms

| Abbreviation | Definition |
|---|---|
| MEA | Microelectrode Array |
| GI | Gastrointestinal |
| ICC | Interstitial Cells of Cajal |
| EGG | Electrogastrography |
| ECG | Electrocardiogram |
| EEG | Electroencephalogram |
| ML | Machine Learning |
| ACh | Acetylcholine |
| AT | Atropine |
| Hex | Hexamethonium |
| FRESH | Feature Extraction based on Scalable Hypothesis tests |
| PCA | Principal Component Analysis |
| HPC | High Performance Computing |
| NeSI | New Zealand eScience Infrastructure |
| SCP | Secure Copy |
| SSH | Secure Shell |
| OvR | One versus Rest |
| CWT | Continuous Wavelet Transform |
| CID | Complexity-Invariant Distance |

# Open Source Contributions

As a part of this project, an open source contribution was made to the `tsfresh` library. This involved integrating the multiclass feature selection methodology proposed by Tang, Blincoe and Kempa-Liehr into the existing feature selection code. This contribututION was made through a pull request submitted to the `tsfresh` GitHub repository, which was accepted and merged into the main branch on the $28^{th}$ of October, 2020. The contribution is currently accessible at `https://github.com/blue-yonder/tsfresh/pull/762`.

# 1 Introduction

The gastrointestinal (GI) system is responsible for digesting and absorbing ingested food and liquids, with contents being moved through the tract by the contraction and relaxation of smooth muscles [1]. One of the major functions of the GI tract is motility, which follows distinct patterns depending on the stage of digestion. Smooth muscles could be working to move and mix food to ensure sufficient contact with digestive tissue or to clear any debris from the GI tract [1]. Disorders in GI tract motility are common and thus to better understand how to treat and prevent them, the mechanisms governing motility need to be understood.

The GI tract exhibits rhythmic contraction and relaxation of smooth muscles. This behaviour is driven by a series of electrical potentials, known as 'slow waves', which are generated by interstitial cells of Cajal (ICC) [2]. Abnormalities in GI slow waves can lead to dysrhythmia, similarly to the electrical abnormalities seen in other organs like the brain and heart. Cardiac dysrhythmia have been well characterised due to their life-threatening nature, in contrast there has been a relative lack of analysis of GI dysrhythmia. However, GI disorder symptoms are widespread, with over 60% of US adults reporting symptoms in a week in 2018 [3], causing a financial burden of over $135 billion dollars annually [4].

There are a number of drugs that could alter GI slow waves via influencing the cellular receptors that mediate slow wave generation [5]. However, the exact spatiotemporal effects of the drugs have not been thoroughly investigated so their effects on GI slow waves in intact tissues are currently not well understood. To better understand the effects of drugs, the capture and analysis of slow waves needs to be conducted. GI slow waves have been monitored non-invasively using cutaneous electrogastrography (EGG), but these recordings have severely limited spatial resolution. This is due to discontinuities in conductivity created by fat and muscle layers in the walls of the abdomen [6]. Microelectrode arrays (MEAs) placed directly on ileum serosa have been demonstrated to be an effective method for the investigation of slow wave frequency and spatial propagation [7]. Directional propagation is especially important in classifying the physiological state of the stomach, for example retching or emesis. Furthermore, characterising the effects of different drugs on slow waves will provide insight into their effectiveness.

Machine learning (ML) based classification of electrocardiogram (ECG) and electroencephalogram (EEG) signals plays an important role in the diagnosis of diseases, and similar applications of ML on GI signals have not been as thoroughly explored. Existing applications of ML in the classification of GI slow waves have considered a limited number of features and do not characterise the effects of drugs [8, 9]. The high spatial and temporal resolution of the data produced from MEAs presents an opportunity for the application of ML algorithms to classify GI signals and subsequently predict drug effects and physiological responses.

Building from the MEA study conducted by Liu et al., a data set containing recordings of GI slow waves from a 60-electrode array is to be explored. This data set has measurements from multiple subjects under the effects of multiple drugs, namely acetylcholine (ACh) followed by atropine (AT) or hexamethonium (Hex). This project aims to investigate the applicability of ML techniques in the classification of slow waves recorded under varying drug effects. Furthermore, it will be investigated how ML techniques can further characterise the effects of ACh and the subsequent administering of AT or Hex.

## 1.1 Aims

The primary objective of this project was to assess the applicability of ML techniques in characterising the spatiotemporal effects of drugs on GI slow waves. The following aims were devised to address this objective:

- To set up a data processing pipeline to extract time series features from raw MEA data for classification.
- To build an ML model to distinguish the drug administered using extracted features from MEA time series data.
- To investigate the important features that characterise the effects of drugs.
- To verify the effectiveness of drugs through their effect on slow waves.

## 1.2 Background

### 1.2.1 Electrophysiology of the Gastrointestinal Tract

GI slow waves are responsible for the phasic contraction of smooth muscles, being generated and propagated by a network of interstitial cells of Cajal (ICC) in the presence of neurotransmitters [1, 2]. ICC serve as pacemakers in GI muscles, being responsible for generating slow waves through the uptake and release of calcium ions [2]. Normal GI tissues generate slow waves synchronously at a common frequency, determining the frequency and propagation of muscular contractions. Disruptions to this pattern are the dysrhythmia that have been suspected to lead to disorders such as dyspepsia, gastroparesis and unexplained nausea and vomiting [10]. Figure 1 displays the one-to-one relationship between slow waves and muscular contractions.

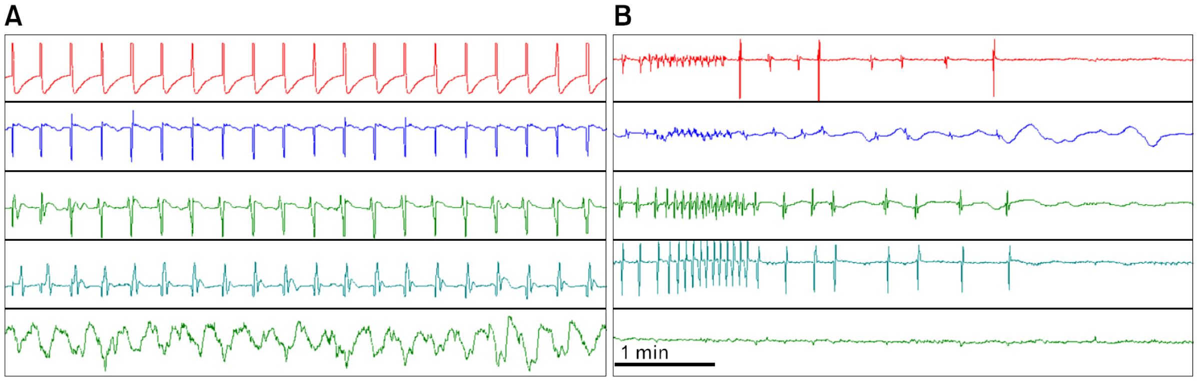


Figure 1: Correlation between slow waves and gastric contractions [11]. The top four channels display serosal myoelectrical recordings, while the bottom channel shows gastric contractions as measured from strain gauges placed on gastric serosa. (A) Normal slow wave activity. (B) Spontaneous gastric dysrhythmia and hypomotility.

Recently, high density MEAs have been demonstrated to produce high resolution mappings of GI electrical activity, providing a platform for improved dysrhythmia research [10]. MEA recordings are not vulnerable to the same levels of interference seen in EGG, as they are placed directly on GI tract tissue. Studies using a similar high-resolution mapping in patients with gastroparesis and chronic unexplained nausea and vomiting, have now demonstrated that spatially-complex dysrhythmia are prevalent in these functional GI disorders [12]. In the recent MEA study by Liu et al., MEA was used to find slow wave frequency, amplitude, propagation velocity and directionality [7]. The large amount of detail available in this data presents the opportunity to further investigate dysrhythmia, as well as their interaction with drug exposure.

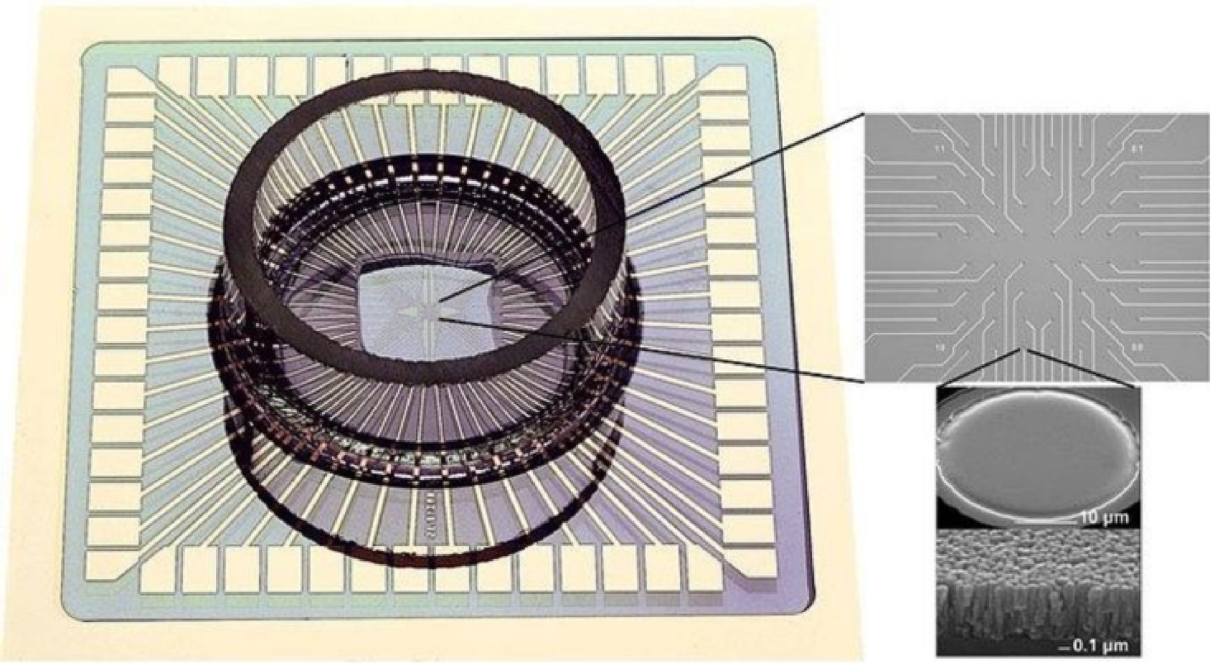


Figure 2: Image of a Microelectrode Array, similar to the one used in [7]

### 1.2.2 Drug Effects on Slow Waves and Gastrointestinal Function

The enteric nervous system, the intrinsic nervous system of the GI tract, releases the neurotransmitter acetylcholine (ACh) , which regulates slow wave frequencies. Under higher concentrations of ACh, the frequency of slow waves in the stomach increases [13]. The level of increase in different parts of the GI tract is different and is at odds with the frequency gradient observed in normal GI tracts [13], which therefore increases the chance of slow wave dysrhythmia. Atropine (AT) is a muscarinic antagonist, an agent which blocks muscarinic ACh receptors, and it has been shown to counteract the effects of ACh on smooth muscle contraction. Therefore, AT cancels out the increase in slow wave frequency caused by ACh to some extent [14, 15]. Hexamethonium (Hex) inhibits nicotinic ACh receptors, having a similar effect to AT, but has been not shown to counteract the inhibitory actions of ACh on pacemaker potentials [16].

### 1.2.3 Time Series Classification

A time series is defined as temporally variant information with regularly updated values, such as, in our case, the measurements of electrodes in an MEA [17]. This can be expressed in the following notation, where $s_i(t_j)$ is the state of sensor $i$ at time step $j$.

$$S_i = (s_i(t_1), s_i(t_2), ..., s_i(t_n))^T$$

Time series classification is the problem where time series are to be assigned discrete categories. We are given a target vector $Y = (y_1, ..., y_m)^T$ where each entry $y_i$ describes the category to which the time series $S_i$ belongs. An ML model creates a mapping that can predict the values in $Y$ using the different time series $S = (S_1, S_2, ..., S_m)$ as inputs [17].

Central to the problem of time series classification where signals are not well characterised is feature engineering. In classification problems where each label is associated with multiple time series, the volume of data makes characterisation and predictive modelling difficult [17, 18]. As such, to characterise a time series, several *features*, measurable attributes of a given time series, should be computed [18]. One example of a time series feature $X$ could be the minimum, which is computed by applying the minimum operator to a time series.

$$X_{min}(S_i) = \min\{s_{i,1}, s_{i,2}, ..., s_{i,n}\}$$

An existing algorithm termed Feature Extraction based on Scalable Hypothesis tests (FRESH), comprises comprehensive, automated feature extraction and selection [17]. First, a comprehensive set of time series features are extracted through well-established mappings. Next, each feature is tested for its statistical significance and adjustments are made to account for the accumulation of statistical error. This process is used to select relevant features that can then form a *feature matrix* that acts as the input to a machine learning classifier. This procedure can be applied to the signals recorded from the MEA in Liu et. al's 2020 study [16] to allow the application of ML techniques to the data.

### 1.2.4 Bioelectrical Signal Classification

Dysrhythmia in bioelectrical activity have long been investigated as causes of severe disorders in organs such as the brain. Different aberrations lead to different disorders, which makes the ability to classify bioelectrical signals highly valuable in early detection and diagnosis. Due to the lack of specific ML applications on GI slow waves, the classification of brain signals recorded through electroencephalograms (EEGs) was initially used as a reference to demonstrate the workflow and techniques that could be utilised in this project.

EEG employs multiple electrodes that are uniformly arrayed on the scalp, thereby capturing the spatiotemporal properties of the brain's electrical activity. ML has been applied on these properties for the prediction of epileptic seizures in numerous studies [19, 20, 21, 22]. Seizure onset is characterised by hidden dynamical patterns several minutes before their electro-clinical onset in patients with partial epilepsy [21]. The high number of relevant features that incorporate spatiotemporal context in addition to signal morphology has driven the exploration of ML techniques in this field, as traditional statistical methods cannot be used to analyse such high-dimensional problems [20]. Automated feature extraction with FRESH generates a high number of features, of which some may be relevant when applied to GI slow waves.

In earlier EEG papers involving ML, a variety of feature extraction and classification methods had been employed, but in recent papers we see a convergence of methodology to principal component analysis (PCA) and support vector machines (SVM). The feature extraction approaches find correlation patterns that are derived from the principal components of the correlation and covariance matrices of different EEG channels [20]. SVMs learn nonlinear decision boundaries from training set features, and have been chosen in this field due to their robustness for estimating predictive models from noisy, sparse and high-dimensional data [23]. In the most recent seizure detection study examined, an SVM produced binary classification accuracy of up to 82% [24].

### 1.2.5 Machine Learning Tools

The Python programming language is host to some of the most widely used data analysis and ML libraries, `pandas` [25] and scikit-learn (`sklearn`) [26] respectively. The `pandas` library contains the key data structures of `DataFrame`, `Series` and `Index`, which serve as the foundation for data analysis in Python [25]. `DataFrame` objects are essentially labelled matrices which can contain raw time series or time series features. The ML library `sklearn` contains a comprehensive array of implementations of ML algorithms, cross-validation schemes and data transformers [26]. `sklearn` has native support for `pandas` objects. The FRESH algorithm has a Python-based implementation in the form of the library `tsfresh`, which is able to be fully integrated with objects and methods from `pandas` and `sklearn` [27]. These key libraries will be the primary tools of analysis in this project.

| Subject ID | First Drug | Second Drug | Duration (s) | Sampling Rate |
|---|---|---|---|---|
| 00_0315 | ACh | AT | 540 | 1000 Hz |
| 02_0315 | ACh | AT | 540 | 1000 Hz |
| 04_0316 | ACh | AT | 540 | 1000 Hz |
| 05_0316 | ACh | AT | 540 | 1000 Hz |
| 06_0317 | ACh | AT | 540 | 1000 Hz |
| 01_0126 | ACh | Hex | 540 | 1000 Hz |
| 02_0126 | ACh | Hex | 540 | 1000 Hz |
| 03_0126 | ACh | Hex | 540 | 1000 Hz |
| 05_0201 | ACh | Hex | 540 | 1000 Hz |
| 06_0201 | ACh | Hex | 540 | 1000 Hz |
| 08_0201 | ACh | Hex | 540 | 1000 Hz |

Table 1: Format of the raw MEA data. For each subject, there were three files available, each consisting of 180 s of GI slow wave recordings under the effect of a different drug. The MEA had a sampling rate of 1000 Hz, meaning that each file had 180,000 readings for each electrode.

# 2 Methods

## 2.1 Data Processing

The first stage of building an ML model is processing the raw data. We will explore the data set that was used in the MEA drug effects study conducted by Liu et. al. [16].

### 2.1.1 Raw Data

A single 60 electrode MEA was placed directly on 1cm segments of Ileum taken from two ICR mice. The drugs were administered by introducing them directly to the organ bath which the subjects were placed in. The MEA had an $8 \times 8$ configuration, with electrodes spaced at intervals of 200 µm over an area of 1.96 mm$^2$. Table 1 describes the data that was recorded from 11 subjects.

For each subject, there was 180 s of baseline recording, 180 s of first drug and 180 s of second drug. Recordings were trimmed such that the organ bath was allowed to stabilise between the introduction drugs, removing any drug administration artefacts. Each recording contained 60 signals, one for each electrode, sampled at 1000 Hz. In total, there were 5.94 million samples, totalling to 3.564 billion data points.

Upon exploration of the raw data, a 2D visualisation was produced using a MATLAB developed for another project which utilised this data set. These visualisations (3) show that there are patches of electrodes which show inconsistent activation with the rest of the MEA. This could be a potential issue when attempting to classify signals, but the feature selection methods used later in this investigation should be robust enough to disregard outlier signals.

### 2.1.2 Data Preparation

The raw data were originally stored in MAT-files that occupied approximately 2.53 GB of disk space. The primary tool for analysis in this investigation was Python, where the data were imported and transformed. Each recording was split into six-second time windows

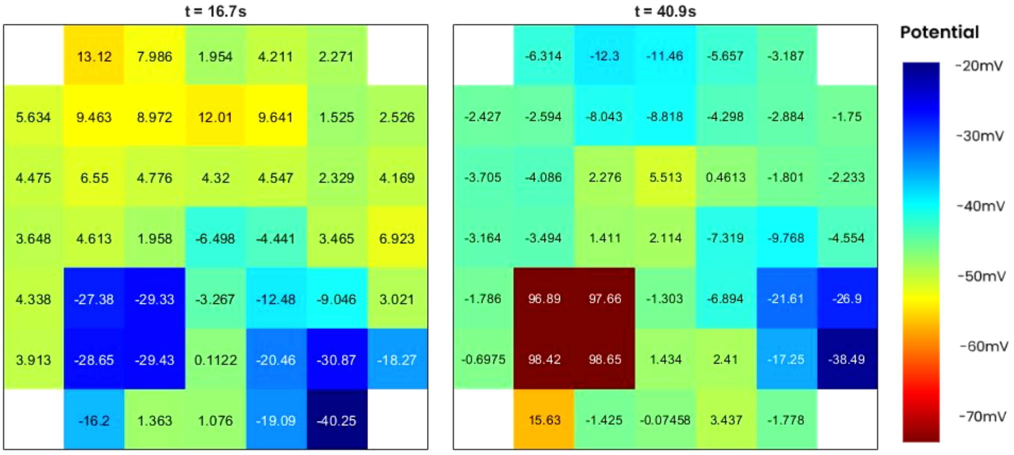


Figure 3: 2D activation of the MEA. The visualised activations are from the baseline recording for subject 05_0316. The contrast in potential in the patch of electrodes near the bottom left of the MEA raise some concerns about the accuracy of the recordings. This figure comes at the courtesy of Louis Lin (project partner).

(6000 time steps). This value was chosen as it is long enough to capture at least one cycle of slow waves whilst ensuring there are as many separate observations as possible.

Since Python was the primary tool for analysis, the scientific Python I/O module `scipy.io` was used to interface with MAT-files. Each recording was in the form of a 2D numeric array, with each row corresponding to one of the 60 electrodes in the MEA. These arrays were transformed, labelled and combined into a single `DataFrame` object. The resulting `DataFrame` had 5.94 million rows, 990 discrete time windows and 60 columns. This format, illustrated in Figure 4 is suitable for automated feature extraction.

$$
\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ 60 \end{array}
\begin{array}{c}
\begin{array}{cccc} t_1 & t_2 & \cdots & t_n \end{array} \\
\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & \ddots & & \\ \vdots & & \ddots & \\ x_{60,1} & & & x_{60,n} \end{pmatrix}
\end{array}
\quad (\times 33)
$$

$$\downarrow$$

| t | id | 1 | 2 | ... | 60 | y |
|---|---|---|---|---|---|---|
| $t_1$ | 1 | $x_{1,1}$ | $x_{2,1}$ | ... | $x_{60,1}$ | 'baseline' |
| $t_2$ | 1 | $\vdots$ | $\ddots$ | | | 'baseline' |
| $\vdots$ | $\vdots$ | $\vdots$ | | $\ddots$ | | $\vdots$ |
| $t_N$ | 990 | $x_{1,N}$ | | | $x_{60,N}$ | 'hex' |

Figure 4: Raw and Processed Data Formats. Here, $n$ represents the number of time steps in a single recording (180,000) and $N$ is the total number of time steps (5,940,000).

Electrode readings from four different time windows show large variability between different electrodes, but the mean signals illustrate some distinctions between the samples examined, as shown in Figure 5.

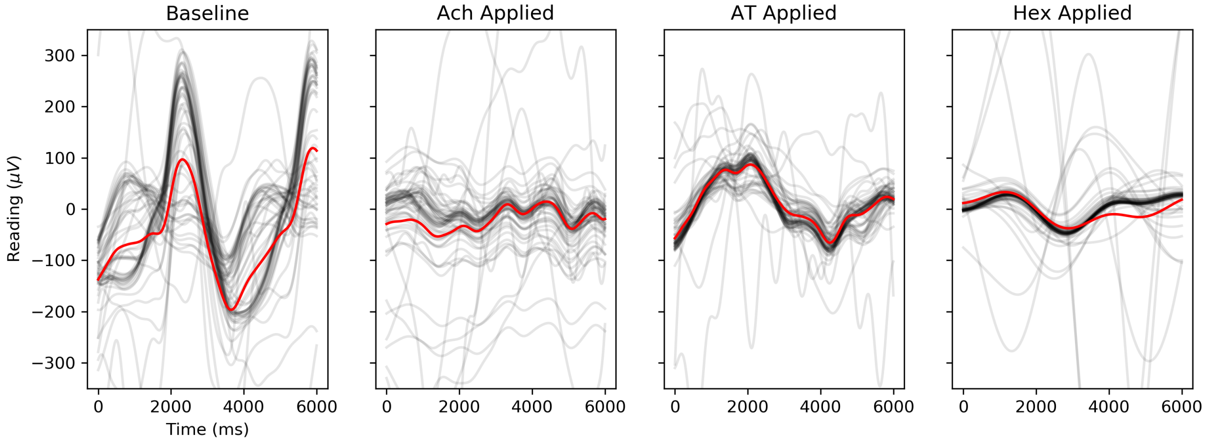


Figure 5: Plot of four windows of MEA data. Each black line represents the reading from one of the electrodes, while the red line represents the mean signal across the MEA for that time window. The baseline, ACh and AT windows were retrieved from subject 00_0315, the Hex window was retrieved from subject 01_0126.

## 2.2 Time Series Engineering

In order to capture the spatial characteristics of slow wave propagation, the creation of 'virtual' signals from the combination of signals from different electrodes was considered [28]. Differences between readings on adjacent electrodes were computed, adding another 68 time series to the existing 60. Given that electrodes are a fixed distance apart from each other, relative differences in their readings should provide some insight into the spatial propagation of GI slow waves. Additionally, the mean signal from the MEA during each time window was computed as another virtual signal.

## 2.3 Automated Feature Extraction

The FRESH algorithm is implemented in a Python-based machine learning library called `tsfresh` [27]. Feature extraction in `tsfresh` computes a total of 794 time series features based on 63 time series characterisation methods.

### 2.3.1 Parallelised Computation of Features

Feature extraction with `tsfresh` has a high computational cost, with significant memory requirements that make it unfeasible to run extraction on the MEA data on a local machine. As such, the high performance computing (HPC) cluster 'Mahuika', provided by New Zealand eScience Infrastructure (NeSI) was utilised [29]. `tsfresh` supports parallelisation through the `multiprocessing` Python library, which leads to significant improvements in run time for the feature extraction process when employing multiple CPU cores [27]. In addition to built-in parallelisation of feature extraction on a single time series, feature extraction on the MEA data was trivially separated into 129 jobs, one for each time series. The workflow and architecture of this process is shown in Figure 6.

A Python script `extract_features.py` was developed which takes command line arguments that specify which time series to run feature extraction on. The Mahuika cluster uses the *Slurm* workload manager to schedule computation jobs and optimise resource usage [30]. Slurm supports the use of 'job arrays' which offer a mechanism for

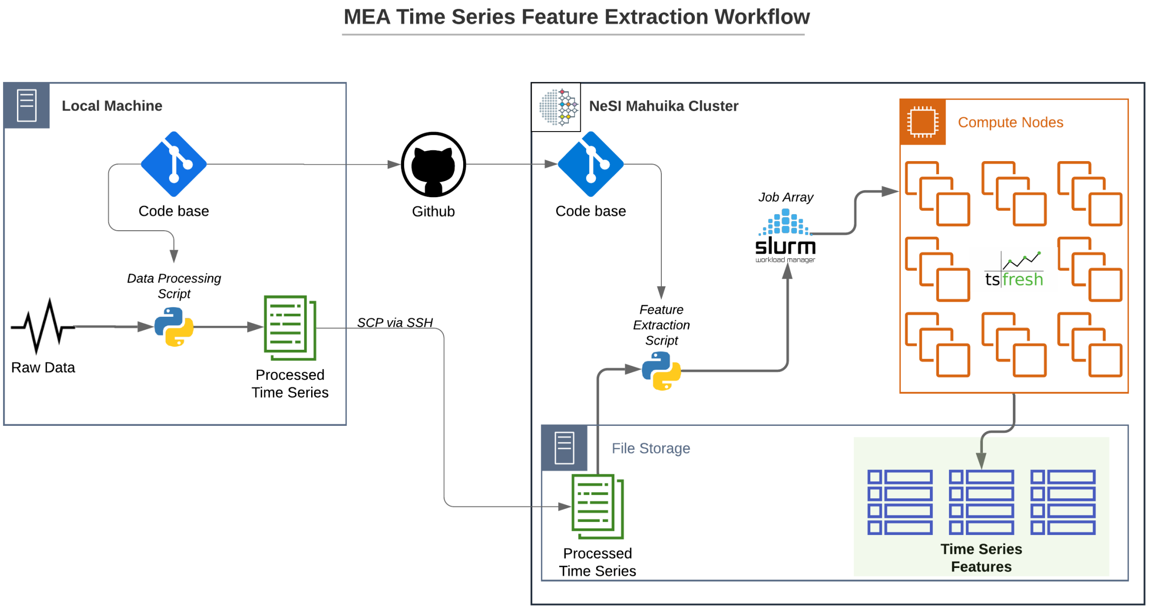


Figure 6: Feature extraction architecture and workflow. The raw MEA data was processed into six second time windows locally, then sent to the Mahuika cluster through secure copy (SCP) via a secure shell (SSH) connection. GitHub was used for code storage and version control, which allowed Python feature extraction scripts to be sent to the cluster. Slurm scripts facilitated the deployment of 129 feature extraction jobs on the compute nodes.

submitting multiple, similar jobs simultaneously [30]. Using a length 129 job array with the `extract_features.py`, feature extraction was executed on each time series separately.

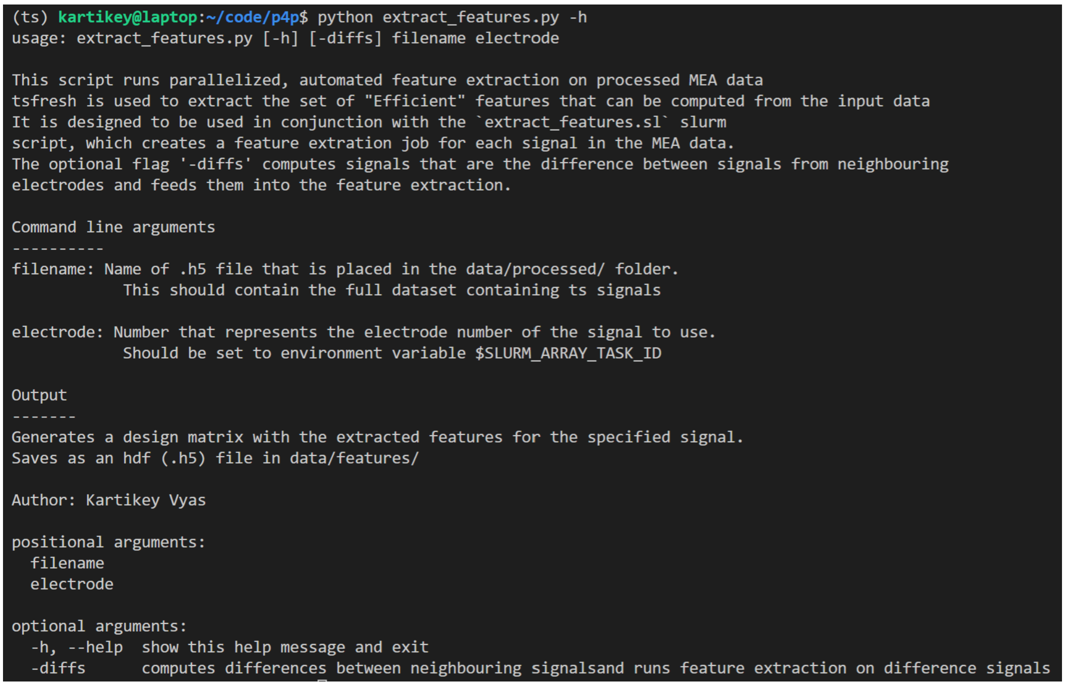
Each job was allocated 16 GB of RAM and 16 CPU cores. The average elapsed time for extracting features from each time series was approximately 20 minutes.

### 2.3.2 Extracted Features

Due to the large amount of data from the MEA, features which have a particularly high computational cost were omitted from feature extraction by specifying the `EfficientFCParameters` class under feature extraction settings [31]. 750 features were calculated for each time series, resulting in a total of 97,650 features for 990 observations. Extracted feature types include common summary statistics, as well as attributes of certain transformations of the time series data such as fourier transforms and ricker wavelet transforms, among others.

## 2.4 Feature Selection

The large number of features generated needed to be filtered according to the FRESH algorithm. `tsfresh` provides an implementation of the feature selection process that can be applied to binary classification problems. There are two key steps in feature selection; hypothesis testing and false discovery rate correction [17, 27].



```
(ts) kartikey@laptop:~/code/p4p$ python extract_features.py -h
usage: extract_features.py [-h] [-diffs] filename electrode

This script runs parallelized, automated feature extraction on processed MEA data
tsfresh is used to extract the set of "Efficient" features that can be computed from the input data
It is designed to be used in conjunction with the `extract_features.sl` slurm
script, which creates a feature extration job for each signal in the MEA data.
The optional flag '-diffs' computes signals that are the difference between signals from neighbouring
electrodes and feeds them into the feature extraction.

Command line arguments
----------
filename: Name of .h5 file that is placed in the data/processed/ folder.
           This should contain the full dataset containing ts signals

electrode: Number that represents the electrode number of the signal to use.
           Should be set to environment variable $SLURM_ARRAY_TASK_ID

Output
-------
Generates a design matrix with the extracted features for the specified signal.
Saves as an hdf (.h5) file in data/features/

Author: Kartikey Vyas

positional arguments:
  filename
  electrode

optional arguments:
  -h, --help  show this help message and exit
  -diffs      computes differences between neighbouring signalsand runs feature extraction on difference signals
```

Figure 7: This documentation is automatically inferred from the configuration of the command line interface using the package `argparse` for `extract_features.py`. Similar documentation has been prepared for all other scripts that accept command line arguments.

### 2.4.1 Hypothesis Testing

When testing the relevance of real-valued features for predicting a binary target, `tsfresh` uses the Mann-Whitney U test [27]. The null hypothesis Mann-Whitney U test stipulates that two samples come from the same population, meaning that the two groups have the same distribution [32]. In the context of feature selection, the following hypotheses are tested for each feature $X_\phi$ [17].

$$H_0^\phi = \{X_\phi \text{ is irrelevant for predicting } Y\}, \; H_1^\phi = \{X_\phi \text{ is relevant for predicting } Y\}.$$

From each test, a p-value $p_\phi$ is retrieved which serves to quantify the probability that the feature $X_\phi$ is relevant for predicting $Y$. In this context, a wrongly selected feature is one for which the null hypothesis has been rejected when it should not have been. For an individual feature, the risk of this occurring is quantified by the p-value, but when comparing multiple hypotheses and features simultaneously, inferential errors can accumulate [17]. FRESH controls this through a measure called the 'False Discovery Rate' (FDR) by applying the Benjamini-Yekutieli procedure [17]. FDR represents the overall proportion of null hypotheses that are incorrectly rejected. Reducing this measure results in fewer features being selected, as illustrated in Figure 8. For the remainder of this investigation, an FDR value of 0.05 was used.

Using an FDR of 0.05, the maximum acceptable p-value was approximately $7 \times 10^{-4}$. Decreasing FDR to 0.001, the maximum acceptable p-value fell to $7 \times 10^{-6}$. Increasing the FDR to 0.1, the maximum acceptable p-value rose to $1 \times 10^{-3}$.

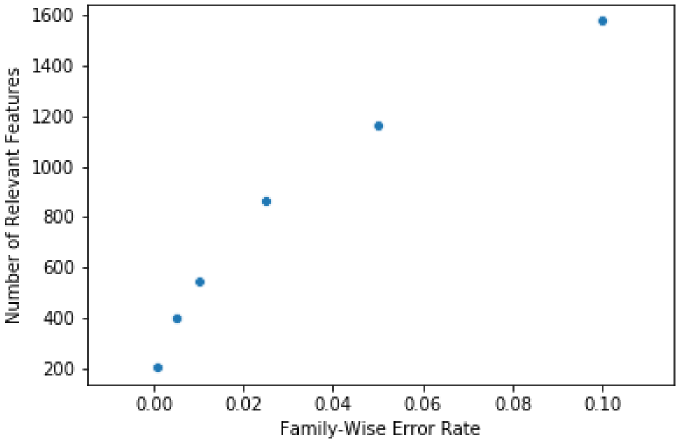


Figure 8: Plots the number of features selected as the FDR level in the Benjamini-Yekutieli procedure was varied. The set of features considered here are a subset of those extracted from the MEA data. Hypothesis testing was conducted against a binary target vector with classes ‘baseline’ and ‘drug’.

The `tsfresh` implementation of this process in the feature_selection sub-module uses the `multiprocessing` library to facilitate local parallelisation [27].

### 2.4.2 Multiclass Feature Selection

Among the classification problems considered, several were multiclass, where the target vector contains more than two unique categories. For example, a model was built that aimed to distinguish GI slow waves from subjects not under the effect of any drugs, subjects under the effect of ACh and subjects under the effect of AT (administered after ACh). `tsfresh` is limited in that its feature selection methods for classification problems assume a binary target variable [31]. As such, modifications and additions to the open source `tsfresh` code base were proposed that allowed for a multiclass feature selection implementation.

**Methodology**

The multiclass implementation involves conducting hypothesis tests multiple times for each feature, depending on the number of distinct classes in the target variable. This is done on a ‘one vs. rest’ (OvR) basis, following the approach described in Tang et. al. [33]. If the target variable $Y$ contains $n$ distinct classes $y_1, ..., y_n$, the *binary* target variables $Y^{\text{bin}}$ are tested for each feature $X_\phi$ and are defined as follows.

$$Y_i^{\text{bin}} = \begin{cases} 1 \text{ if } Y = y_i, \\ 0 \text{ if } Y \neq y_i, \end{cases} \quad \text{for } i \in \{1, 2, ..., n\}.$$

The p-values of the hypothesis tests for each class for each feature are saved and corrected by the Benjamini-Yekutieli procedure, which then determines which features are relevant for each class. Features are then deemed relevant for the *multiclass* problem if they are relevant for at least $k$ classes. This feature selection method provides a more stringent selection criteria for multiclass problems than the existing implementation, thereby further restricting the set of relevant features that are extracted by `tsfresh`.

With the high number of features extracted from the MEA data, this implementation provides a more effective filter to remove irrelevant features before fitting a model.

**Implementation**
A forked version of the `tsfresh` repository was installed locally, inside which the proposed changes were implemented. Functions within the 'feature_selection' and 'transformers' sub-modules were amended. The 'transformers' sub-module contains the `sklearn` pipeline compatible class `FeatureSelector` which implements the feature selection process and incorporates all the changes made. The following new arguments were introduced in this transformer:

| Name | Type | Description |
|---|---|---|
| `multiclass` | boolean | Toggles whether the multiclass implementation will be used or not |
| `n_significant` | integer | Sets the minimum number of classes $k$ for which a feature must be a relevant predictor in order to be selected |
| `multiclass_p_values` | string | The desired method for choosing how to display multiclass p-values for each feature. Either ''avg'', ''max'', ''min'', ''all''. Defaults to ''min'', meaning the p-value with the highest significance is chosen. When set to ''all'', the attributes 'self.feature_importances_' and 'self.p_values' are of type pandas.DataFrame, where each column corresponds to a target class |

Table 2: These parameters were introduced to control the multiclass feature selection process. They became additional arguments for the `calculate_relevance_table`, `select_features` and `FeatureSelector` functions/classes.

Several unit tests were written to ensure code quality, with current test coverage at 93.58%. Furthermore, the highly parallel nature of the feature selection process was preserved, with no changes made to how the `multiprocessing` library is called. This open-source contribution was a key part of implementing feature selection for the various classification problems considered. The outcomes of its use will be further discussed in the Section 3 of this report. As of the 28th of October, 2020, the proposed changes have been accepted and merged into the `tsfresh` GitHub repository [34].

## 2.5 Machine Learning

Using `sklearn`, a pipeline was built that incorporated the feature selector transformer from Section 2.4.2 with a random forest classifier. A grouped cross-validation scheme was used to evaluate the model and tune hyperparameters to optimise F1 score.

### 2.5.1 Random Forests

Random forests are a type of *ensemble* model that combine multiple decision trees and allow them to vote for the most popular predicted class [35]. This model type was chosen for the GI slow wave classification task for several reasons. Firstly, random forests work well with high dimensional problems and tend to be robust to outliers [36, 37]. This is very important for this task since there are 96,750 features considered and visual inspection of the raw data has shown a high amount of variability. The effectiveness of random forest classifiers in feature-based time series classification is further highlighted by their use in testing the FRESH algorithm [17]. Secondly, random forests provide a simple way to identify the relative importance of features through the 'Gini impurity' measure [35]. This is especially important for characterising the effects of drugs on GI slow waves; identifying important features will serve as the basis for this characterisation. The `RandomForestClassifier` class in `sklearn` is especially suited to this task as it has a built-in method to retrieve feature importance. This is computed as the normalised total reduction of the Gini impurity criterion brought by that feature.

The `sklearn` implementation of the random forest classifier has several hyperparameters, of which those in Table 3 were optimised through a grid search algorithm.

| Parameter | `sklearn` Default | Description |
|---|---|---|
| `n_estimators` | 100 | The number of trees in the forest |
| `max_depth` | None | The maximum depth of each tree |
| `min_samples_split` | 2 | The minimum number of samples required to split an internal node |
| `min_samples_leaf` | 1 | The minimum number of samples required to be at a leaf node |

Table 3: These hyperparameters were varied over sensible ranges. A grid search algorithm tested the performance of a model that used every combination of these parameters, evaluated using the same grouped cross validation scheme.

### 2.5.2 Cross Validation

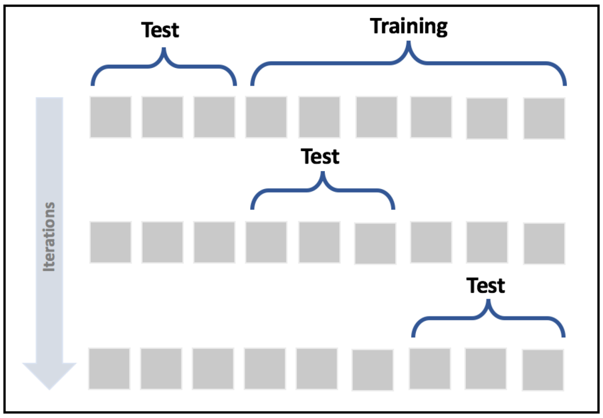


Figure 9: Illustration of cross validation, retrieved from [38]. In this investigation, each test set (fold) was data from a different subject.

To evaluate the performance of models during hyperparameter tuning, observations were grouped on the basis of which subject they originated from. This formed a K-fold cross validation scheme, where a model is trained on all but one subject and tested on the subject that was left out of training. After iterating through each subject, the evaluation metric from each 'fold' is averaged to obtain an unbiased estimate of the model's overall performance.

Individual subjects were used as the folds to simulate the models potential use in a clinical setting where there is no existing data for a patient.

### 2.5.3 Performance Metrics

Since several problems considered had imbalanced distributions of classes, a simple accuracy measure could be misleading. The F1 score, also known as balanced F-score or

F-measure, provides a more robust measure that is appropriate for comparison across problems with different class distributions [39]. The F1 score is defined as the weighted mean of precision (Pr) and recall (Re). Each of these metrics are defined as follows, with respect to true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad \text{and} \quad \mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

$$\mathrm{F} = 2 \cdot \frac{\mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}}.$$

When aggregating F1 scores across the $k$ folds in a cross validation scheme, the simple average was computed. $F^{(i)}$ refers to the test set performance on fold $i$, where $i \in 1, 2, ..., k$. The quoted F1 score for each model is $F_{avg}$, which is defined as follows

$$F_{\text{avg}} = \frac{1}{k} \cdot \sum_{i=1}^{k} F^{(i)}.$$

This scoring metric was used for all of the classification problems which generalised across subjects.

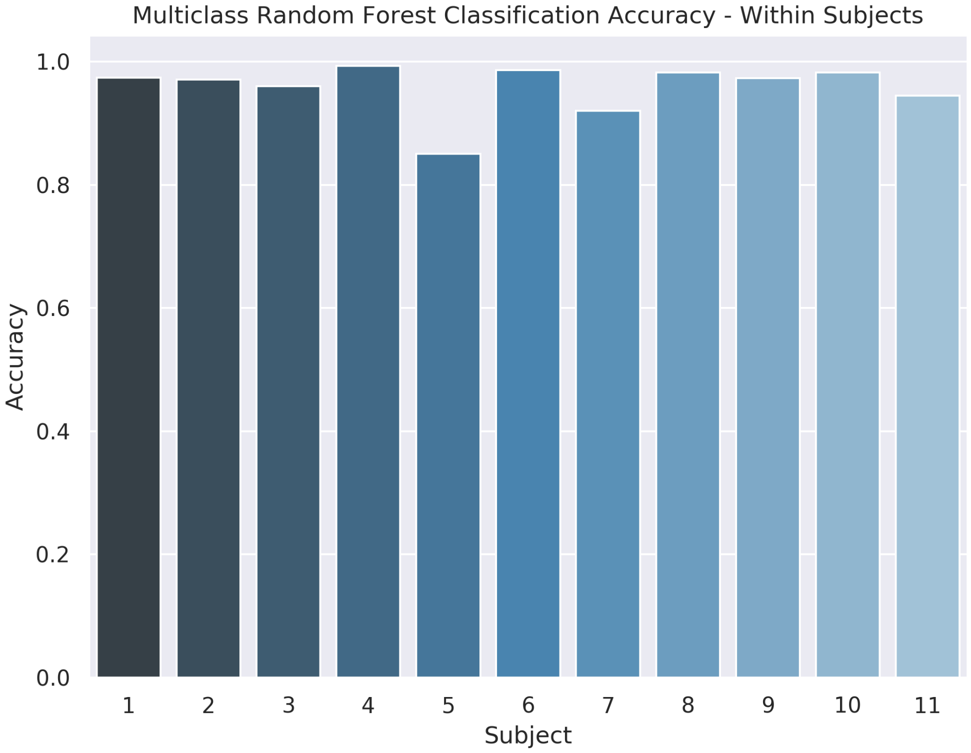


Figure 10: Classification accuracy when models were trained and tested on the same subject. Repeated, stratified cross validation was used to obtain unbiased estimates of model performance for each subject.

# 3 Results

## 3.1 Separate Models for Each Subject

The first problem explored was distinguishing between baseline, ACh and AT recordings. This problem was chosen to validate an ML approach based upon the findings from the drug effect study by Liu et. al. [16]. It was concluded in this study that ACh inhibits pacemaker activity and AT serves to reverse some of these inhibitory effects, suggesting that there should be a clear distinction between the signals when analysing each subject individually [16].

Features from each subject were filtered by the feature selection process outlined in 2.4.2. A separate random forest classifier was fit on the filtered feature matrices from each subject. Simple accuracy was used to measure classification performance for each model as each of the subjects have balanced data sets. For subjects that were not administered AT, the model would distinguish between only baseline and ACh. An unbiased estimate for accuracy was obtained through repeated stratified cross validation. The mean accuracy across all of the models was 98.2%, with a standard deviation of 4.36%. The high accuracy and relatively consistent performance for each model relies on ACh and AT having distinct effects on GI slow waves. This result provides additional validation to the conclusions reached by Liu et. al [16].

The next stage of exploring the applications of ML on this data was to generalise a model across subjects.

## 3.2 Overall Classification Results

| Classes | N | Observations | Features Selected | F1-score | $\sigma_{CV}$ |
|---|---|---|---|---|---|
| Baseline, ACh, AT, Hex | 4 | 990 | 7792 | 0.431 | 0.189 |
| Baseline, ACh, AT | 3 | 810 | 2250 | 0.616 | 0.231 |
| Baseline, ACh, Hex | 3 | 840 | 4064 | 0.638 | 0.244 |
| Baseline, ACh, (AT or Hex) | 3 | 990 | 3907 | 0.480 | 0.143 |
| Baseline, Any Drug | 2 | 990 | 16320 | 0.703 | 0.196 |
| Baseline, ACh | 2 | 660 | 14362 | 0.700 | 0.240 |
| Baseline, AT | 2 | 480 | 18485 | 0.755 | 0.290 |
| Baseline, Hex | 2 | 510 | 17134 | 0.807 | 0.260 |
| ACh, (AT or Hex) | 2 | 660 | 2844 | 0.569 | 0.216 |
| ACh, AT | 2 | 480 | 8751 | 0.744 | 0.316 |
| ACh, Hex | 2 | 510 | 11915 | 0.691 | 0.288 |
| AT, Hex | 2 | 330 | 16238 | 0.950 | 0.154 |

Table 4: Random Forest Classification Results. The stated F1-score is the mean score for each model in the group cross validation scheme. Metrics are rounded to three decimal places.

Several classification problems were considered. For each problem, feature selection was conducted using the modified `FeatureSelector` from `tsfresh`. Next, the filtered feature matrix was fed into a grid search that fit several hundered models, exploring a range of random forest hyperparameters. Each model was evaluated using grouped cross validation scheme where a different subject was used as the test set on each iteration. The models with the highest mean F1 scores for each classification task were saved, the results are summarised in Table 4.

For most classification tasks, performance varied dramatically when testing across different subjects, as shown by the relatively high F1 score standard deviations $\sigma_{CV}$. The 'Baseline, ACh, AT' classifier performed considerably worse when generalised across subjects than the models that were fit on individual subjects, with a mean F1 score of 0.616. Of all the classification tasks considered, distinguishing between AT and Hex recordings was the only one to have consistently strong performance, with a mean F1 score of 0.950. It is interesting to note that some of the worst performing classifiers arose from combining GI slow waves from subjects that had been administered AT or Hex into one class.

The 'Baseline, ACh, AT' and 'AT, Hex' classifiers will be examined more closely in the following sections.

### 3.2.1 Baseline vs. Acetylcholine vs. Atropine

After multiclass feature selection with `n_significant` set to 3, 2,250 features were selected out of 96,750. The best random forest contained 1,000 trees, with no maximum depth and default settings for all other hyperparameters. The inconsistent performance across subjects is visible in Figure 11. Additionally, after computing the top three principal components of the selected features in this problem, there is poor separation of the target classes.

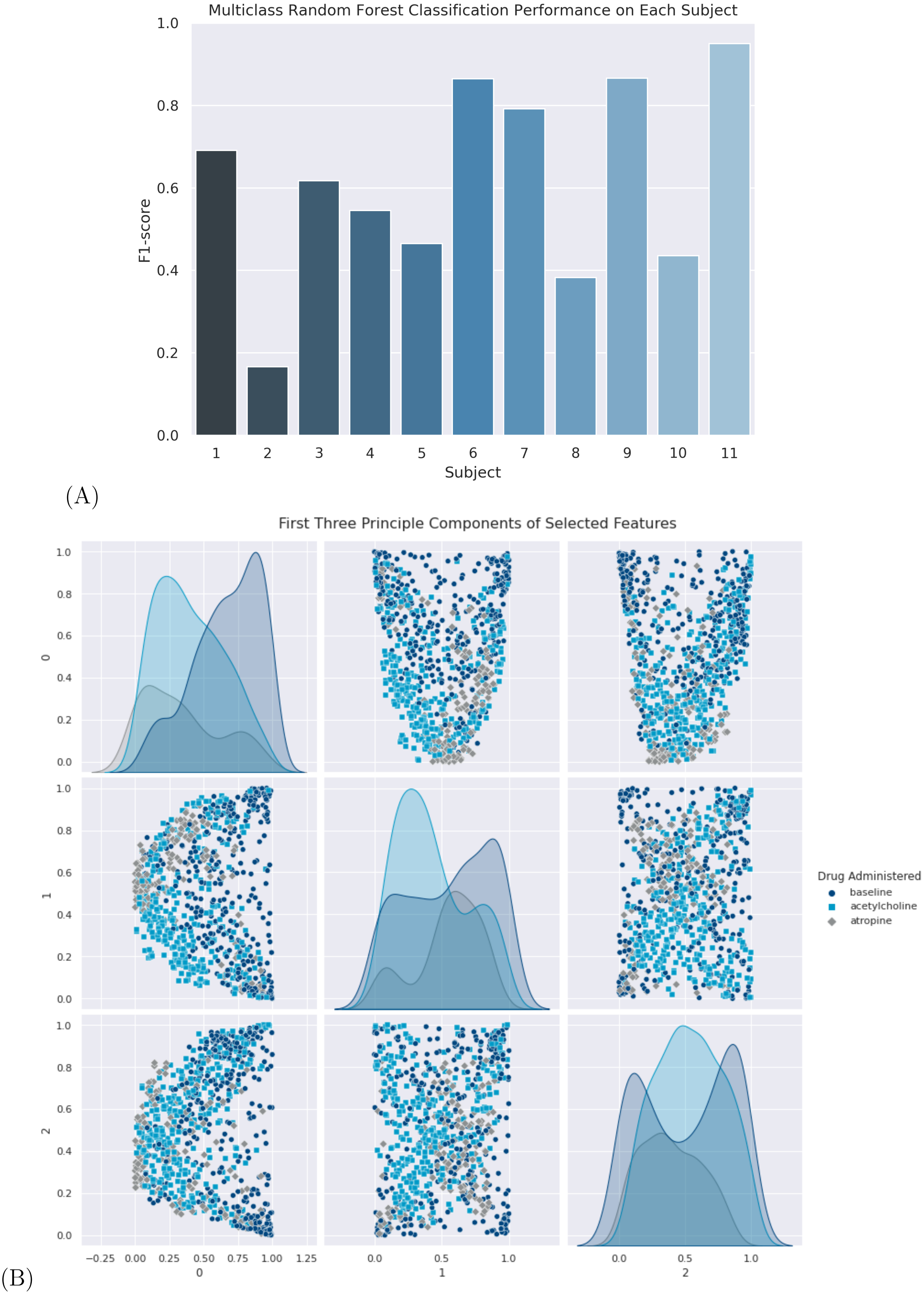


Figure 11: (A) - F1 scores of random forest, tested on each subject under grouped cross validation. (B) - Pairwise distributions of top three principal components of the filtered feature matrix for 'Baseline, ACh, AT'.

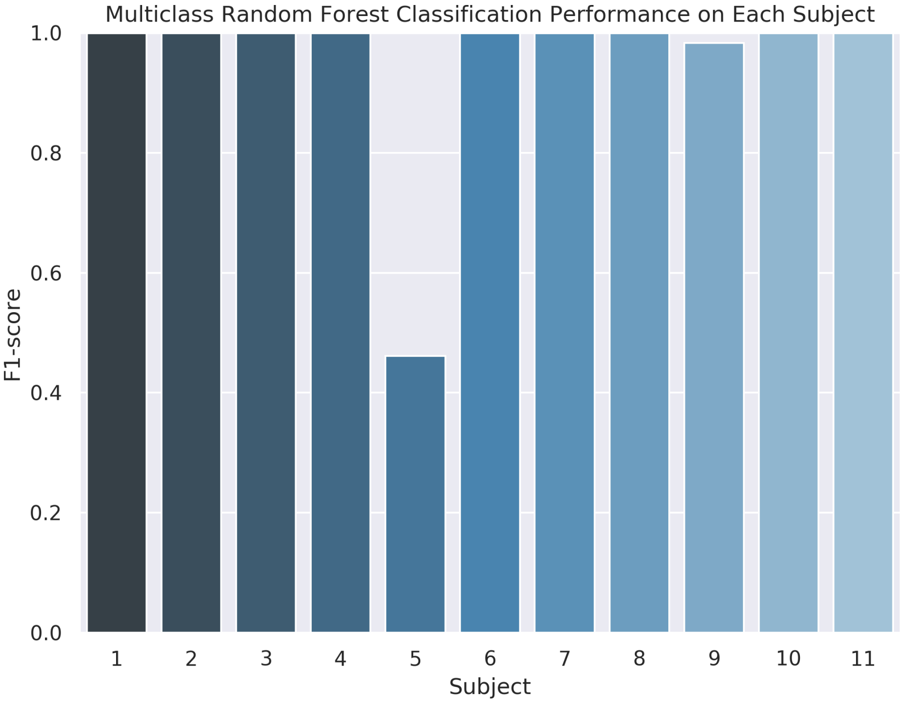


Figure 12: F1 scores of ‘AT vs. Hex’ random forest classifier when each subject was used as a test set.

### 3.2.2 Atropine vs. Hexamethonium

After binary feature selection, 16,238 features were identified as relevant out of 96,750. The best random forest contained 500 trees, with a maximum depth of 2 and default settings for all other hyperparameters. This classifier worked very well on all except one subject, which is responsible for the high standard deviation in the F1 scores as shown in Figure 12. The top five features as ranked by reduction in Gini impurity are as follows:

1. `51__number_cwt_peaks__n_5`
2. `35_30_diff__change_quantiles__f_agg_"var"__isabs_True__qh_0.6__ql_0.4`
3. `54_51_diff__number_cwt_peaks__n_5`
4. `6_8_diff__number_cwt_peaks__n_5`
5. `35_30_diff__spkt_welch_density__coeff_8`

Each of these features are attributes of the signal from a single time series. Plotting pairwise distributions of these features illustrates their effectiveness in separating the target classes. Using only feature 1 and 3 in Figure 13, near perfect separation of the data set can be achieved.

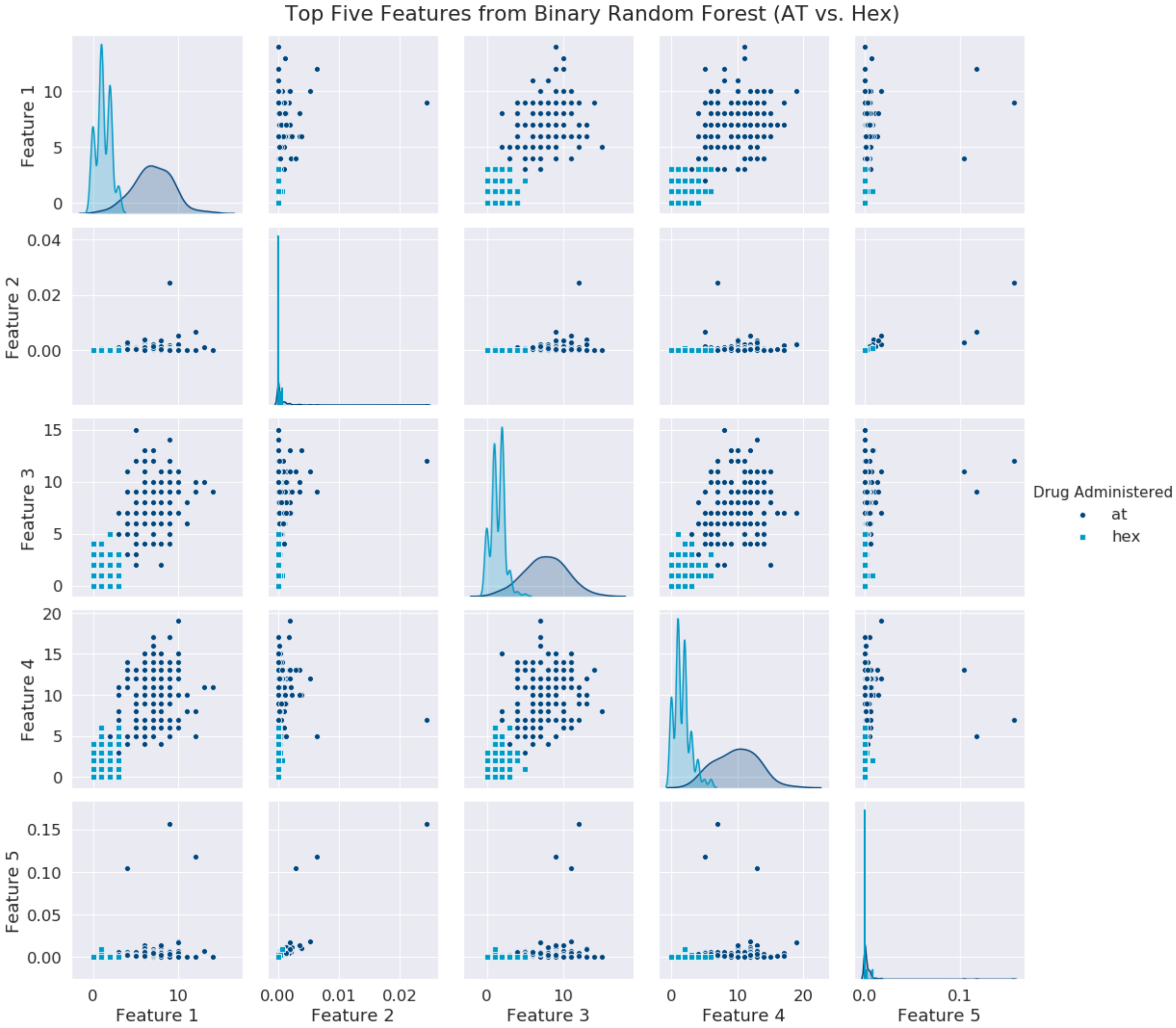


Figure 13: Pairwise distributions of the top five features of the random forest for 'AT, Hex'.

## 3.3 Analysis of Key Features

Three of the top five features in 'AT vs. Hex' are of the same type and share very similar distributions, indicating that the random forest classifier used many different collinear features during training. After aggregating the Gini importance of each major feature group over the top 200 features in the random forest, the three most useful types of features are `number_cwt_peaks`, `change_quantiles` and `cid_ce`. Although a `spkt_welch_density` feature appears in the top five, when the Gini importance is aggregated over the top 200 features, the relative importance of this feature type falls.

A high proportion of features of type `number_cwt_peaks` are in the top 200, and all features of this type were identified as relevant during feature selection. The feature type `change_quantiles` has a similar number of occurrences in the top 200 features, but there are 30 times as many of these features available. `cid_ce` is the next most important feature group, but the impact of this feature type is much smaller relative to the top two. The most important feature type, `number_cwt_peaks` will be explored further.

|  |  |  |  | Occurrences |  |  |
|---|---|---|---|---|---|---|
| **Rank** | **Symbol** | **Feature Type** | **Importance** | **Top 200** | **Filtered** | **Extracted** |
| 1 | $X_{np}$ | number_cwt_peaks | 0.338 | 69 | 258 | 258 |
| 2 | $X_{cq}$ | change_quantiles | 0.313 | 68 | 3514 | 7740 |
| 3 | $X_{cid}$ | cid_ce | 0.093 | 22 | 204 | 258 |

Table 5: Aggregated Gini importance and number of occurrences for the top three feature types. Occurrences are counts of the number of features of each type in the top 200 features in the random forest, the set of filtered features for 'AT vs. Hex' and the overall set of extracted features.

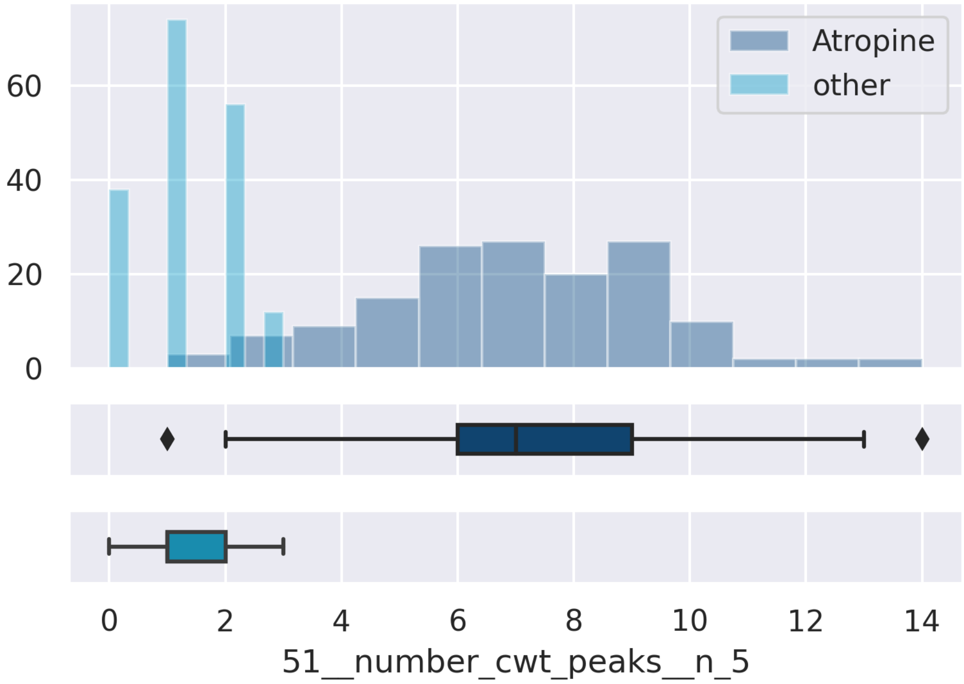


Figure 14: Conditional distribution of the number of peaks in a CWT of the signal from electrode 51.

### 3.3.1 Continuous Wavelet Transform

The most important feature of this type is `51__number_cwt_peaks__n_5`, which refers to the number of peaks in a continuous wavelet transform (CWT) of the signal from electrode 51, while considering a width of up to 5. The continuous wavelet transform computed in `tsfresh` uses a Ricker wavelet to smooth the time series and returns the number of maxima that occur at enough width scales, with sufficiently high Signal-to-Noise-Ratio. Figure 14 shows the distribution of the most important feature of this type. There is strong separation between the two classes, with GI slow waves under the effect of AT generally having between 4 and 10 peaks and Hex influenced slow waves only having up to 3.

The mechanism by which CWT peaks are found is through a convolution of the time series $S$ with the Ricker wavelet [40]. The Ricker wavelet models the function $f(x)$ shown

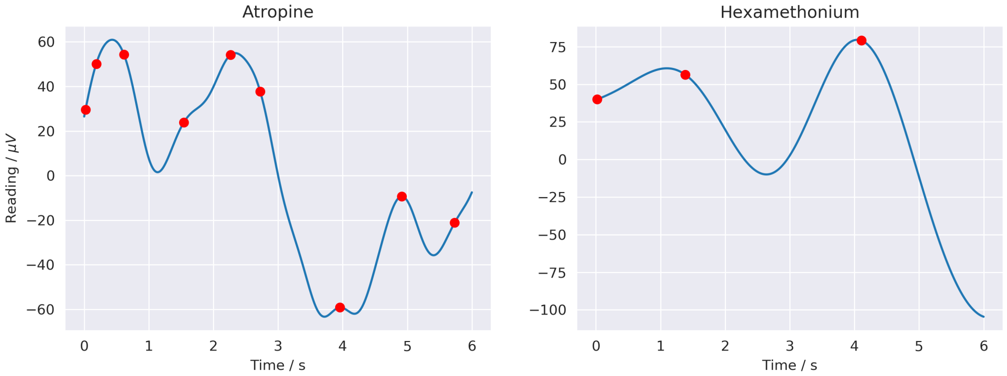


Figure 15: GI slow waves recorded under the effects of AT and Hex, taken from subject 00_0315 and subject 01_0126 respectively. CWT peaks are denoted by red dots. Widths up to $a = 5$ were considered.

below, where $x$ is the raw signal and $a$ is the width parameter:

$$f(x) = A \cdot \left(1 - \left(\frac{x}{a}\right)^2\right) \cdot e^{-\frac{x^2}{2a^2}},$$

$$\text{where} \quad A = \frac{2}{\sqrt{3a}\pi^{1/4}}.$$

The resulting feature $X_{np}$ is a count of local maxima in the CWT of time series $S$. Applying this transformation to AT and Hex samples through the `scipy.signal` submodule and overlaying the identified peaks on the raw time series illustrates the drivers of the separation that is seen in this feature. This feature appears to correlate strongly with the frequency of the time series, but also with noise and variability. Hexamethonium influenced GI slow waves appear to be much smoother and of lower frequency than those under the effect of Atropine, visible in Figure 15.

### 3.3.2 Consecutive Changes

The second most important feature type is calculated by aggregating the absolute values of consecutive changes within a corridor, fixed between two *quantiles* of the distribution of the time series. The feature `35_30_diff__change_quantiles__f_agg_"var"__isabs_True__qh_0.6__ql_0.4` considers the virtual time series that represents the difference between the readings from electrodes 35 and 30, $S$. A corridor is fixed between the 2nd (0.4) and 3rd (0.6) quintiles in the distribution of the time series. The consecutive values considered, $s_i$, $s_{i+1}$ are defined as being larger than the $40^{th}$ percentile $P_{40}(S)$ and smaller than the $60^{th}$ percentile $P_{60}(S)$. The percentiles are computed such that, for a randomly chosen value $s \in \{s_1, ..., s_j, ..., s_k, ..., s_T\}$:

$$\Pr[s \leq P_{40}(S)] = 0.4,$$
$$\Pr[s \leq P_{60}(S)] = 0.6.$$

From this, $i$ is defined as $i \in \{j, ..., k-1\}$, where

$$P_{40}(S) \leq s_i \leq P_{60}(S).$$

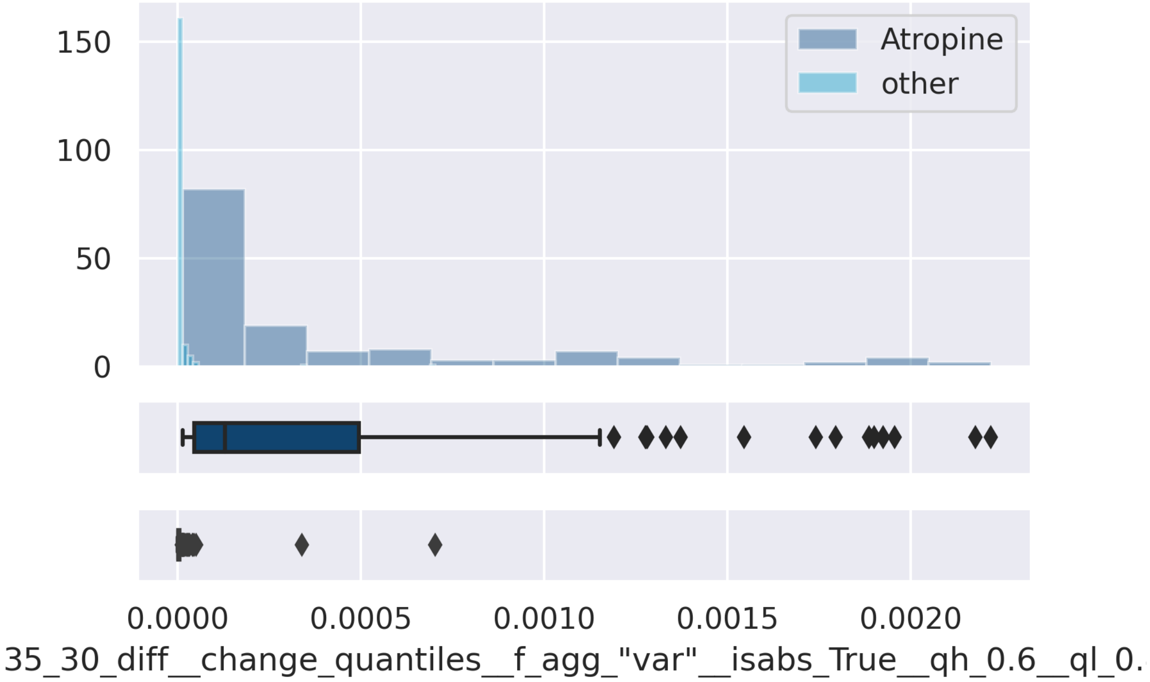


Figure 16: Conditional distribution of the variance of consecutive absolute changes in the 3rd quintile for the difference between the readings from electrodes 35 and 30. The maximum value on the $x$-axis has been limited to 0.0025 to improve readability.

Next, the variance of the absolute values of consecutive changes within this corridor are computed. The feature $X_{cq}$ is thus computed from time series $S$ as

$$X_{cq} = \mathrm{Var}(\{|s_{i+1} - s_i|;\ s_i \in \{j, ..., k-1\}).$$

The conditional distribution in Figure 16 shows that GI slow waves under the effect of AT exhibit far more variance from reading to reading than GI slow waves from subjects that have been administered Hex. Furthermore, AT recordings have a much wider range of variance values in this quantile. GI slow waves under the effect of Hex mostly show near-zero variance for the quantile considered.

### 3.3.3 Time Series Complexity

The third most important feature type is an estimate of time series complexity, using complexity-invariant distance (CID) [41]. In `tsfresh`, this is defined by

$$\mathrm{CID} = \sqrt{\sum_{i=1}^{n-1} (s_i - s_{i-1})^2},$$

where $s_i$ is the value of the time series $S$ at time step $i$ and $n$ is the number of time steps in the window considered. The most important feature of this type in the 'AT vs. Hex' random forest classifier is `1__cid_ce__normalize_True`, which considers signals recorded by electrode 1 and *normalises* them. The normalisation procedure utilises the mean $\mu$ and standard deviation $\sigma$ of time series $S$ and is calculated by

$$s_{i,\mathrm{norm}} = \frac{s_i - \mu}{\sigma},$$

Where $s_{i,\text{norm}}$ is the value of the normalised time series $S_{\text{norm}}$ at time step $i$. The resulting feature $X_{cid}$ is the CID computed from $S_{\text{norm}}$. The conditional distribution of this feature in Figure 17 shows strong separation between AT and Hex recordings. Generally, normalised GI slow waves from subjects under the effects of AT have a higher CID than those from subjects under the effects of Hex.

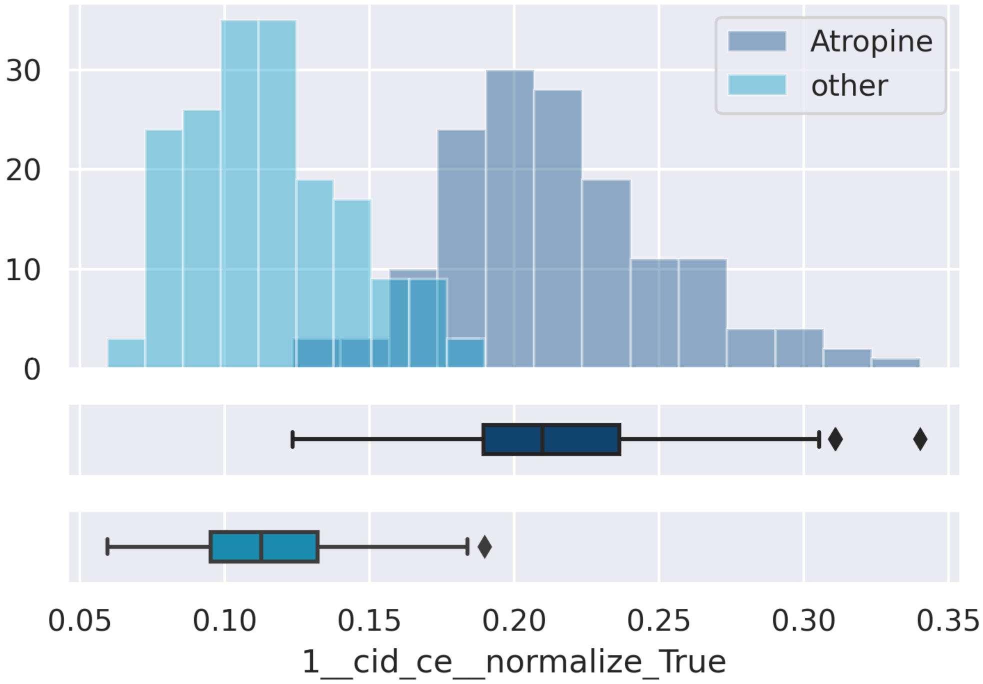


Figure 17: Conditional distribution of the CID of the normalised signals from electrode 1.

Each of the three most important feature type indicate that AT causes an excitatory action on GI slow waves when administered after ACh, compared to the effects of Hex. AT influenced recordings have a higher number of peaks, more variance in consecutive readings and higher complexity than those recorded under the effect of Hex.

# 4 Discussion

The characterisation of the effects of drugs on GI slow wave activity using ML has not been conducted prior to this investigation. This has been due to a general lack of high resolution GI slow wave recordings, but with the MEA drug study conducted by Liu et. al. [16], an opportunity to explore ML techniques was presented. This investigation found that it could be predicted with high accuracy whether a subject had been administered Atropine or Hexamethonium after the administering of Acetylcholine. The specific time series features that characterise the differences in GI slow waves in response to each drug were described. Results from this investigation concur with the conclusions from Liu et. al. that AT has an excitatory effect on GI slow waves, reversing the inhibitory effects of ACh, while Hex does not [16]. The predictive model devised serves as a foundation for future work and potential clinical applications of ML in GI-related drug discovery and the diagnosis of GI disorders.

## 4.1 Automated Time Series Feature Extraction

The use of automated time series feature extraction subverts the need for the laborious process of manual time series feature engineering. The highly parallel implementation of the FRESH algorithm in `tsfresh` allows comprehensive feature extraction to be conducted in feasible run times, with 96,750 features extracted in total. A key contribution from this project was the addition of multiclass feature selection using an OvR scheme to the `tsfresh` library, further streamlining the process of feature selection for multiclass problems [33, 34]. This approach provides a more strict filter that can be tuned with the addition of the parameter `n_significant` which controls the number of classes for which a feature must be a relevant predictor.

## 4.2 Spatiotemporal Characteristics of GI Slow Waves

During the time series engineering stage of this investigation, 68 *virtual* sensors which represented the differences in recordings of neighbouring electrodes were created. Of the 16,238 features determined to be relevant for predicting whether a subject has been administered AT or Hex after already having received ACh, 10,353 (63.8%) were extracted from these difference signals. Out of the top 200 features used in the random forest classifier as determined by relative Gini importance, 147 (73.5%) were extracted from difference signals. These augmented signals were created in order to capture some of the spatial characteristics of GI slow wave propagation and proved to be very useful in classifying drug effects.

The most important features in distinguishing AT from Hex through GI slow wave recordings were the number of peaks in a continuous wavelet transform, the variance of consecutive changes and time series complexity. Each of these features showed clear separation between the two classes upon examining their conditional distributions. Recordings from subjects under the effect of AT had consistently higher values for each of these features, clearly portraying the excitatory effects on GI slow waves that AT has [16].

## 4.3 MEA Data Limitations

Applying the systematic feature engineering process to the MEA data in this investigation yielded varying results, with all of the multiclass classifiers developed having inconsistent performance on different test subjects. Comparing the classification performance on the 'Baseline, ACh, AT' problem when testing within subjects versus testing on new subjects reveals the limitations of the MEA data. When testing on unseen data from the same subject that the training data originated from, an average classification accuracy of 98.2% was achieved. However, when data from multiple subjects was used for training and testing was conducted on an unseen subject, the classification F1 score varied from as low as 0.167 to as high as 0.950. This suggests that there is significant variability in GI slow waves across different subjects, and these subjects may have differing responses to the administering of ACh and AT. Each GI tissue subject was retrieved from one of two ICR mice and they were taken from different sections of the ileum, which could have contributed to the inconsistency in the data [16, 7]. Data points that capture underlying attributes of the subjects such as their relative position in the ileum, biological data on the ICR mice from which they were retrieved and conditions under which MEA recording was undertaken could help to remedy this. Such features are easily combined with GI slow wave recordings in a random forest classifier.

There is also the issue of noise and incorrect readings. It was highlighted earlier (3) that some electrodes have anomalous readings, which could be attributed to poor contact with the subject or other environmental factors. Although a random forest is robust to such outliers, this may limit the validity of features that capture spatial characteristics of slow waves.

## 4.4 Future Work

To extend the applicability of ML techniques in characterising the effects of drugs on GI slow waves, more training data from distinct subjects is needed. Furthermore, the additional data points discussed in 4.3 will contribute to more robust predictive models. Additional approaches to signal classification that focus on the spatial propagation of slow waves should be considered, such as the creation of space-delay matrices, as seen in EEG signal classification [20]. Moreover, additional ways to quantify the spatial propagation from short MEA recording samples should be explored to facilitate the creation of more potentially informative features. Finally, using images of activation patterns in the MEA could serve as a foundation for exploring image classification with deep learning models

# 5 Conclusions

The effects of different drugs on the characteristics of GI slow waves were explored using ML techniques. The available data consisted of GI slow wave potentials recorded from a 60 electrode MEA. Automated feature extraction was leveraged to conduct systematic feature engineering on this data set. 96,750 features were computed from 129 different time series. The FRESH algorithm was used to select features based on scalable hypothesis tests [17]. A multiclass variant of this feature selection process was implemented and added to the `tsfresh` Python library [34]. Random forest classifiers were fitted on the extracted features, which could reliably predict whether a subject was administered AT or Hex subsequent to the administering of ACh, with 16,238 relevant features and a mean F1 score of 0.950. The most important features included the number of peaks in a CWT of the time series, the variance of consecutive changes in one quintile of the the time series and the CID of the normalised time series. The conditional distributions of these features revealed strong separation between the classes 'AT' and 'Hex'. GI slow waves from subjects under the effect of AT consistently had higher values for each of the most important features, indicating that AT has an excitatory effect on pacemaker potentials when administered after ACh, supporting the conclusions of Liu. et. al. [16]. Overall, a systematic framework has been established for characterising GI slow waves through ML that avoids laborious feature engineering.

# References

[1] B. Greenwood-Van Meerveld, A. C. Johnson, and D. Grundy, "Gastrointestinal Physiology and Function," in *Gastrointestinal Pharmacology*, ser. Handbook of Experimental Pharmacology, B. Greenwood-Van Meerveld, Ed. Cham: Springer International Publishing, 2017, pp. 1–16. [Online]. Available: https://doi.org/10.1007/164_2016_118

[2] K. M. Sanders, S. D. Koh, and S. M. Ward, "Interstitial cells of cajal as pacemakers in the gastrointestinal tract," *Annual Review of Physiology*, vol. 68, no. 1, pp. 307–343, Jan. 2006, publisher: Annual Reviews. [Online]. Available: https://www.annualreviews.org/doi/10.1146/annurev.physiol.68.040504.094718

[3] C. V. Almario, M. L. Ballal, W. D. Chey, C. Nordstrom, D. Khanna, and B. M. R. Spiegel, "Burden of Gastrointestinal Symptoms in the United States: Results of a Nationally Representative Survey of Over 71,000 Americans," *American Journal of Gastroenterology*, vol. 113, no. 11, pp. 1701–1710, Nov. 2018. [Online]. Available: https://journals.lww.com/ajg/Abstract/2018/11000/Burden_of_Gastrointestinal_Symptoms_in_the_United.25.aspx

[4] A. F. Peery, S. D. Crockett, C. C. Murphy, J. L. Lund, E. S. Dellon, J. L. Williams, E. T. Jensen, N. J. Shaheen, A. S. Barritt, S. R. Lieber, B. Kochar, E. L. Barnes, Y. C. Fan, V. Pate, J. Galanko, T. H. Baron, and R. S. Sandler, "Burden and Cost of Gastrointestinal, Liver, and Pancreatic Diseases in the United States: Update 2018," *Gastroenterology*, vol. 156, no. 1, pp. 254–272.e11, Jan. 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6689327/

[5] F. J. Ehlert, K. J. Pak, and M. T. Griffin, "Muscarinic Agonists and Antagonists: Effects on Gastrointestinal Function," in *Muscarinic Receptors*, ser. Handbook of Experimental Pharmacology, A. D. Fryer, A. Christopoulos, and N. M. Nathanson, Eds. Berlin, Heidelberg: Springer, 2012, pp. 343–374. [Online]. Available: https://doi.org/10.1007/978-3-642-23274-9_15

[6] L. A. Bradshaw, W. O. Richards, and J. P. Wikswo, "Volume conductor effects on the spatial resolution of magnetic fields and electric potentials from gastrointestinal electrical activity," *Medical and Biological Engineering and Computing*, vol. 39, no. 1, pp. 35–43, Jan. 2001. [Online]. Available: https://doi.org/10.1007/BF02345264

[7] J. Y. H. Liu, P. Du, W. Y. Chan, and J. A. Rudd, "Use of a microelectrode array to record extracellular pacemaker potentials from the gastrointestinal tracts of the ICR mouse and house musk shrew (Suncus murinus)," *Cell Calcium*, vol. 80, pp. 175–188, Jun. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0143416019300442

[8] A. C. Nanivadekar, D. M. Miller, S. Fulton, L. Wong, J. Ogren, G. Chitnis, B. McLaughlin, S. Zhai, L. E. Fisher, B. J. Yates, and C. C. Horn, "Machine learning prediction of emesis and gastrointestinal state in ferrets," *PLoS ONE*, vol. 14, no. 10, Oct. 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6799899/

[9] A. S. Agrusa, A. A. Gharibans, A. A. Allegra, D. C. Kunkel, and T. P. Coleman, “A Deep Convolutional Neural Network Approach to Classify Normal and Abnormal Gastric Slow Wave Initiation From the High Resolution Electrogastrogram,” *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 3, pp. 854–867, Mar. 2020, conference Name: IEEE Transactions on Biomedical Engineering.

[10] G. O’Grady, T. H.-H. Wang, P. Du, T. Angeli, W. J. Lammers, and L. K. Cheng, “Recent progress in gastric arrhythmia: Pathophysiology, clinical significance and future horizons,” *Clinical and experimental pharmacology & physiology*, vol. 41, no. 10, pp. 854–862, Oct. 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4359928/

[11] J. Yin and J. D. Z. Chen, “Electrogastrography: Methodology, Validation and Applications,” *Journal of Neurogastroenterology and Motility*, vol. 19, no. 1, pp. 5–17, Jan. 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3548127/

[12] P. Du, S. Calder, T. R. Angeli, S. Sathar, N. Paskaranandavadivel, G. O’Grady, and L. K. Cheng, “Progress in Mathematical Modeling of Gastrointestinal Slow Wave Abnormalities,” *Frontiers in Physiology*, vol. 8, Jan. 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5775268/

[13] T. W. Kim, S. D. Koh, T. Ördög, S. M. Ward, and K. M. Sanders, “Muscarinic regulation of pacemaker frequency in murine gastric interstitial cells of Cajal,” *The Journal of Physiology*, vol. 546, no. Pt 2, pp. 415–425, Jan. 2003. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2342515/

[14] L. E. A. Montgomery, E. A. Tansey, C. D. Johnson, S. M. Roe, and J. G. Quinn, “Autonomic modification of intestinal smooth muscle contractility,” *Advances in Physiology Education*, vol. 40, no. 1, pp. 104–109, Feb. 2016, publisher: American Physiological Society. [Online]. Available: https://journals.physiology.org/doi/full/10.1152/advan.00038.2015

[15] S. K. Sarna, *Regulatory Mechanisms.* Morgan & Claypool Life Sciences, 2010, publication Title: Colonic Motility: From Bench Side to Bedside. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK53475/

[16] J. Y. H. Liu, P. Du, and J. A. Rudd, “Acetylcholine exerts inhibitory and excitatory actions on mouse ileal pacemaker activity: role of muscarinic versus nicotinic receptors,” *American Journal of Physiology-Gastrointestinal and Liver Physiology*, vol. 319, no. 1, pp. G97–G107, Jun. 2020, publisher: American Physiological Society. [Online]. Available: https://journals-physiology-org.ezproxy.auckland.ac.nz/doi/full/10.1152/ajpgi.00003.2020

[17] M. Christ, A. W. Kempa-Liehr, and M. Feindt, “Distributed and parallel time series feature extraction for industrial big data applications,” Oct. 2016. [Online]. Available: https://arxiv.org/abs/1610.07717v3

[18] B. D. Fulcher, “Feature-based time-series analysis,” Sep. 2017. [Online]. Available: https://arxiv.org/abs/1709.08055v2

[19] A. Shoeb and J. Guttag, “Application of Machine Learning To Epileptic Seizure Detection,” p. 8, 2010.

[20] J. R. Williamson, D. W. Bliss, D. W. Browne, and J. T. Narayanan, “Seizure prediction using EEG spatiotemporal correlation structure,” *Epilepsy & Behavior*, vol. 25, no. 2, pp. 230–238, Oct. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1525505012004763

[21] M. Valderrama, C. Alvarado, S. Nikolopoulos, J. Martinerie, C. Adam, V. Navarro, and M. Le Van Quyen, “Identifying an increased risk of epileptic seizures using a multi-feature EEG–ECG classification,” *Biomedical Signal Processing and Control*, vol. 7, no. 3, pp. 237–244, May 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1746809411000450

[22] S. Grewal and J. Gotman, “An automatic warning system for epileptic seizures recorded on intracerebral EEGs,” *Clinical Neurophysiology*, vol. 116, no. 10, pp. 2460–2472, Oct. 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1388245705002671

[23] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[24] B. Boashash and S. Ouelha, “Automatic signal abnormality detection using time-frequency features and machine learning: A newborn EEG seizure case study,” *Knowledge-Based Systems*, vol. 106, pp. 38–50, Aug. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705116301174

[25] W. McKinney, “pandas: a Foundational Python Library for Data Analysis and Statistics,” p. 9.

[26] 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.

[27] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package),” *Neurocomputing*, vol. 307, pp. 72–77, Sep. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231218304843

[28] A. Kempa-Liehr, J. Oram, and T. Besier, “Automating time series feature engineering for activity recognition from synchronized inertial measurement units,” Jul. 2018.

[29] “Platforms.” [Online]. Available: https://www.nesi.org.nz/services/high-performance-computing-and-analytics/platforms

[30] “Slurm Workload Manager - Overview.” [Online]. Available: https://slurm.schedmd.com/overview.html

[31] “tsfresh — tsfresh 0.17.1.dev9+gcd5b209 documentation.” [Online]. Available: https://tsfresh.readthedocs.io/en/latest/index.html

[32] P. E. McKnight and J. Najab, “Mann-Whitney U Test,” in *The Corsini Encyclopedia of Psychology.* American Cancer Society, 2010, pp. 1–1, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470479216.corpsy0524. [Online]. Available: http://onlinelibrary.wiley.com/doi/abs/10.1002/9780470479216.corpsy0524

[33] Y. Tang, K. Blincoe, and A. W. Kempa-Liehr, “Enriching feature engineering for short text samples by language time series analysis,” *EPJ Data Science*, vol. 9, no. 26, pp. 1–59, 2020.

[34] K. Vyas, “Multiclass,” GitHub, github.com, Pull Request 762, 2020. [Online]. Available: https://github.com/blue-yonder/tsfresh/pull/762

[35] M. Pal, “Random forest classifier for remote sensing classification,” *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, Jan. 2005, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01431160412331269698. [Online]. Available: https://doi.org/10.1080/01431160412331269698

[36] C. Nguyen, Y. Wang, and H. N. Nguyen, “Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic,” vol. 2013, May 2013, publisher: Scientific Research Publishing. [Online]. Available: http://www.scirp.org/journal/PaperInformation.aspx?PaperID=31887

[37] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, “An assessment of the effectiveness of a random forest classifier for land-cover classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 93–104, Jan. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271611001304

[38] “Cross-validation - Mastering Machine Learning Algorithms [Book].” [Online]. Available: https://www.oreilly.com/library/view/mastering-machine-learning/9781788621113/02dbd5eb-fa55-4237-ba62-65bbd30441ef.xhtml

[39] G. Forman and M. Scholz, “Apples-to-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement,” p. 10.

[40] P. Du, W. A. Kibbe, and S. M. Lin, “Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching,” *Bioinformatics*, vol. 22, no. 17, pp. 2059–2065, Sep. 2006, publisher: Oxford Academic. [Online]. Available: https://academic.oup.com/bioinformatics/article/22/17/2059/274284

[41] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, and V. M. A. de Souza, “CID: an efficient complexity-invariant distance for time series,” *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 634–669, May 2014. [Online]. Available: https://doi.org/10.1007/s10618-013-0312-3