



PDkit: A Scalable Computational Data Science Toolbox for High-Frequency Assessment of Parkinson's Disease

George Roussos
g.roussos@bbk.ac.uk

LONDON'S EVENING UNIVERSITY

- Motivation
- PDkit data processing pipeline
- PDkit user and contributor support
- Longitudinal processing with dataflow
- CUSSP study
- Future work

1. Google Scholar: 1,000+ papers on Parkinsonian tremor using accelerometers and ML in 2018-19
 - Impossible to replicate and to compare results
 - Differences in data processing and algorithm implementation
 - In most cases, insufficient details provided to replicate algorithm used
2. Common pattern emerging:
 - Machine Learning processing pipeline
 - From raw data to severity assessment (often using MDS-UPDRS scores)

- Specify digital analytics protocol precisely using software templates
- Unified extensible implementation of digital biomarkers
- Designed using a data processing pipeline approach
- API design follows common python machine learning approach scikit-learn

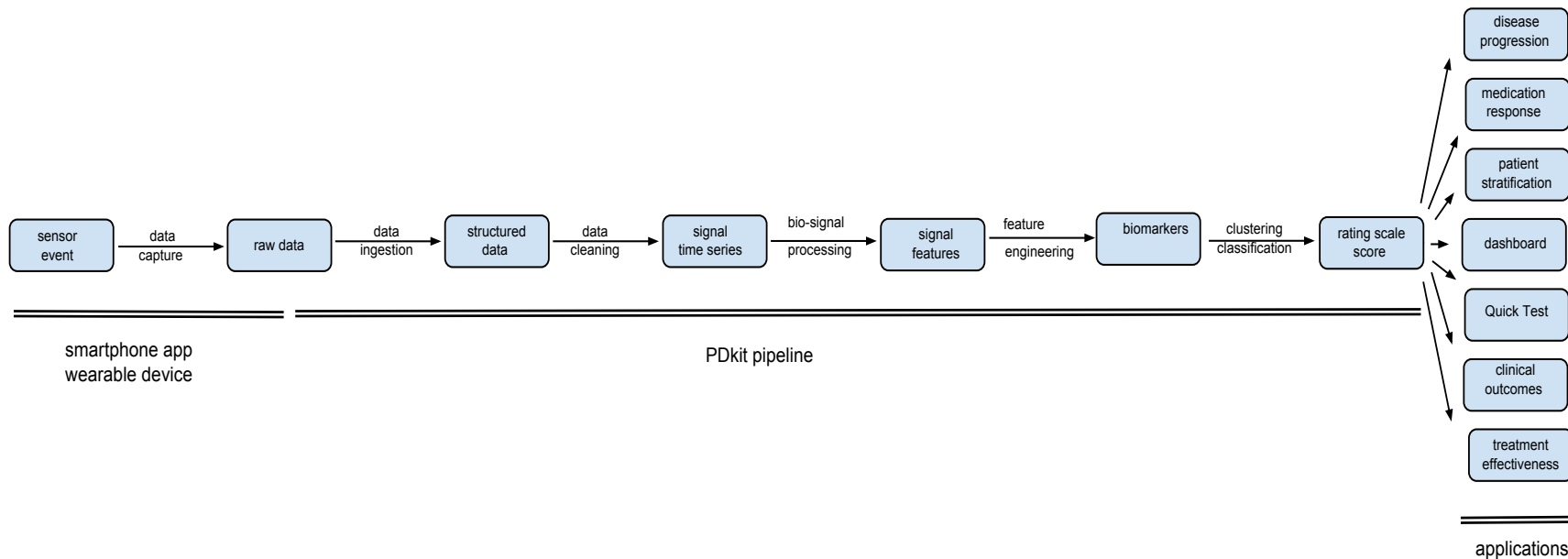
Source code `https://github.com/pdkit/pdkit`

Documentation `https://pdkit.readthedocs.io/`

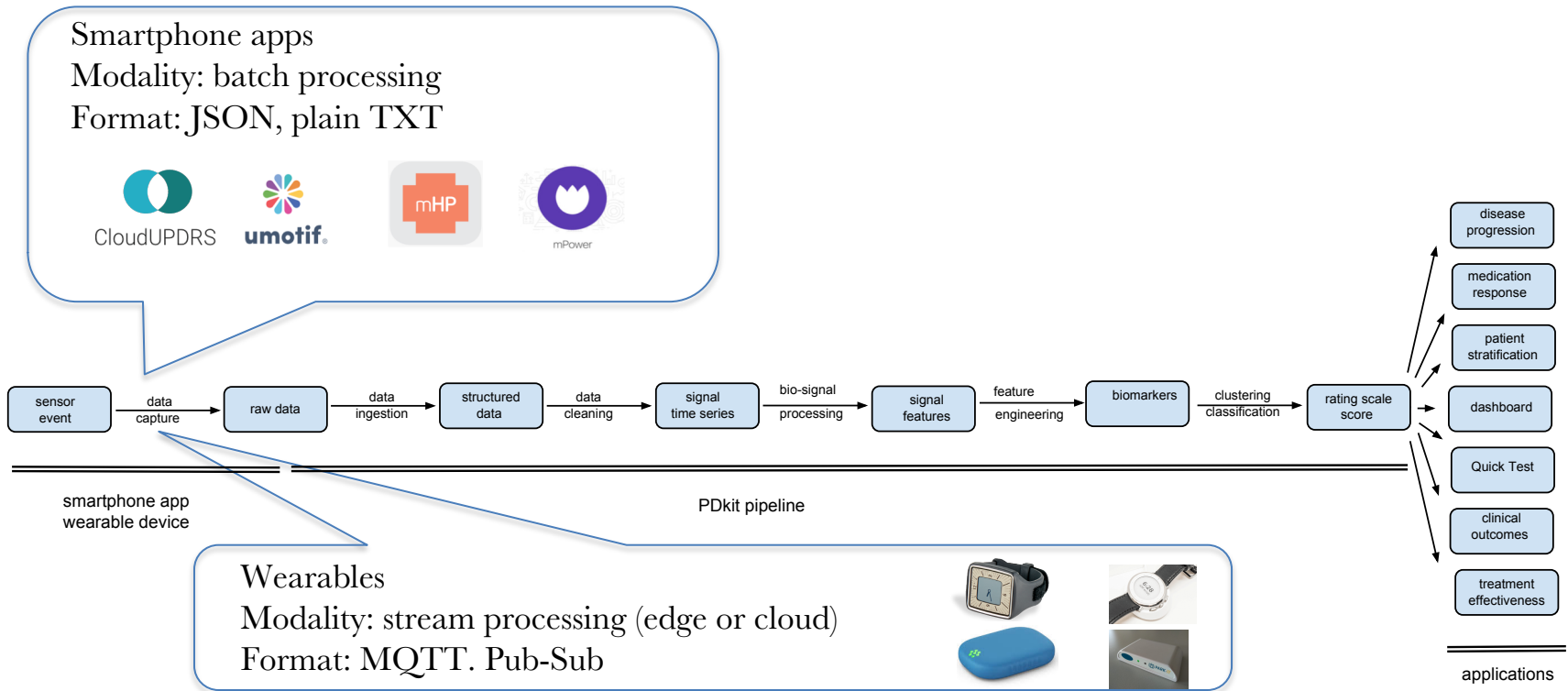
Project info `https://pdkit.github.io/`

Docker images `https://hub.docker.com/r/pdkit/pdkit`

PDkit Processing Pipeline

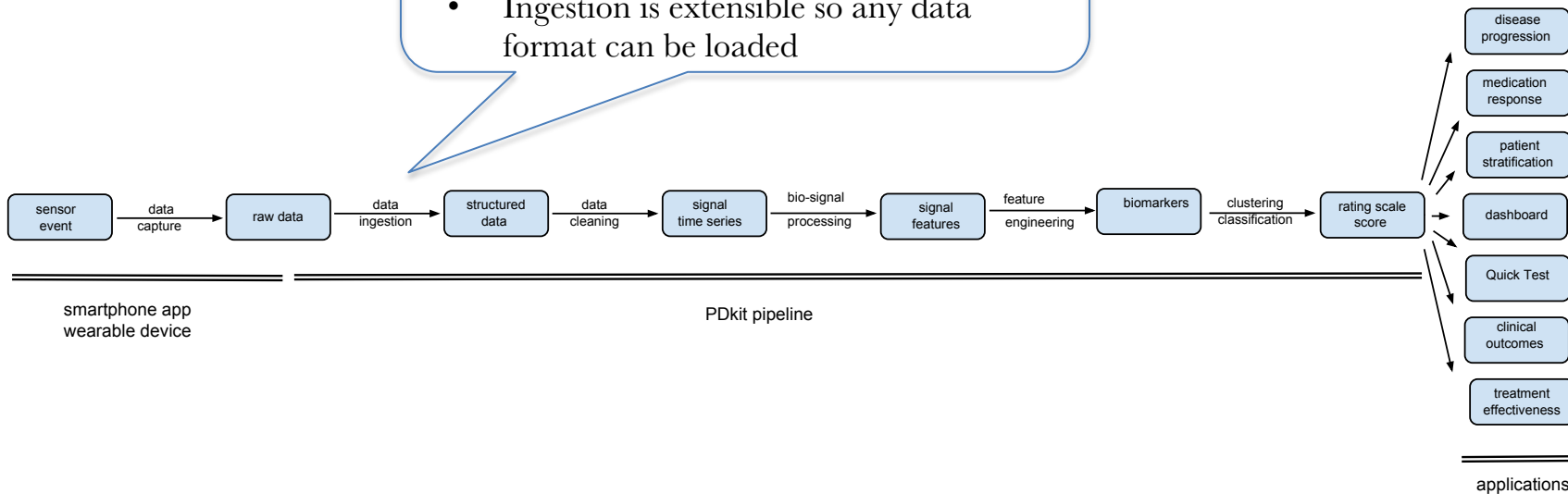


PDkit Processing Pipeline

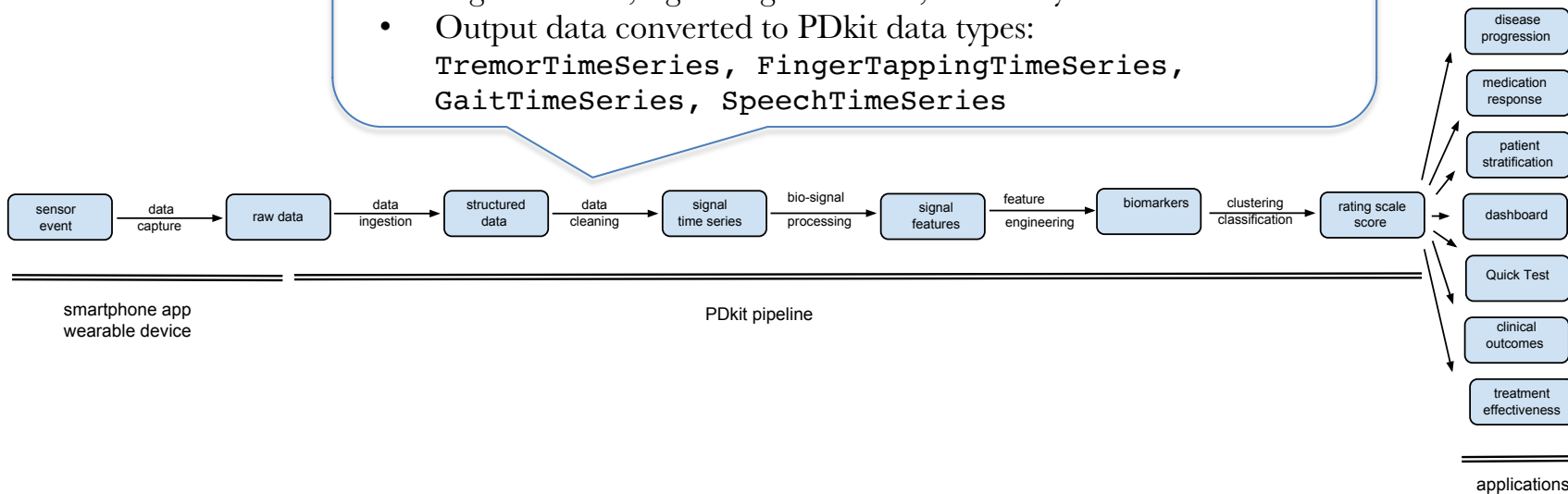


PDkit Processing Pipeline

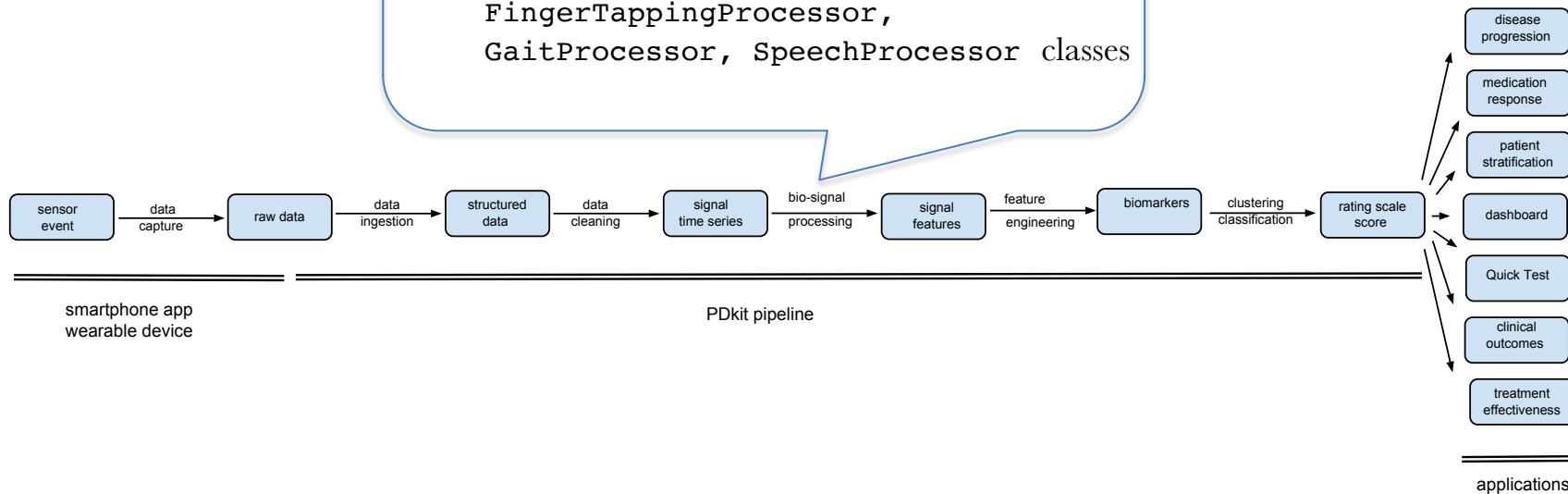
- Data converted to standardized representation based on PANDAS dataframes
- Ingestion is extensible so any data format can be loaded



- Missing and out of range values; data type normalization; indexing; standardised labelling; signal resampling
- Advanced features : gesture verification with DL, data augmentation, signal segmentation, extremity exclusions
- Output data converted to PDkit data types:
`TremorTimeSeries`, `FingerTappingTimeSeries`,
`GaitTimeSeries`, `SpeechTimeSeries`



- Feature extraction
- 500+ tremor, bradikinesia, tapping, gait, turning and speech features
- Implemented in the TremorProcessor, FingerTappingProcessor, GaitProcessor, SpeechProcessor classes



PDkit Processing Pipeline

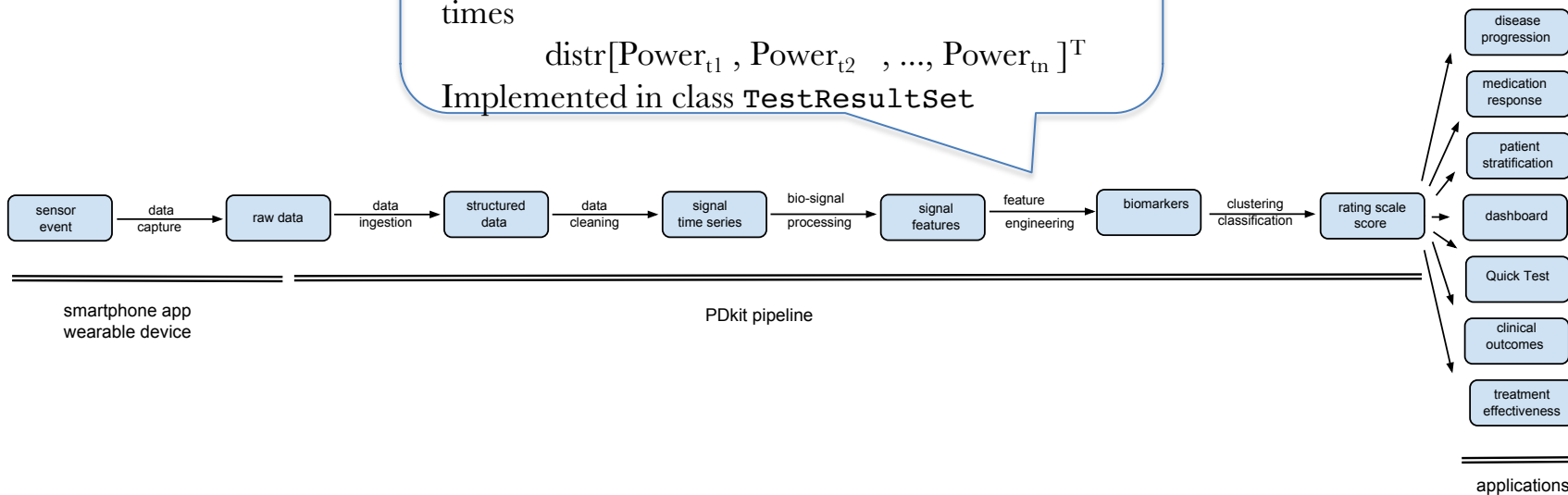
Classic unitary and vector instantaneous biomarkers

$$[\text{Power}, \text{Amplitude}]^T$$

Novel longitudinal biomarkers introducing temporal element e.g. same feature calculated at different times

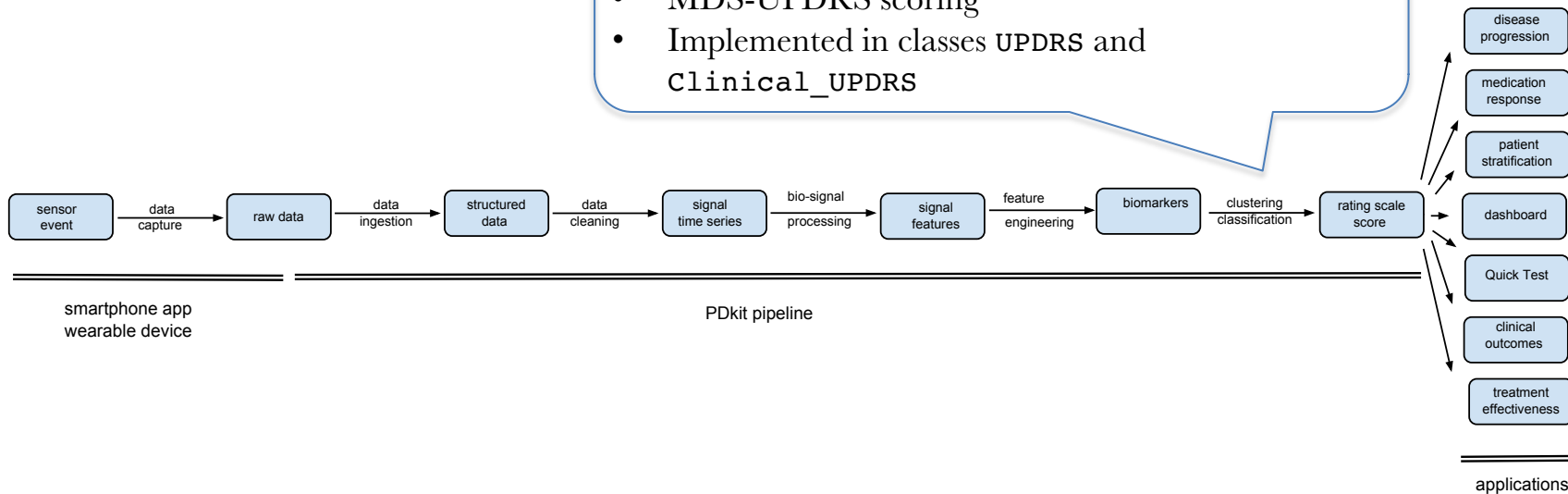
$$\text{distr}[\text{Power}_{t_1}, \text{Power}_{t_2}, \dots, \text{Power}_{t_n}]^T$$

Implemented in class `TestResultSet`



PDkit Processing Pipeline

- Supervised and unsupervised learning for scoring
- Clinical labels or classes created from the data only
- MDS-UPDRS scoring
- Implemented in classes `UPDRS` and `Clinical_UPDRS`



Train a model with clinical labels for MDS-UPDRS scoring

```
>> tp = pdkit.TremorProcessor("raw_data_folder")  
>> ts = pdkit.TremorTimeSeries()
```

initialisation

```
>> amplitude, freq = tp.amplitude(ts, 'welch')  
>> welch_biomarker = pdkit.TestResultSet().process()  
>> clinical_UPDRS = pdkit.Clinical_UPDRS(labels, testResultSet)
```

analytical protocol

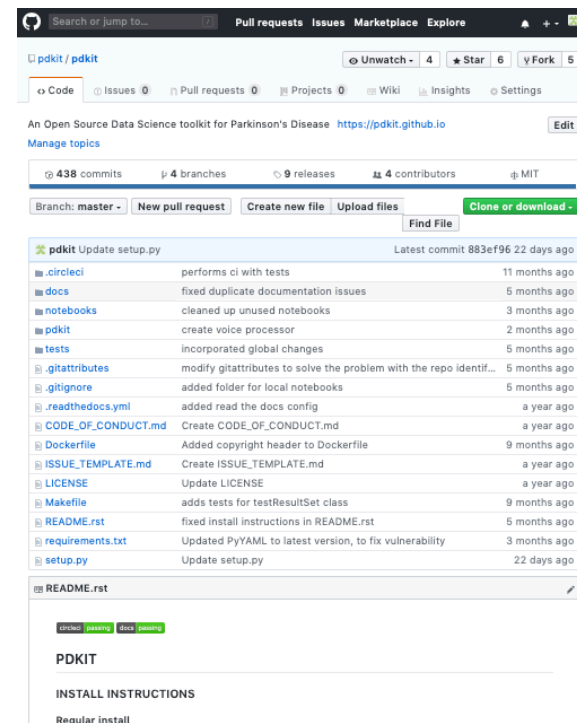
```
>> clinical_UPDRS.predict(new_measurement)
```

score new test

Calculate key gait features on a single data file:

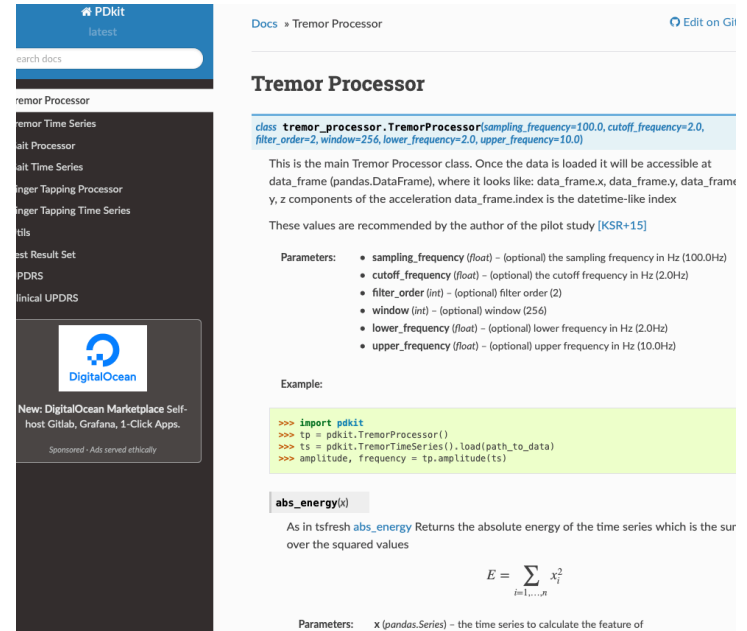
```
>> ts = pdkit.GaitTimeSeries().load(filename)  
>> gp = pdkit.GaitProcessor()  
>> step_regularity, stride_symmetry = gp.walk_regularity_symmetry(ts)
```

- Source code on github
 - Continuous Integration ensures working code
 - MIT License



The screenshot shows the GitHub repository page for `pdkit/pdkit`. The repository is an Open Source Data Science toolkit for Parkinson's Disease, located at <https://pdkit.github.io>. It has 438 commits, 4 branches, 9 releases, and 4 contributors. The repository is licensed under MIT. The file list shows various files and folders, including `.circleci`, `docs`, `notebooks`, `pdkit`, `tests`, `gitattributes`, `gitignore`, `readthedocs.yml`, `CODE_OF_CONDUCT.md`, `Dockerfile`, `ISSUE_TEMPLATE.md`, `LICENSE`, `Makefile`, `README.rst`, `requirements.txt`, and `setup.py`. The `README.rst` file is selected, showing the PDKIT logo and the "INSTALL INSTRUCTIONS" section, which includes a "Regular install" instruction.

- Source code on github
 - Continuous Integration ensures working code
 - MIT License
- Documentation on Read the Docs
 - References to literature
 - Explanation of calculations
 - Auto-generated from the source code



The image shows two side-by-side screenshots. The left screenshot is the GitHub repository page for 'pdkit', showing the 'latest' version and a list of files including 'Tremor Processor', 'Tremor Time Series', 'Tilt Processor', 'Tilt Time Series', 'Finger Tapping Processor', 'Finger Tapping Time Series', 'utils', 'Test Result Set', 'PDRS', and 'Clinical UPDRS'. The right screenshot is the Read the Docs page for 'Tremor Processor', showing the class definition and documentation.

PDKit
latest

Search docs

Tremor Processor

Tremor Time Series
Tilt Processor
Tilt Time Series
Finger Tapping Processor
Finger Tapping Time Series
utils
Test Result Set
PDRS
Clinical UPDRS

DigitalOcean

New: DigitalOcean Marketplace Self-host Gitlab, Grafana, 1-Click Apps.

Sponsored - Ads served ethically

Docs » Tremor Processor [Edit on GitHub](#)

Tremor Processor

```
class TremorProcessor(sampling_frequency=100.0, cutoff_frequency=2.0, filter_order=2, window=256, lower_frequency=2.0, upper_frequency=10.0)
```

This is the main Tremor Processor class. Once the data is loaded it will be accessible at `data_frame` (pandas.DataFrame), where it looks like: `data_frame.x`, `data_frame.y`, `data_frame.z` components of the acceleration data. `data_frame.index` is the datetime-like index

These values are recommended by the author of the pilot study [KSR+15]

Parameters:

- `sampling_frequency` (float) - (optional) the sampling frequency in Hz (100.0Hz)
- `cutoff_frequency` (float) - (optional) the cutoff frequency in Hz (2.0Hz)
- `filter_order` (int) - (optional) filter order (2)
- `window` (int) - (optional) window (256)
- `lower_frequency` (float) - (optional) lower frequency in Hz (2.0Hz)
- `upper_frequency` (float) - (optional) upper frequency in Hz (10.0Hz)

Example:

```
>>> import pdkit
>>> tp = pdkit.TremorProcessor()
>>> ts = pdkit.TremorTimeSeries().load(path_to_data)
>>> amplitude, frequency = tp.amplitude(ts)
```

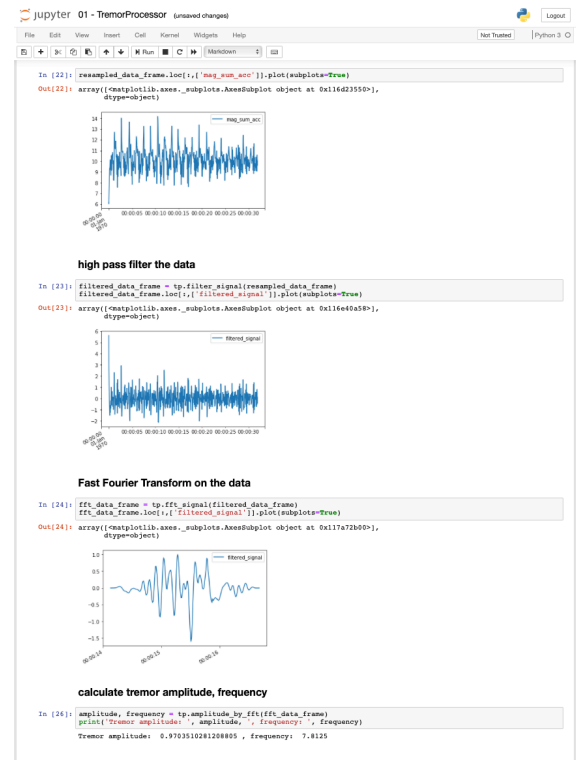
abs_energy(x)

As in `tsfresh.abs_energy` Returns the absolute energy of the time series which is the sum over the squared values

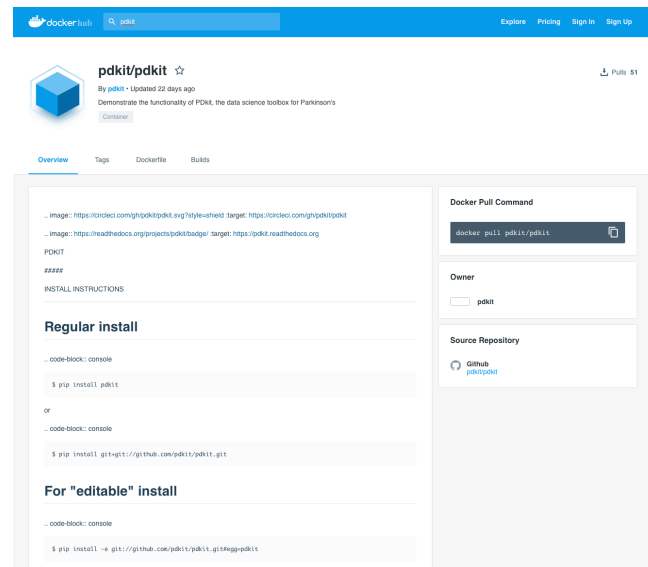
$$E = \sum_{i=1, \dots, n} x_i^2$$

Parameters: `x` (pandas.Series) - the time series to calculate the feature of

- Source code on github
 - Continuous Integration ensures working code
 - MIT License
- Documentation on Read the Docs
 - References to literature
 - Explanation of calculations
 - Auto-generated from the source code
- Jupyter Notebooks demonstrate functionality
 - Notebooks can run on Binder, Deepnote and Colaboratory

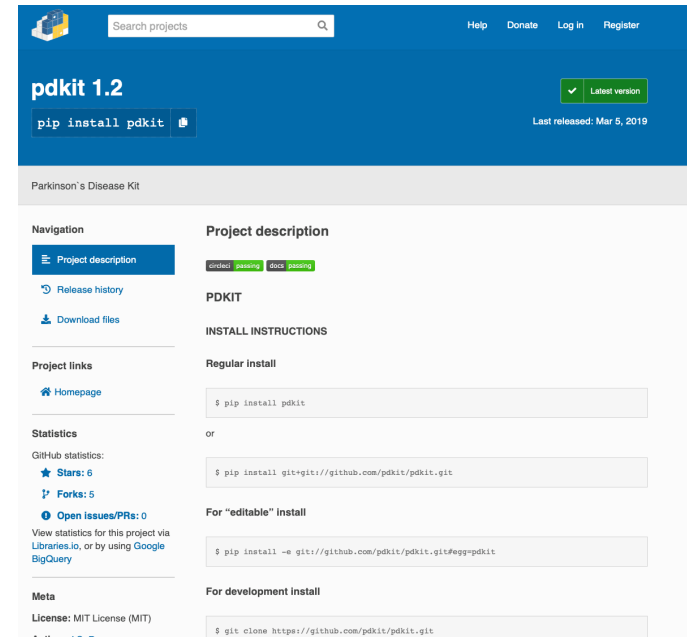


- Source code on github
 - Continuous Integration ensures working code
 - MIT License
- Documentation on Read the Docs
 - References to literature
 - Explanation of calculations
 - Auto-generated from the source code
- Jupyter Notebooks demonstrate functionality
 - Notebooks can run on Binder, Deepnote and Colaboratory
- Docker images for quick deployment
 - Publicly available at Docker Hub



The screenshot shows the Docker Hub interface for the `pdkit/pdkit` image. The page includes a search bar at the top with the text "pdkit", navigation links for "Explore", "Pricing", "Sign In", and "Sign Up", and a "Pull 91" button. The main content area displays the image name "pdkit/pdkit" with a star icon, the author "By pdkit", and the update time "Updated 22 days ago". Below this, there is a description: "Demonstrate the functionality of PDKit, the data science toolbox for Parkinson's". The page is divided into sections: "Overview", "Tags", "Dockerfile", and "Builds". The "Overview" section is active and shows the image's Dockerfile content, including the base image, the PDKIT environment variable, and the installation instructions. The "Regular install" section shows the command `$ pip install pdkit`. The "For 'editable' install" section shows the command `$ pip install -e git://github.com/pdkit/pdkit.git#egg=pdkit`. On the right side, there is a "Docker Pull Command" section with the command `docker pull pdkit/pdkit`, an "Owner" section with the name "pdkit", and a "Source Repository" section with a link to the GitHub repository.

- Source code on github
 - Continuous Integration ensures working code
 - MIT License
- Documentation on Read the Docs
 - References to literature
 - Explanation of calculations
 - Auto-generated from the source code
- Jupyter Notebooks demonstrate functionality
 - Notebooks can run on Binder, Deepnote and Colaboratory
- Docker images for quick deployment
 - Publicly available at Docker Hub
- Available on PyPI the python package registry



The screenshot shows the GitHub repository page for 'pdkit 1.2'. The repository is titled 'Parkinson's Disease Kit'. The page includes a search bar, navigation links (Help, Donate, Log in, Register), and a 'Latest version' badge. The main content area is divided into several sections: 'Navigation' with links to Project description, Release history, and Download files; 'Project links' with a link to the Homepage; 'Statistics' showing 6 stars, 5 forks, and 0 open issues/PRs; 'Meta' information including the MIT License and author 'G. Ross'; 'Project description' with badges for CI status (all green); 'PDKIT' title; 'INSTALL INSTRUCTIONS' section with 'Regular install' and 'For "editable" install' instructions; and 'For development install' instructions.

```
Search projects
```

Help Donate Log in Register

pdkit 1.2

pip install pdkit

Latest version

Last released: Mar 5, 2019

Parkinson's Disease Kit

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Statistics

GitHub statistics:

- Stars: 6
- Forks: 5
- Open issues/PRs: 0

View statistics for this project via Libraries.io, or by using Google BigQuery

Meta

License: MIT License (MIT)

Author: G. Ross

Project description

CI status: tests: passing docs: passing

PDKIT

INSTALL INSTRUCTIONS

Regular install

```
$ pip install pdkit
```

or

```
$ pip install git+git://github.com/pdkit/pdkit.git
```

For "editable" install

```
$ pip install --e git://github.com/pdkit/pdkit.git#egg=pdkit
```

For development install

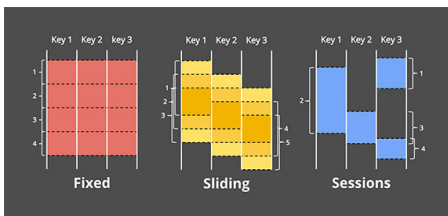
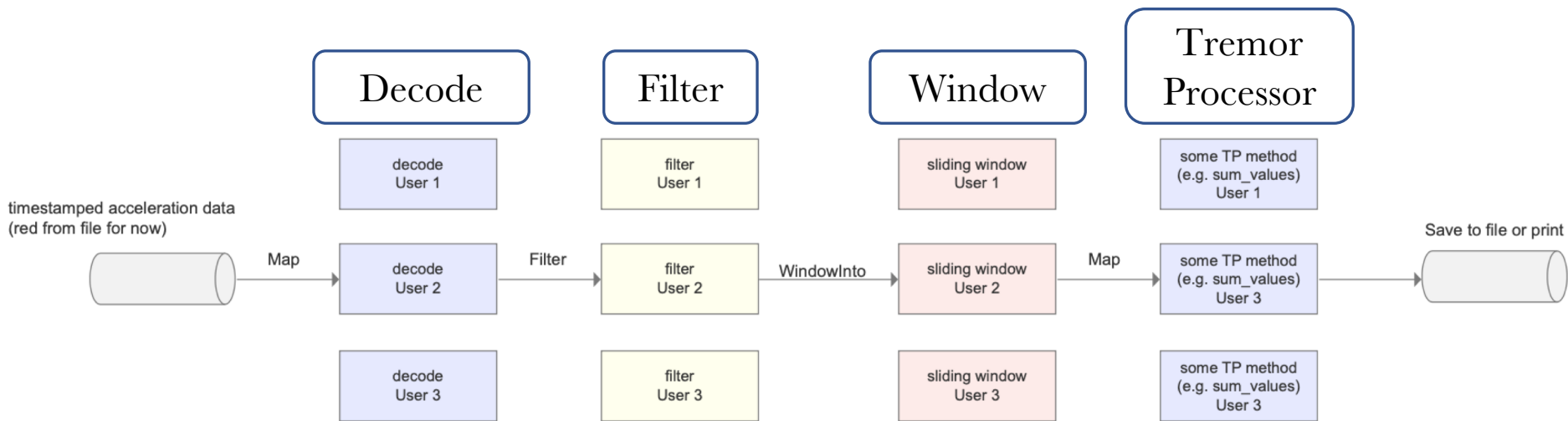
```
$ git clone https://github.com/pdkit/pdkit.git
```

- Advanced unified programming model:
 - stream processing from wearables
 - batch processing from smartphones
- Can deal with infinite out of order datasets
- Scaling out on cloud architectures
- Platform selected Apache BEAM
- Describe calculations as a directed graph which is mapped automatically by BEAM to the underlying infrastructure
- Infrastructure mapped using runners
- Demonstrate streaming PDKit



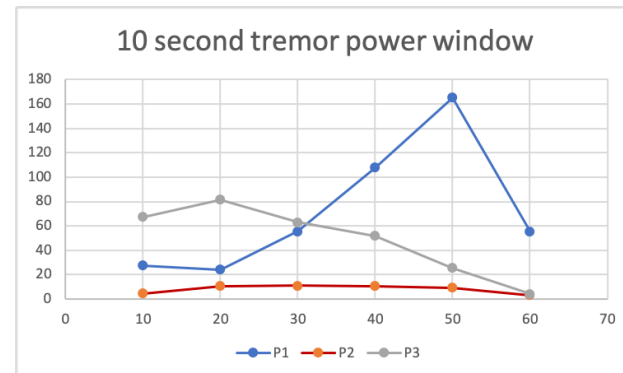
Source code <https://github.com/pdkit/pdkit-beam>

Tremor processing with Beam-PDkit

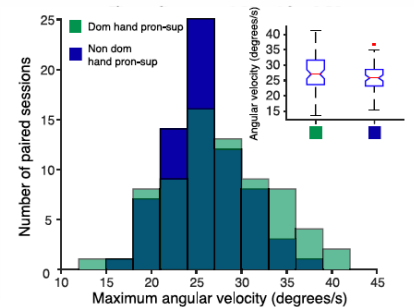


```
with beam.Pipeline(options=options) as p:
    messages = (p
        | 'Read from Stream' >> beam.io.gcp.pubsub.ReadFromPubSub(topic=known_args.input_topic)
        .with_output_types(bytes)
        )
    windowed_data = (messages
        | 'ParseMagSumAcc' >> ParseMagSumAcc(30,10)
        )
    grouped = (windowed_data
        | 'GroupWindowsByUser' >> beam.GroupByKey()
        | 'Calculate pdkit method' >> CalculatePDkitMethod()
        | 'UserScoresDict' >> beam.ParDo(UserDict())
        )
    welch = (grouped
        | 'Parse it' >> beam.Map(lambda elem: (elem['user'], elem['result'], elem['start'], elem['end'])))
    )
```

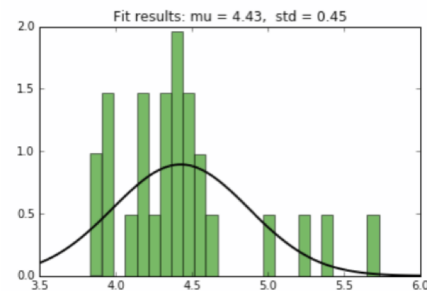
- Experiment:
 - 100,000 concurrent patients
 - 3d wrist sensor sampled at 50 Hz
 - 12 hours monitoring
 - Tremor power using Welch PSD calculated
- Show rapid tremor variations
- Total cost on GCE: \$ 9.43



- Observation: Any feature can be turned to a longitudinal biomarker
- Consider the sample distribution rather than individual measurements
- Dataflow processing is the key ingredient towards a process sampling paradigm
- Early evidence suggests significantly higher consistency and precision



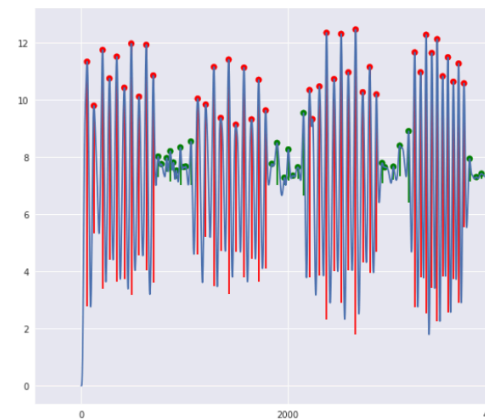
Pissadaki *et al* IBM Journal, 2018



July (43 samples)

- CUSSP at the UCL Institute of Neurology and Homerton Hospital (UK)
 - Details <https://clinicaltrials.gov/ct2/show/NCT02937324>
 - Data collection completed in May
 - 74 patients
- 20 lines of PDkit source code specify processing protocol
- 2-3 hours of software development
- Can recreate results in 1 hour on standard laptop
- Clinical results to be published this summer

- Development of gait biomarkers (except symmetry) present distinct challenges
- Better objective measures seem to be obtained by turning rather than straight line walking performance
- Fully automatic assessment of turning requires:
 - Signal segmentation with Bellman k-segmentation algorithm
 - Turn speed detection with Madgewick filter



- PDKit is an effective tool to precisely specify analytical processing framework of a study
- Open and inclusive development model
- Challenge: Published code not always properly licensed
- Challenge: Increase user and contributor network



Joan Saez



Cosmin Stamate



David Weston



Ashwani Jha

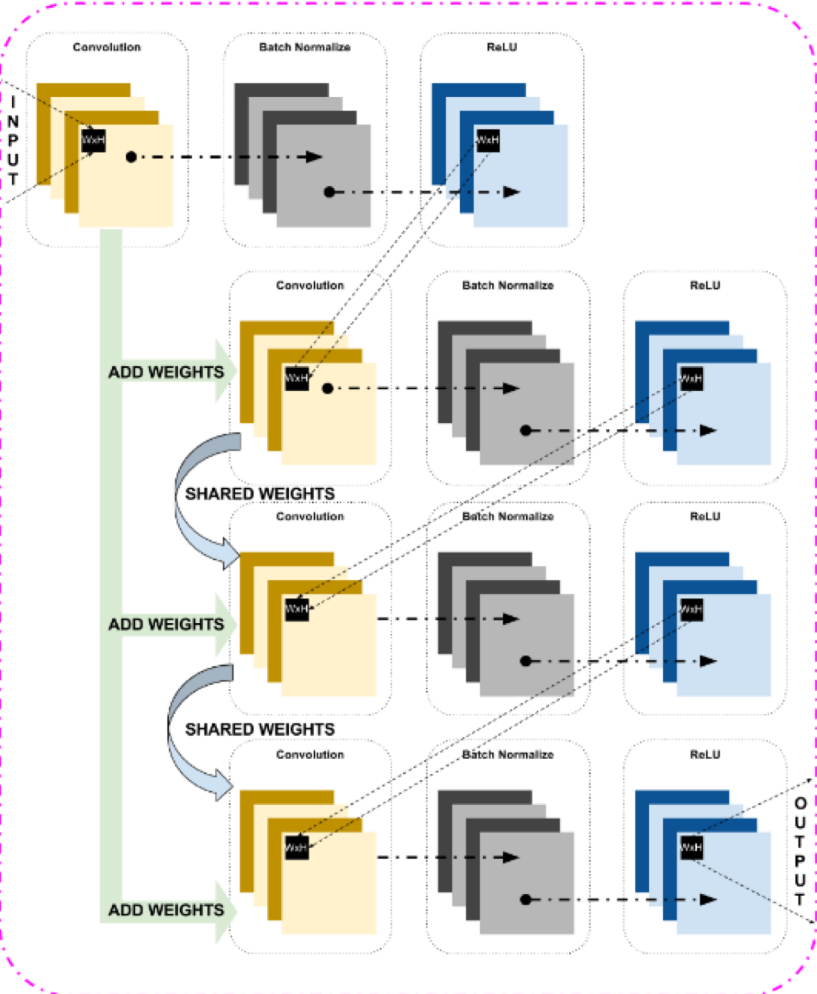
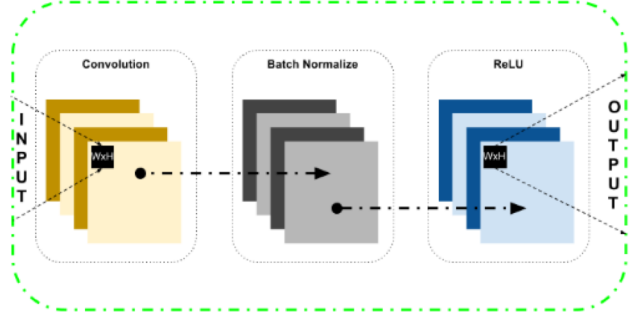
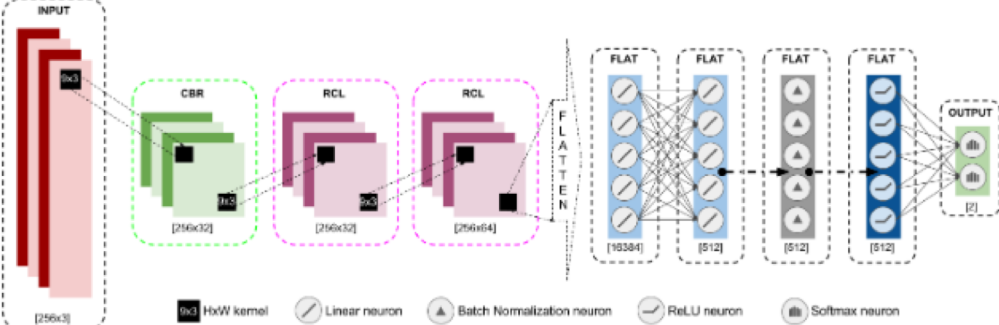


John Rothwell

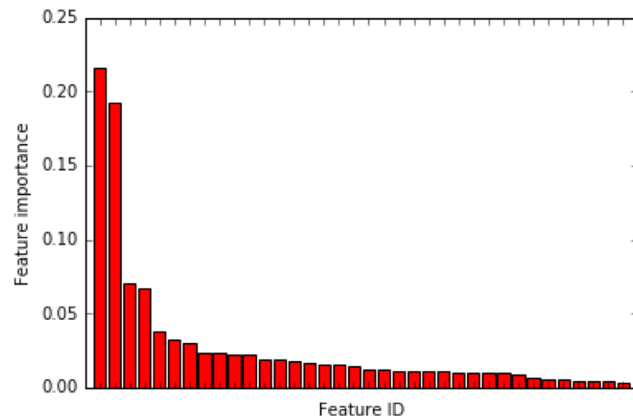


Nikos Fragopanagos

Deep Learning Architecture



- UPDRS exhaustive search of all possible symptoms
- Each patient presents only a few
- Symptoms typically change slowly e.g. 6 months
- ~6 features are predictive of overall score
- Use ML to identify the specific tests that offer the highest inferential power
 - Observer five full tests
 - Apply standard ensemble of randomized decision tree method to rank tests according to predictive strength
 - Select top 3 tests for individualised quick test



Signal quality assessment

