Mobile Health:
Practical 2

Machine Learning and Features of Health Data

Ian Tang
Sotirios Vavaroutas

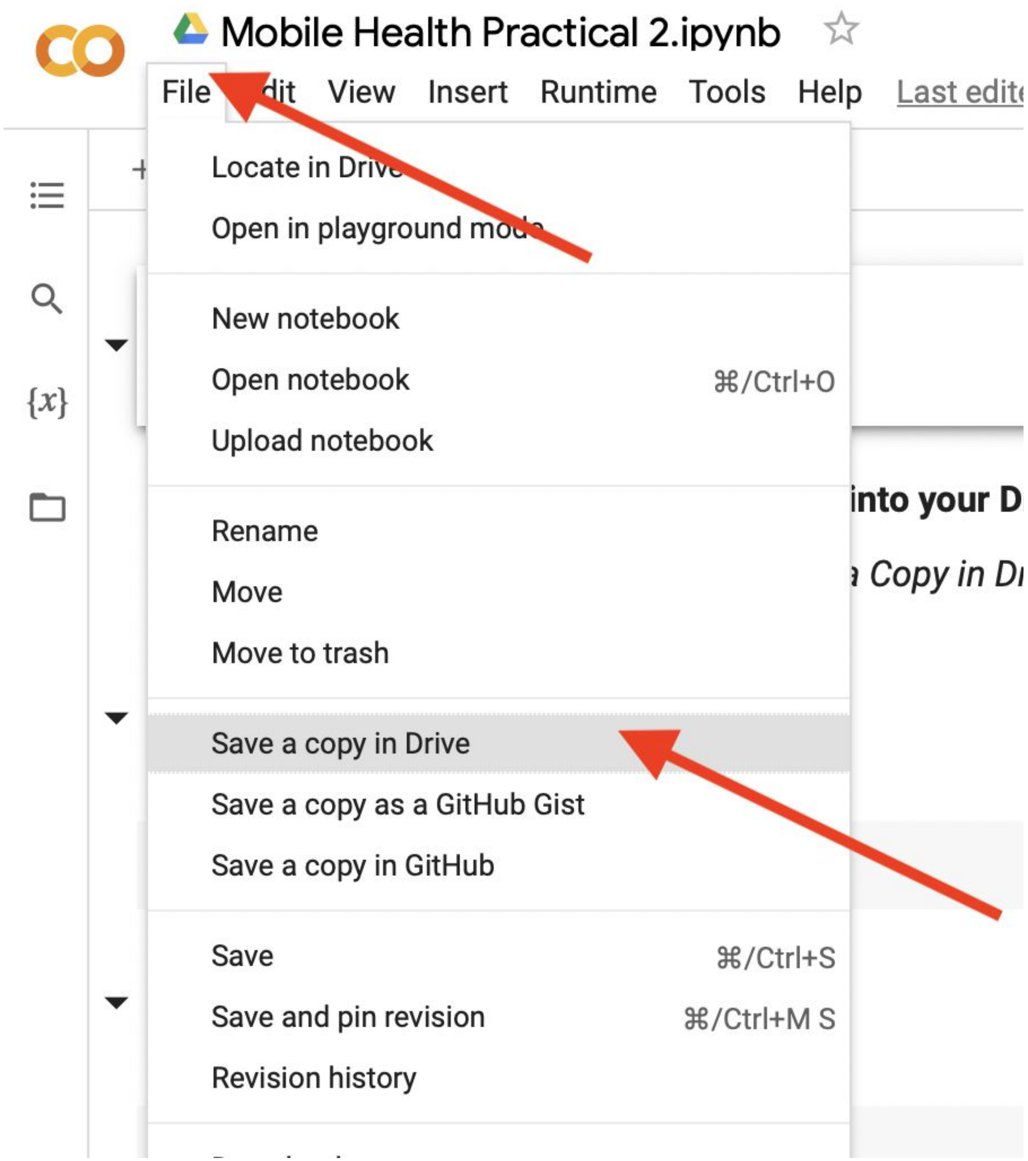
UNIVERSITY OF CAMBRIDGE

Colab Notebook

Please open the Colab notebook for today's session by visiting the relevant link on Moodle

Then, please save a copy of this notebook into your Drive:

- *File > Save a Copy in Drive*



UNIVERSITY OF CAMBRIDGE

Recap: IMU data

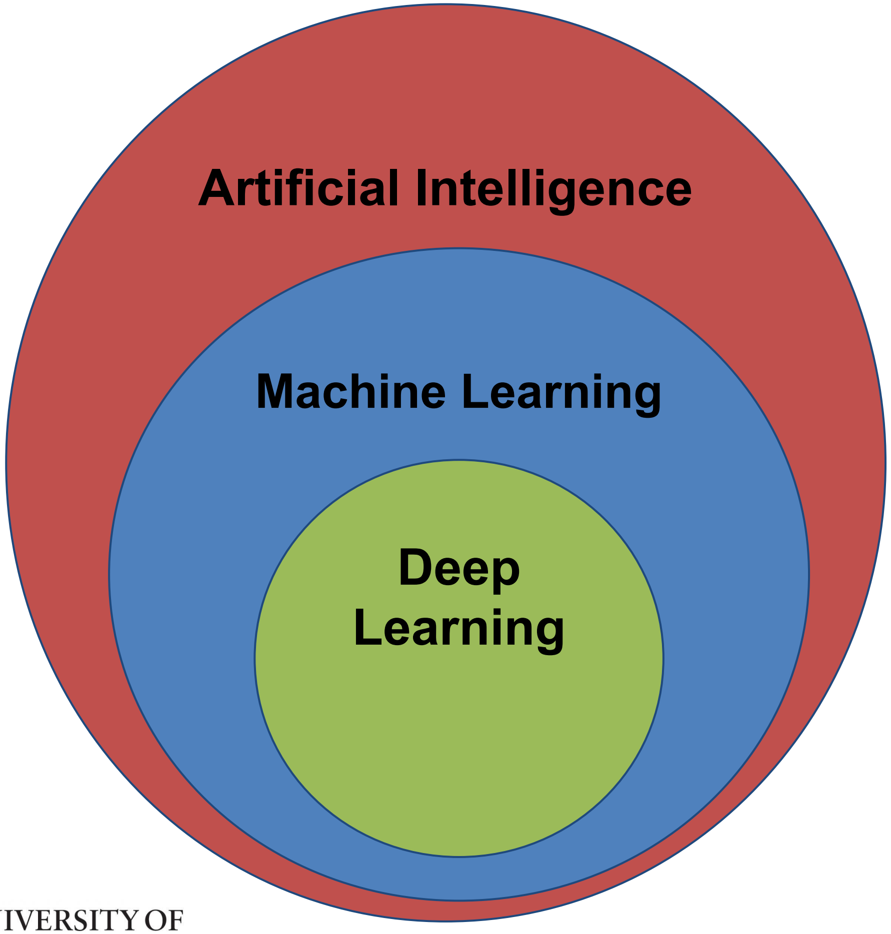
Inertial Measurement Unit (IMU) data is collected via:

- Accelerometers
- Gyroscopes
- Magnetometers

Its preprocessing may involve:

- Signal filtering (removing certain frequencies)
- Magnitude normalisation
- Localising temporal patterns of interest
- Mapping classes with windows

We will be focusing on Deep Learning

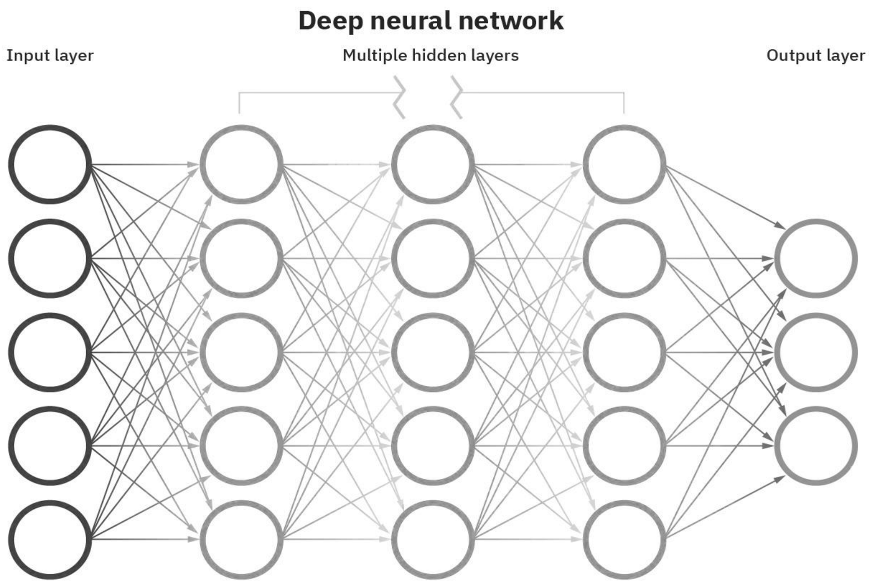


Traditional ML
Enables machines to "learn" how to undertake certain tasks with no human supervision

Deep Learning
Subset of ML targeted on building artificial neural networks

More on Deep Learning

- DL is implemented using Neural Networks
- **Layers** are the highest-level building blocks in DL:
  - They receive weighted inputs and transform them using mathematical functions
  - Each layer passes the transformed values as output to the next layer
  - Consequently, higher-level features are identified from lower-level features obtained at previous layers



UNIVERSITY OF CAMBRIDGE

MotionSense: IMU dataset we will be using

- Time-series data generated by accelerometer and gyroscope sensors
  - Altitude, gravity, user acceleration, rotation rate
- Collected with an iPhone 6s kept in the participant's front pocket
- All data collected in 50Hz sample rate
- 24 participants
- 6 activities in 15 trials in the same environment and conditions
  - Going downstairs, upstairs, walking, jogging, sitting, and standing

Malekzadeh, Mohammad & Clegg, Richard & Cavallaro, Andrea & Haddadi, Hamed. (2019). Mobile sensor data anonymization. Proceedings of the International Conference on Internet of Things Design and Implementation, 49-58.

Libraries to import to our Colab notebook

- `import tensorflow as tf`
- `import pandas as pd`
- `import numpy as np`
- `import sklearn.model_selection`
- `import sklearn.metrics`
- `import scipy`
- `import requests`
- `import zipfile`
- `import os`
- `import re`
- `import glob`

UNIVERSITY OF CAMBRIDGE

1.1 Downloading the Dataset

```
file_url = "..."
dataset_file_name = "B_Accelerometer_data.zip"
data_directory = "data"
accelerometer_data_folder_path = "data/B_Accelerometer_data/B_Accelerometer_data"

r = requests.get(file_url)

with open(dataset_file_name, 'wb') as f:
   f.write(r.content)

with zipfile.ZipFile(dataset_file_name, 'r') as zip_ref:
   zip_ref.extractall(os.path.join(data_directory, dataset_file_name.split(".")[0]))
```

UNIVERSITY OF CAMBRIDGE

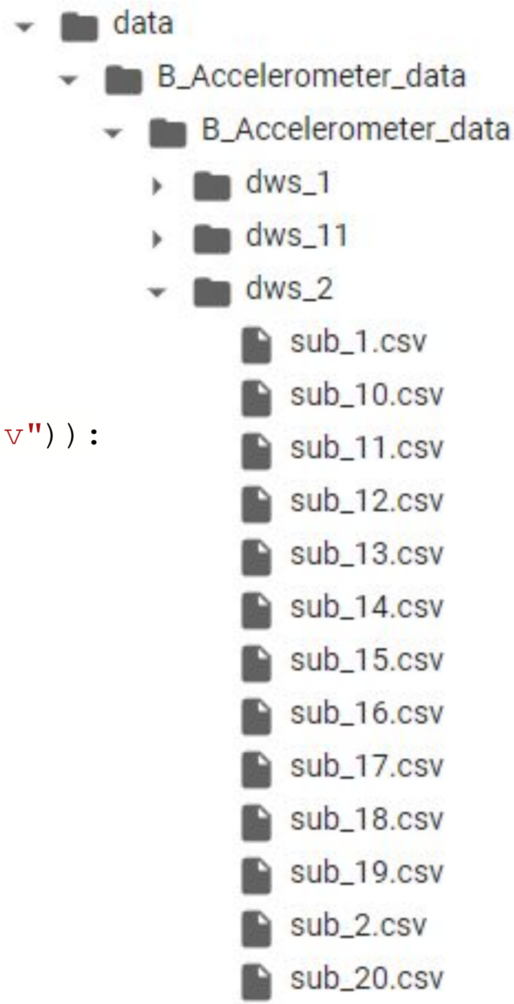
Labels in this Dataset

There are 6 different labels:

- **dws:** downstairs
- **ups:** upstairs
- **sit:** sitting
- **std:** standing
- **wlk:** walking
- **jog:** jogging

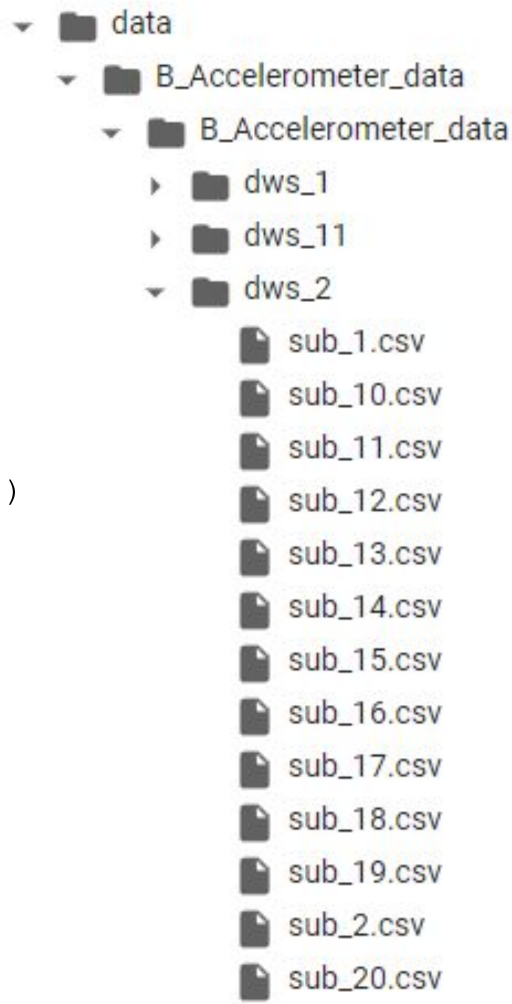
1.2 Loading the Dataset to Memory

```python
# Loop through every trial folder
for trial_folder in all_trials_folders:

    # Get the label of the trial
    ...

    # Loop through files for every user of the trail
    for trial_user_file in sorted(glob.glob(trial_folder + "/*.csv")):

        # Use regex to match the user id
        ...
            # Read file
            ...

            # Extract the x, y, z channels
            values = user_trial_dataset[["x", "y", "z"]].values
```

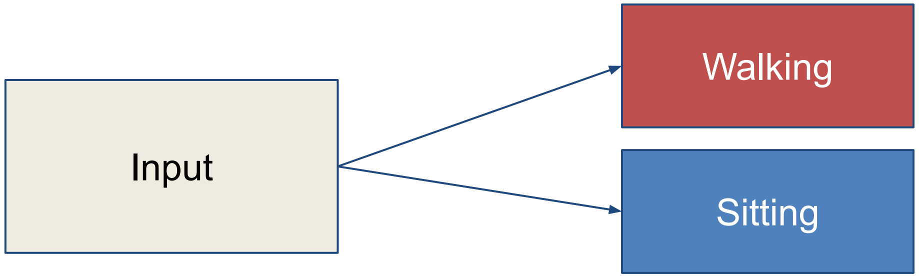


1.2 Loading the Dataset to Memory

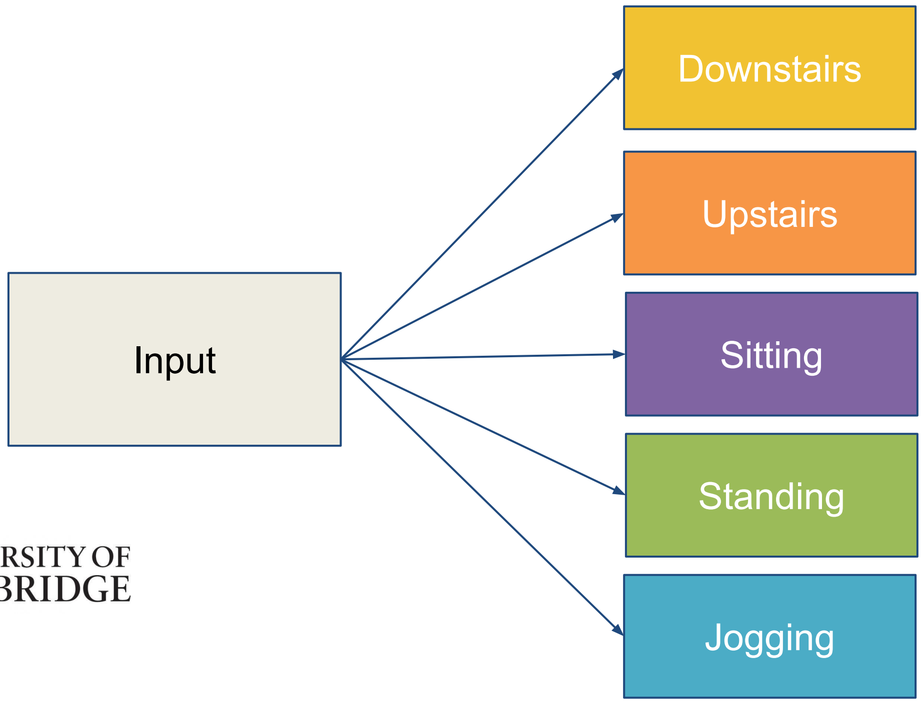
```
            # The label is the same during the entire trial,
            # so it is repeated here to pad to the same length
            # as the values
            labels = np.repeat(label, values.shape[0])

            if user_id not in user_split_datasets:
                user_split_datasets[user_id] = []
            user_split_datasets[user_id].append((values, labels))
        else:
            print("[ERR] User id not found", trial_user_file)

label_list = list(label_set.keys())
user_split_datasets.keys()
```



Potential tasks that can be performed



Binary Classification
Data is classified into two mutually exclusive groups



Multi-Class Classification
Data is classified into three or more groups

UNIVERSITY OF CAMBRIDGE

Splitting the dataset: train/validation/test

- **Training set:**
  - Data used to fit the model
- **Validation set:**
  - Data used to evaluate the model while tuning model's hyperparameters
  - Can lead to biases as the "knowledge" of the validation set can indirectly affect the training
- **Test set:**
  - Data used to evaluate success of the final model
  - Fully unbiased

How to partition our dataset into the subsets for optimal evaluation?

UNIVERSITY OF CAMBRIDGE

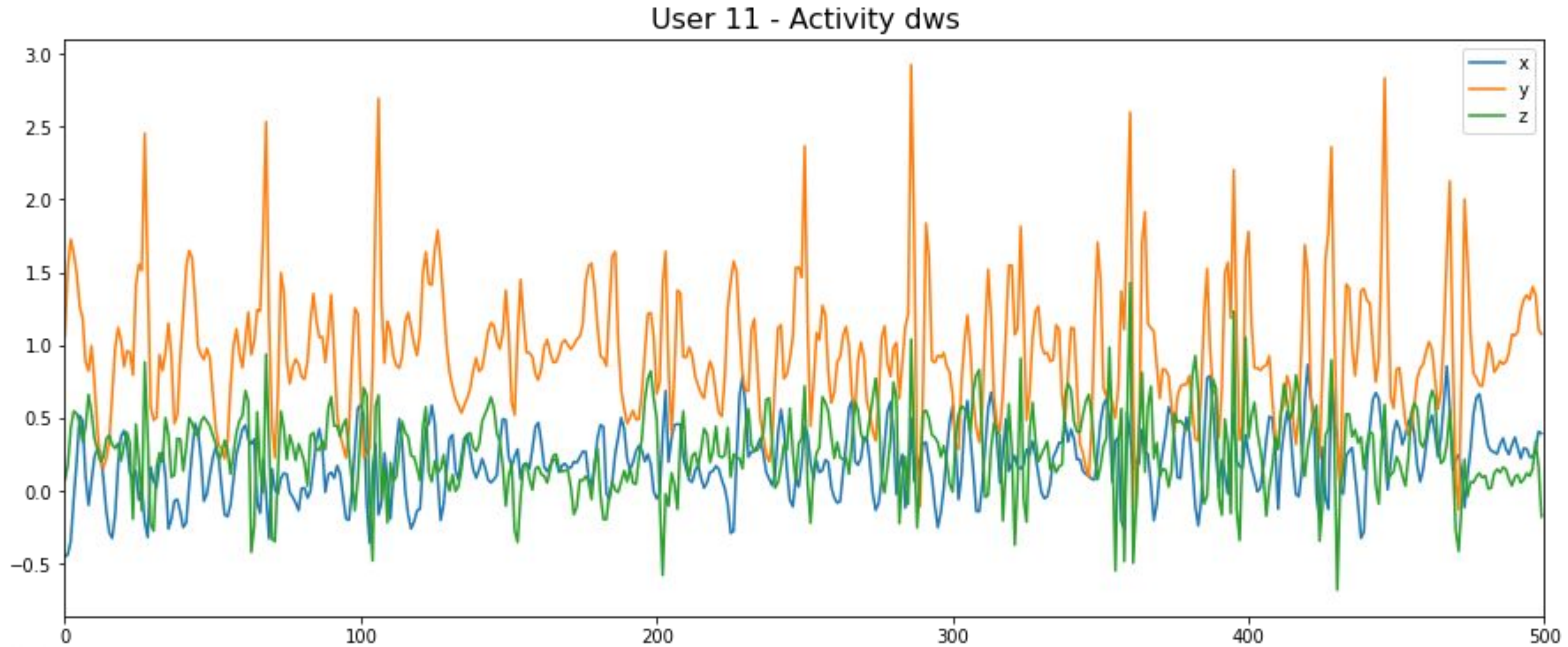
Pre-processing the dataset

We need to:

1. Use a sliding window to make a windowed dataset
2. Split the dataset into a training and a test set
3. Normalise the datasets
4. Apply label encoding
5. Subdivide the training set into training and validation sets

1.3 Visualising the Data

```
timeseries, labels = user_split_datasets[user_id][session]
plot_accelerometer_timeseries(timeseries[:500])
```



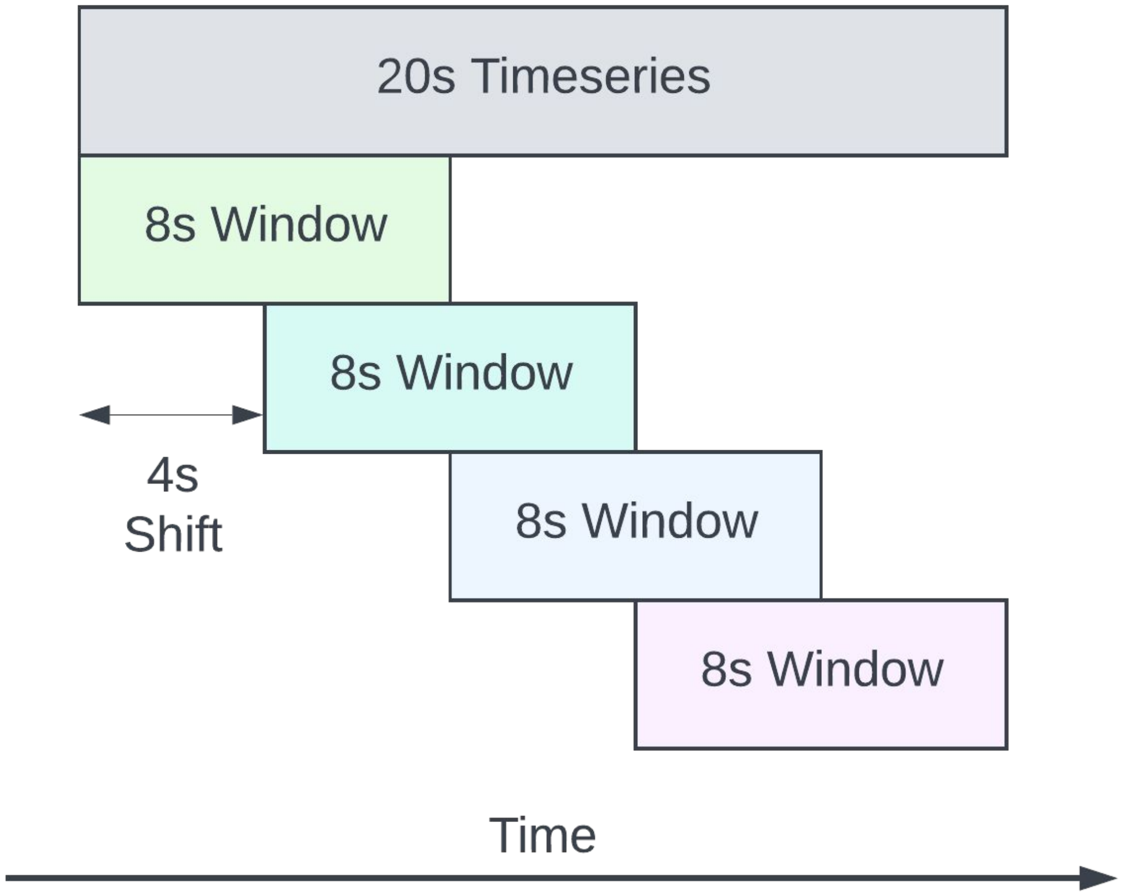
UNIVERSITY OF CAMBRIDGE

Pre-processing the dataset

We need to:

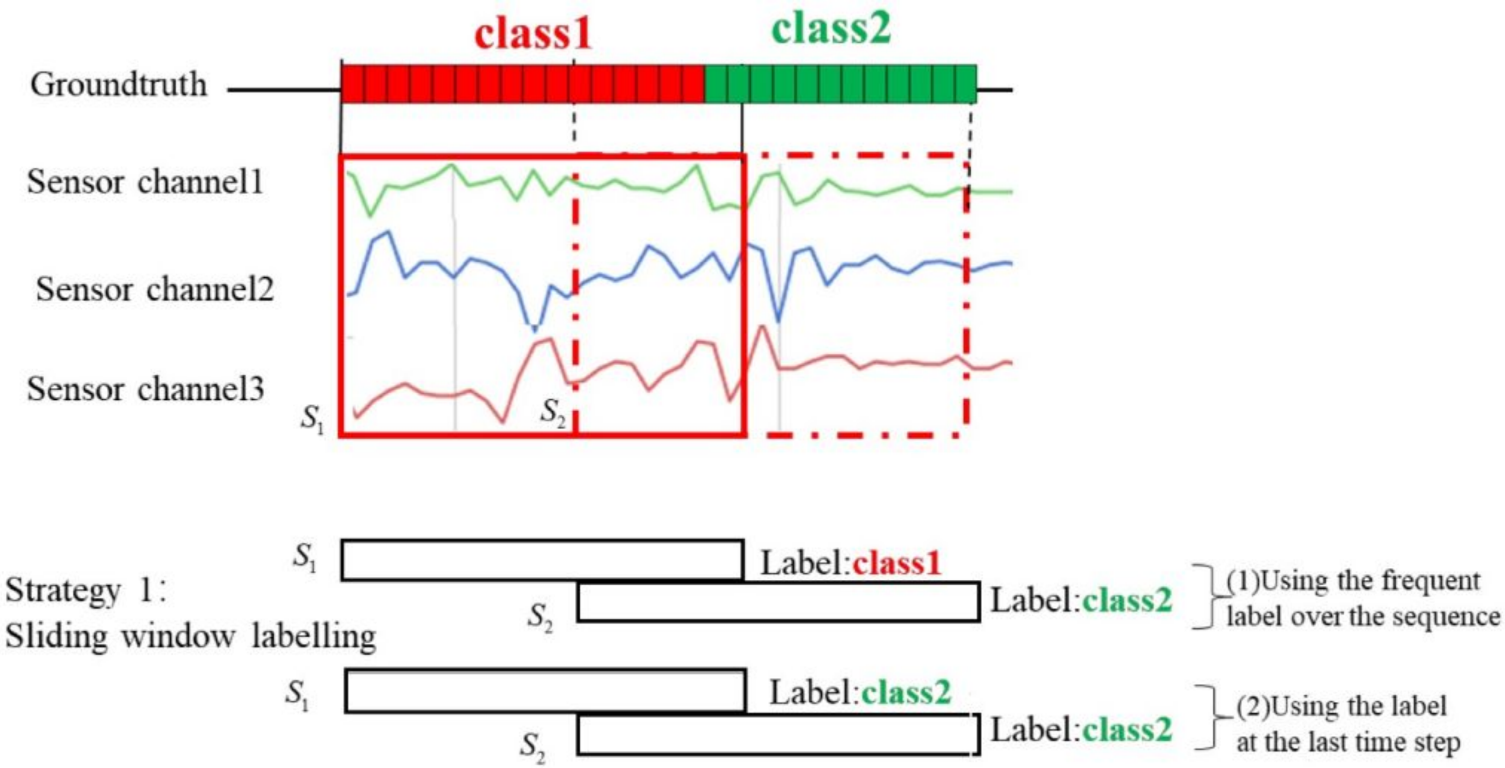
1. **Use a sliding window to make a windowed dataset**
2. Split the dataset into a training and a test set
3. Normalise the datasets
4. Apply label encoding
5. Subdivide the training set into training and validation sets

2.1 Sliding Window



UNIVERSITY OF CAMBRIDGE

2.1 Sliding Window



UNIVERSITY OF CAMBRIDGE

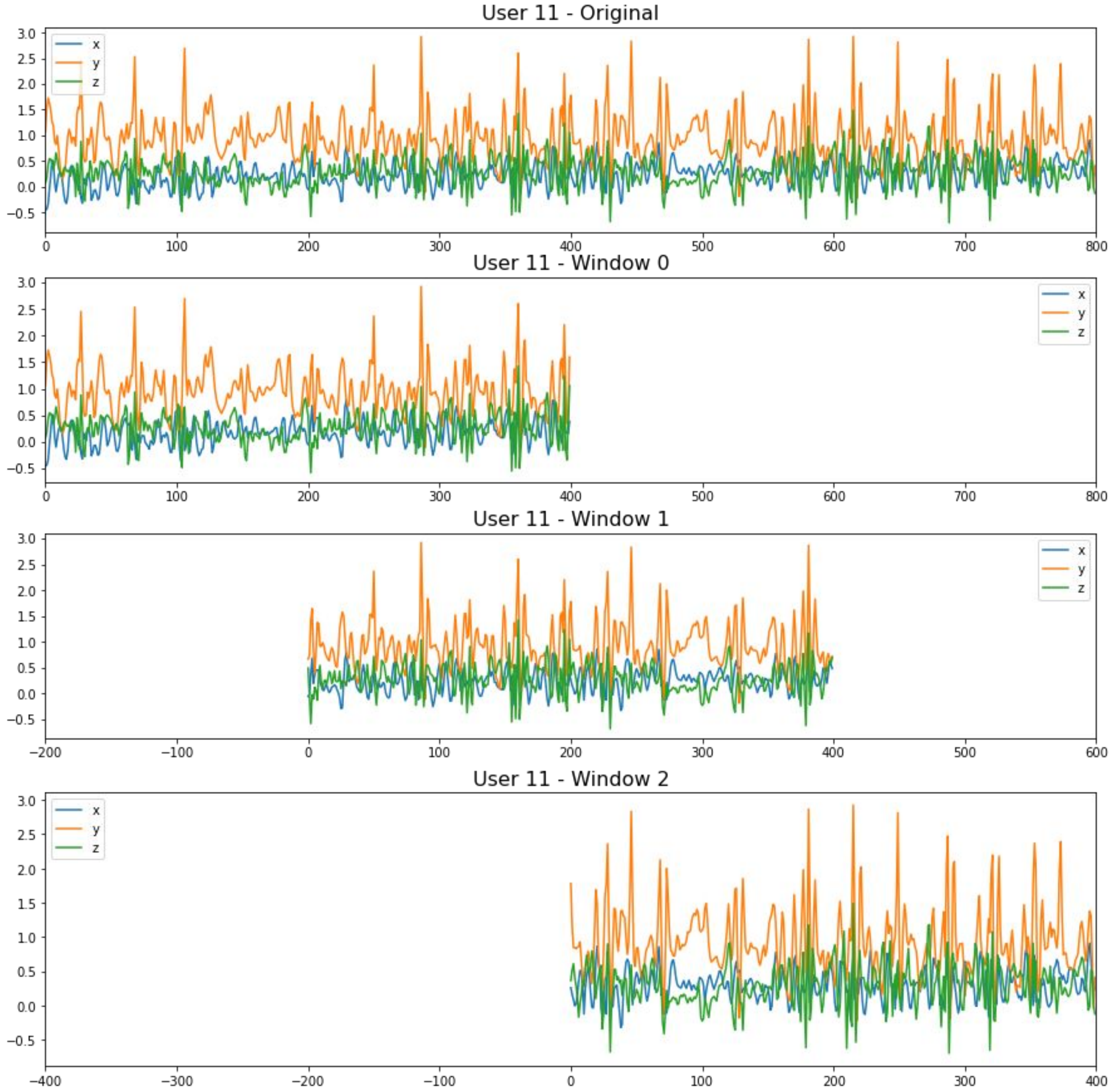
Zhang, Yong & Zhang, Yu & Zhang, Zhao & Bao, Jie & Song, Yunpeng. (2018). Human activity recognition based on time series analysis using U-Net.

2.1 Making a Windowed Dataset

```
def get_windows_dataset_from_user_list_format(user_datasets, window_size=400,
shift=200, stride=1, verbose=0)
```

```
window_size = 400
user_datasets_windowed =
        get_windows_dataset_from_user_list_format(
            user_split_datasets,
            window_size=window_size,
            shift=window_size//2
        )
```

2.1.1 Windowed Dataset



Pre-processing the dataset

We need to:

1. Use a sliding window to make a windowed dataset
2. **Split the dataset into a training and a test set**
3. Normalise the datasets
4. Apply label encoding
5. Subdivide the training set into training and validation sets

UNIVERSITY OF CAMBRIDGE

2.2 Training & Testing Split

```
def combine_windowed_dataset(user_datasets_windowed, train_users, test_users =
None, verbose = 0)
```

```
Training set
+  (280, 400, 3)        Samples from User 10
+  (279, 400, 3)        Samples from User 11
+  (248, 400, 3)        Samples from User 12
+  (238, 400, 3)        Samples from User 13
+  (281, 400, 3)        Samples from User 15
+  (311, 400, 3)        Samples from User 16
+  (258, 400, 3)        Samples from User 17
+  (297, 400, 3)        Samples from User 18
+  (292, 400, 3)        Samples from User 2
+  (259, 400, 3)        Samples from User 20
+  (327, 400, 3)        Samples from User 21
+  (263, 400, 3)        Samples from User 22
+  (238, 400, 3)        Samples from User 24
+  (295, 400, 3)        Samples from User 3
+  (262, 400, 3)        Samples from User 4
+  (244, 400, 3)        Samples from User 5
+  (287, 400, 3)        Samples from User 7
+  (285, 400, 3)        Samples from User 8
+  (267, 400, 3)        Samples from User 9
--------------------------------------------
=  (5211, 400, 3)       Samples
```

```
Test set
+  (293, 400, 3)        Samples from User 1
+  (263, 400, 3)        Samples from User 14
+  (343, 400, 3)        Samples from User 19
+  (252, 400, 3)        Samples from User 23
+  (268, 400, 3)        Samples from User 6
--------------------------------------------
=  (1419, 400, 3)       Samples
```

Pre-processing the dataset

We need to:

1. Use a sliding window to make a windowed dataset
2. Split the dataset into a training and a test set
3. **Normalise the datasets**
4. Apply label encoding
5. Subdivide the training set into training and validation sets

2.3 Normalisation

Pre-processing the dataset

We need to:

1. Use a sliding window to make a windowed dataset
2. Split the dataset into a training and a test set
3. Normalise the datasets
4. **Apply label encoding**
5. Subdivide the training set into training and validation sets

2.4 Label Encoding

```
def apply_label_map(y, label_map)
```

```
Mapping
dws --> 0
jog --> 1
sit --> 2
std --> 3
ups --> 4
wlk --> 5
```

```
Original: ['dws' 'wlk' 'sit' 'wlk' 'std' 'sit']
Mapped:   [0 5 2 5 3 2]
```

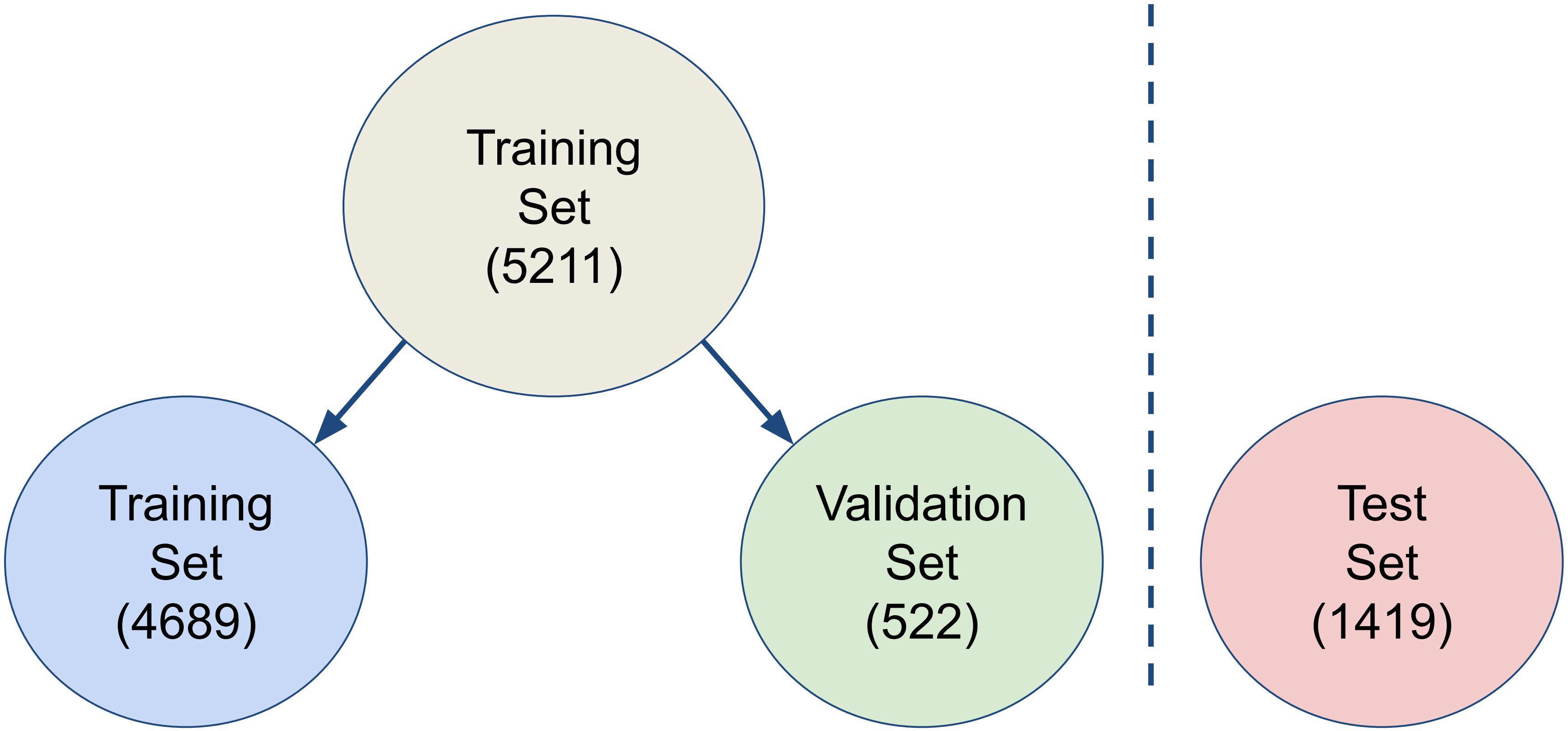
Pre-processing the dataset

We need to:

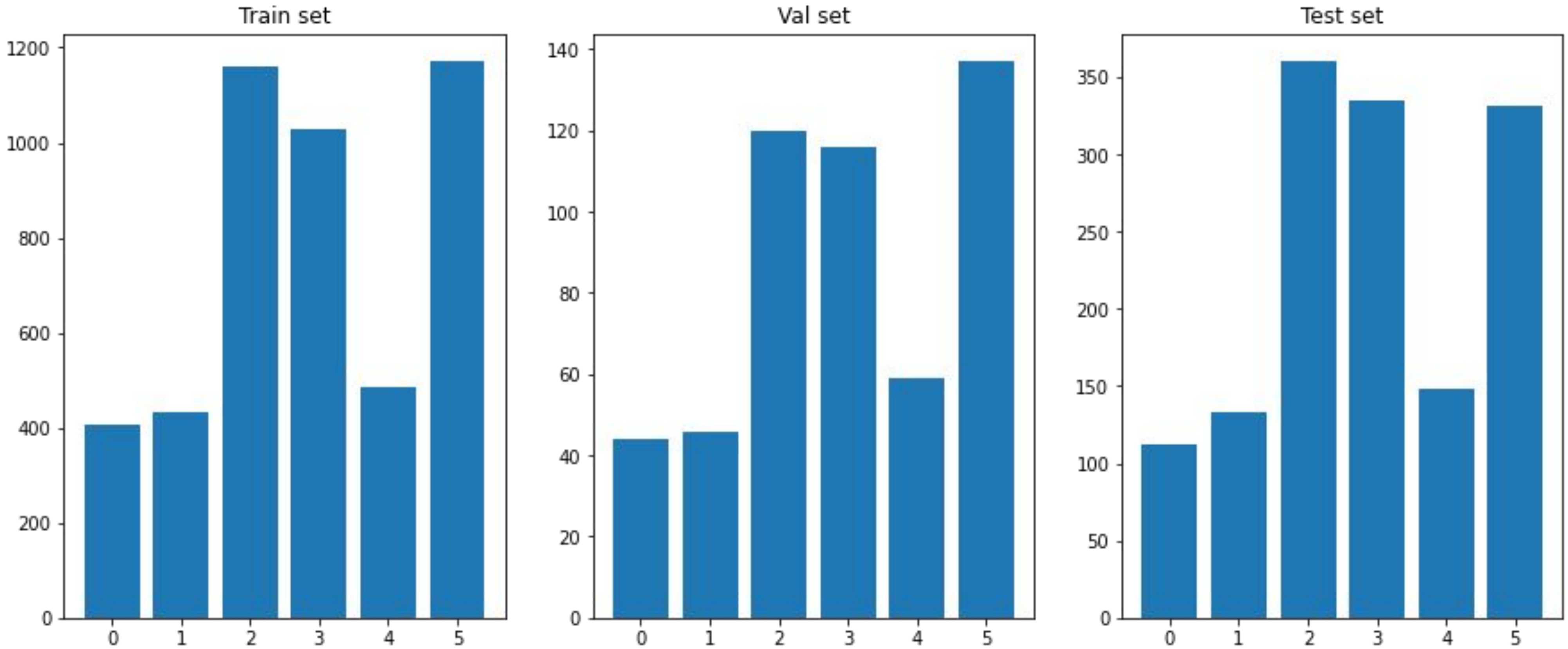
1. Use a sliding window to make a windowed dataset
2. Split the dataset into a training and a test set
3. Normalise the datasets
4. Apply label encoding
5. **Subdivide the training set into training and validation sets**

2.5 Training & Validation Split

`sklearn.model_selection.train_test_split`



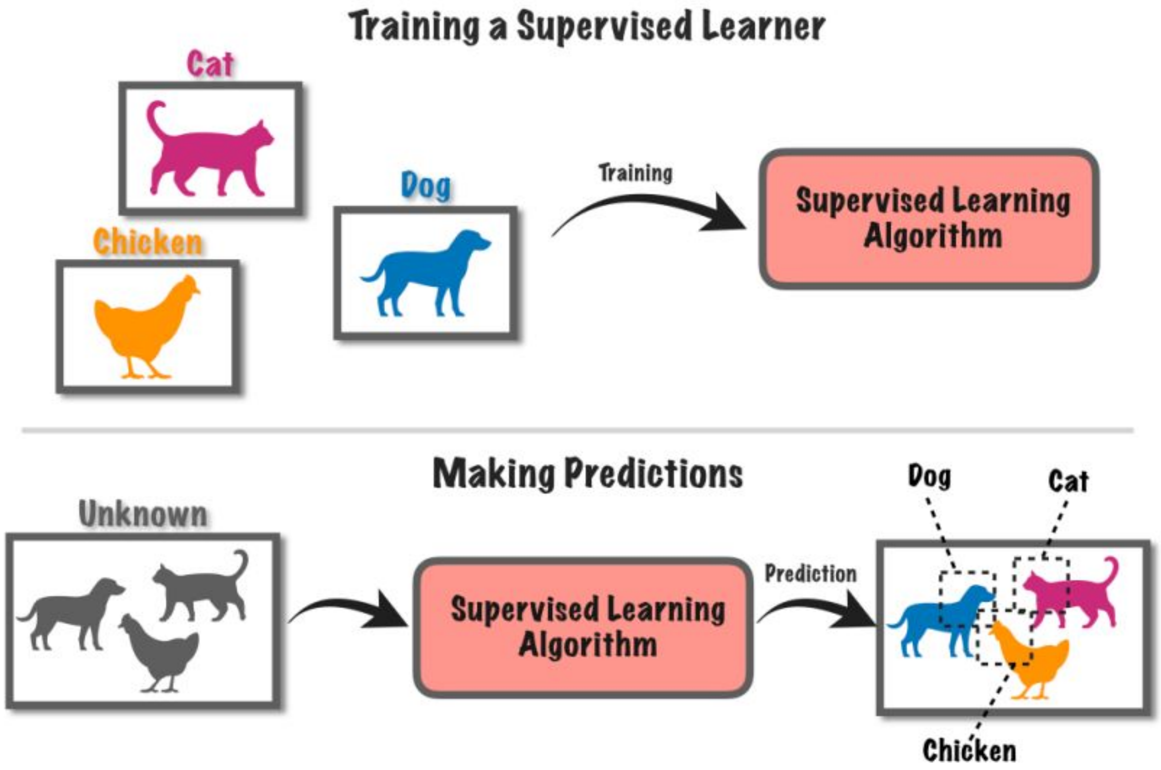
2.6 Label Distribution



UNIVERSITY OF CAMBRIDGE

3 Deep Learning

- Supervised Learning



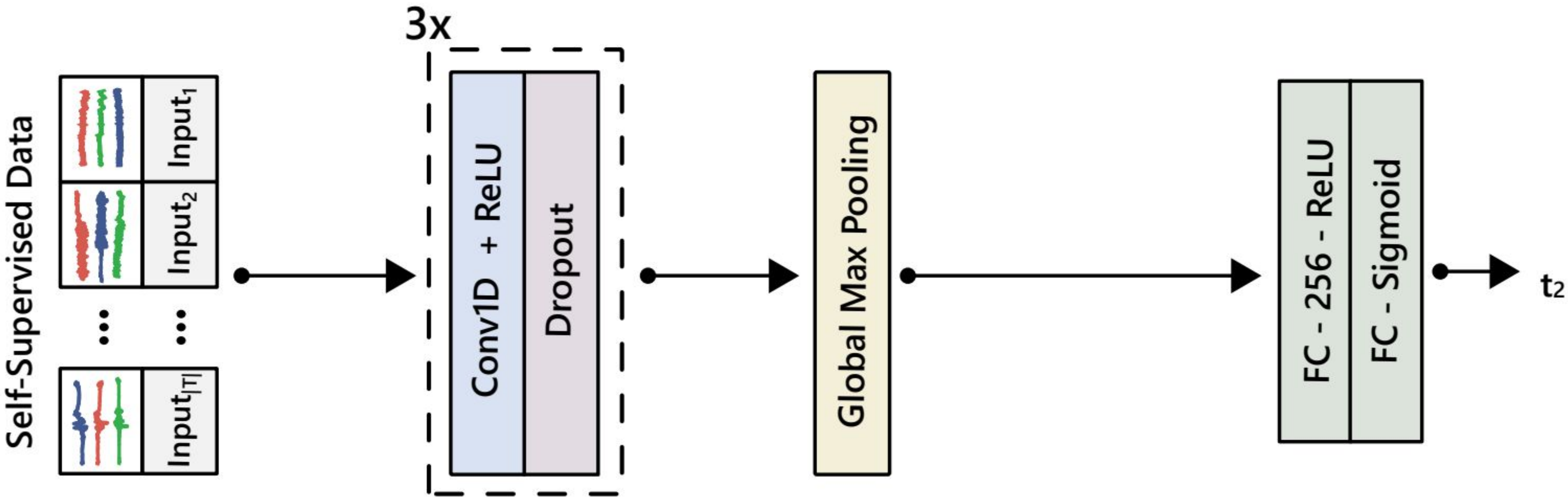

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

UNIVERSITY OF CAMBRIDGE

Jeffares, A. (2018). Supervised vs Unsupervised Learning in 3 Minutes [infographic]. https://towardsdatascience.com/supervised-vs-unsupervised-learning-in-2-minutes-72dad148f242

3.1 Building a CNN

- Transformation Prediction Network (TPN)



Saeed, A., Ozcelebi, T., & Lukkien, J. (2019). Multi-task self-supervised learning for human activity detection. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 3(2), 1-30.

3.1 Building a CNN

- Transformation Prediction Network (TPN)

```
model = tf.keras.Sequential([
    tf.keras.Input(shape=train_x.shape[1:], name='input'),
    tf.keras.layers.Conv1D(32, 24, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(l=1e-4)),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Conv1D(64, 16, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(l=1e-4)),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Conv1D(96, 8, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(l=1e-4)),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.GlobalMaxPool1D(data_format='channels_last', name='global_max_pooling1d'),
    tf.keras.layers.Dense(1000, activation='relu'),
    tf.keras.layers.Dense(output_shape),
    tf.keras.layers.Softmax()
])
```


Saeed, A., Ozcelebi, T., & Lukkien, J. (2019). Multi-task self-supervised learning for human activity detection. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 3(2), 1-30.

3.2 Training a Neural Network

```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.003)
model.compile(
    optimizer=optimizer,
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=['accuracy']
)
```

```
history = model.fit(
    x=train_set[0],
    y=train_set[1],
    validation_data=val_set,
    batch_size=256,
    shuffle=True,
    epochs=30
)
```

3.2 Training History

Slide redacted due to this being a live coding exercise to be completed during the practical

3.2 Training History

Slide redacted due to this being a live coding exercise to be completed during the practical

UNIVERSITY OF CAMBRIDGE

3.3 Evaluation

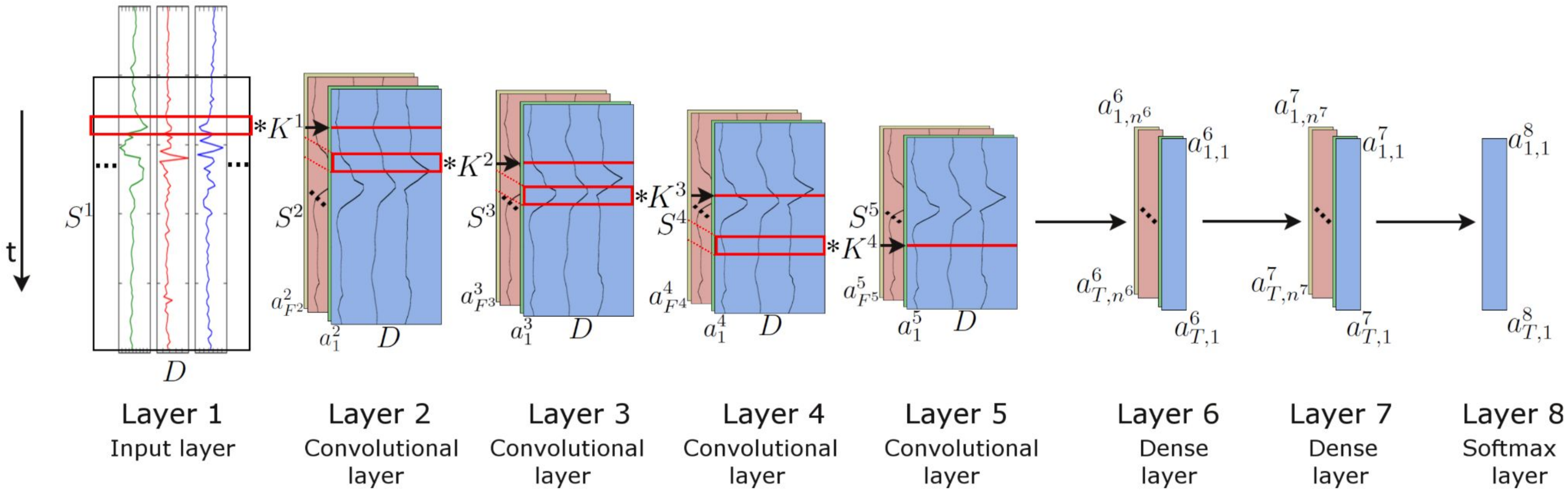
Slide redacted due to this being a live coding exercise to be completed during the practical

UNIVERSITY OF CAMBRIDGE

3.3 Evaluation

Slide redacted due to this being a live coding exercise to be completed during the practical

4 Exploration: DeepConvLSTM



Ordóñez, F. J., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1), 115.

Assignment 2

- Released today, due on the **17th of March**
- Weighting: **70%** of the course grade
- **Part II:** Colab notebook and reflection report of 1,000 words
- **Part III/MPhil:** Colab notebook and a reflection report of 1,500 words
- Please use the help forum on Moodle for any questions

Assessment 2 Forum

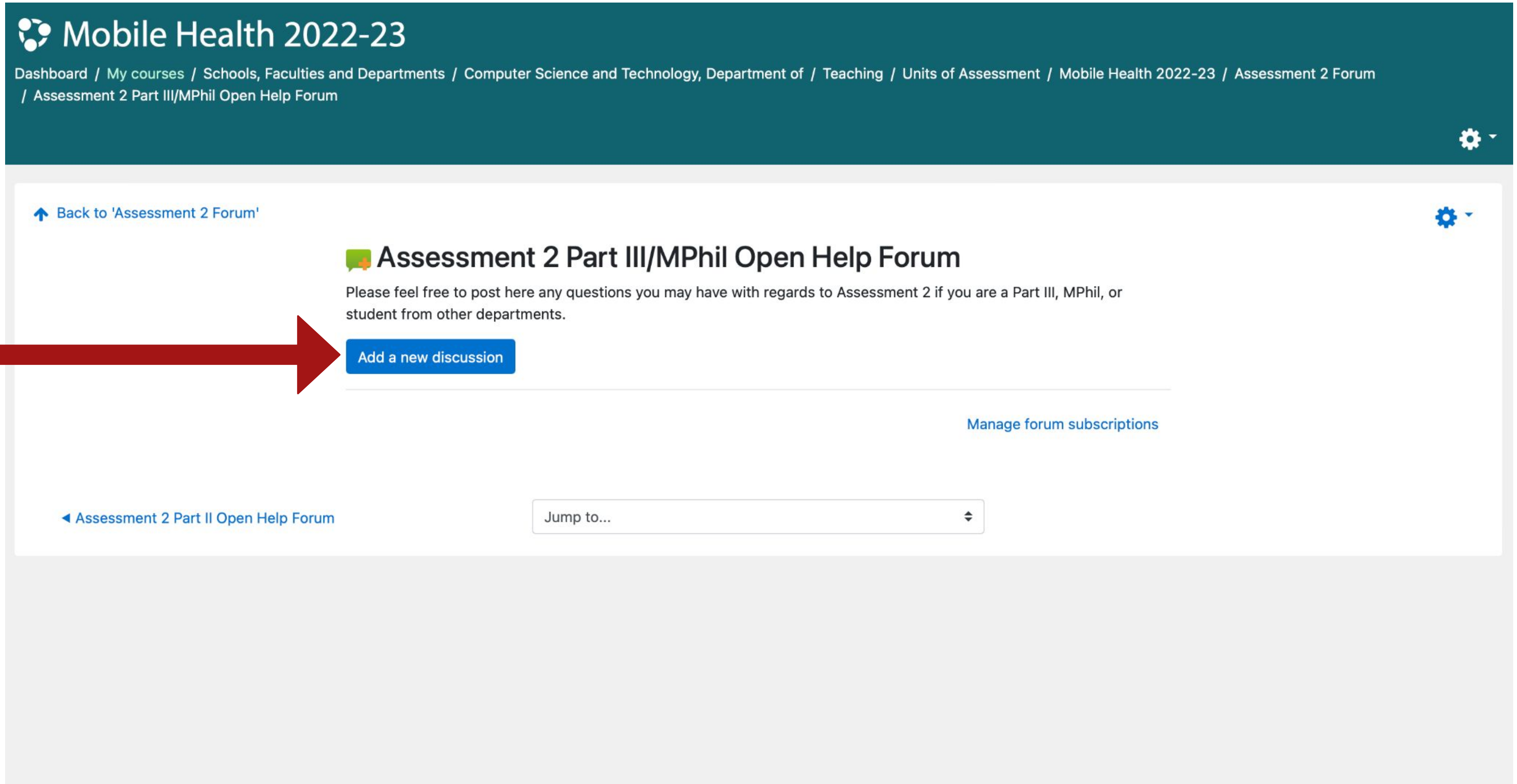
Assessment 2 Part II Open Help Forum

Please feel free to post here any questions you may have with regards to Assessment 2 if you are a Part II student.

Assessment 2 Part III/MPhil Open Help Forum

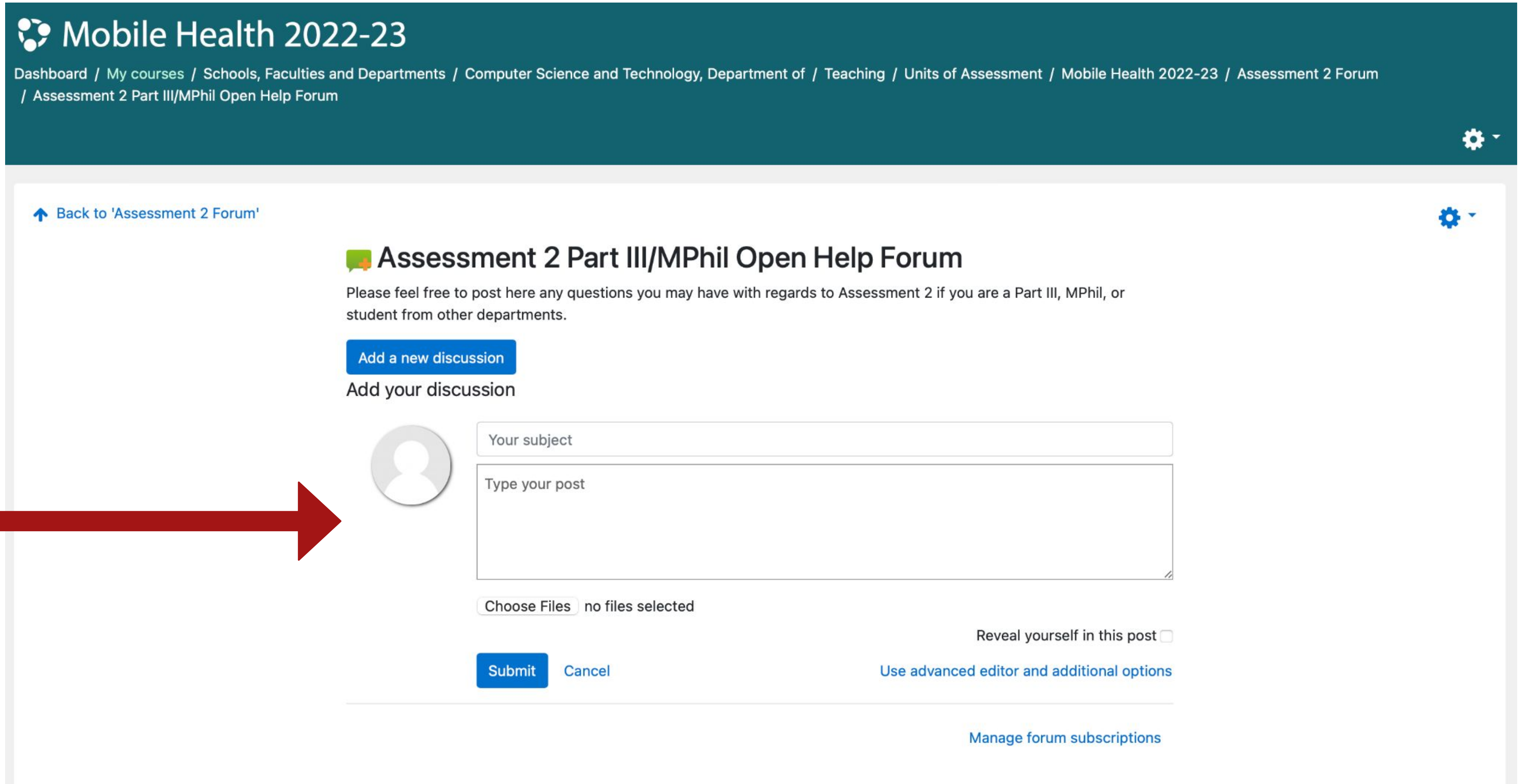
Please feel free to post here any questions you may have with regards to Assessment 2 if you are a Part III, MPhil, or student from other departments.

Open Help Forum

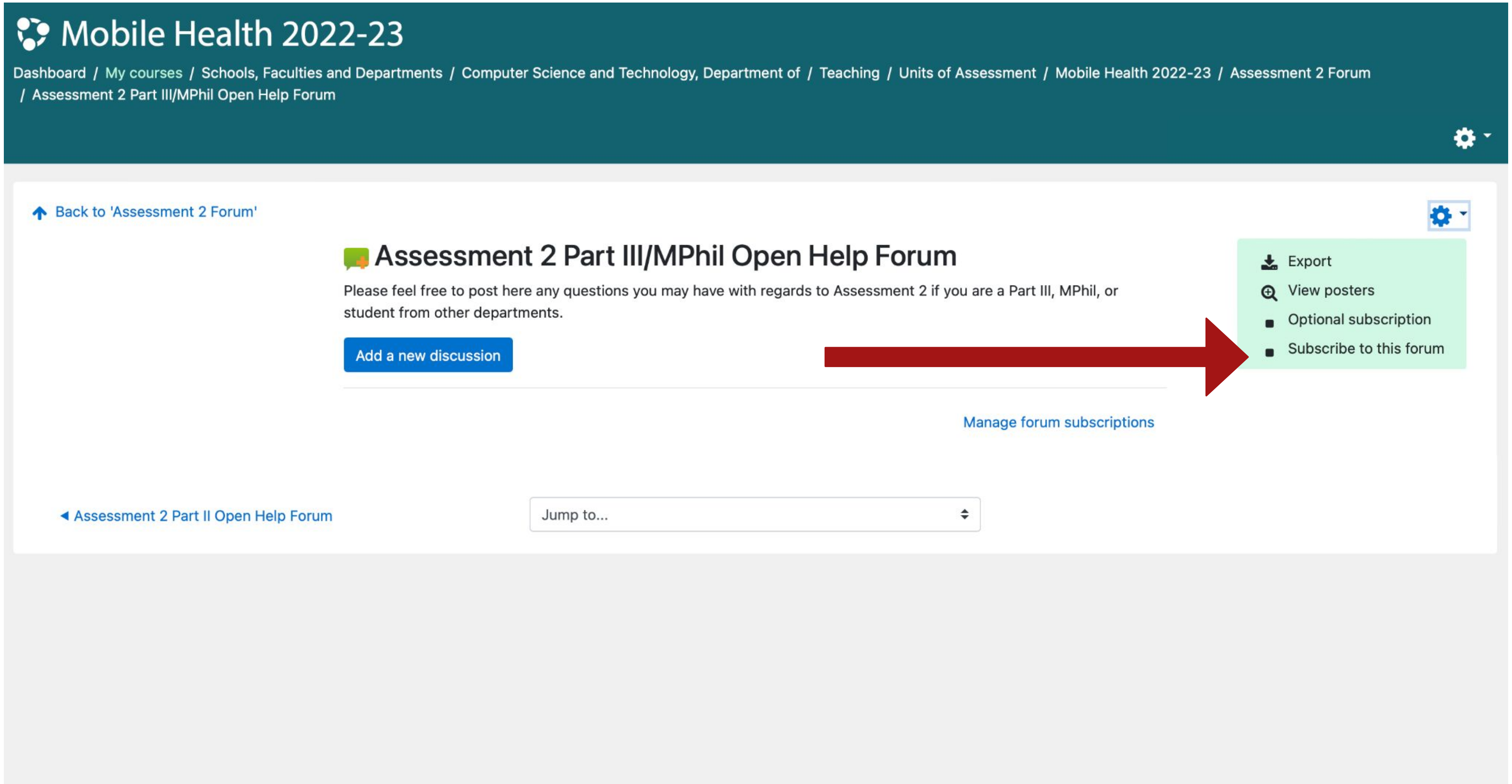


UNIVERSITY OF CAMBRIDGE

Open Help Forum



Open Help Forum



UNIVERSITY OF CAMBRIDGE

Questions