Task 0:

For this task, three univariate time series were selected to be analyzed, they are:

1. **BasicMotions :** this data is a 3D accelerometer's and 3D gyroscope's data from four students activities. It consist of four classes which are standing, walking, running and playing badminton.
2. **AtrialFibrillation :** The dataset described is a collection of two-channel ECG recordings, specifically designed for the Computers in Cardiology Challenge 2004. It was aimed at advancing automated methods to predict the spontaneous termination of atrial fibrillation (AF). The dataset comprises 5-second segments of AF, with each channel sampled at 128 samples per second, structured in a multivariate format. It contains three classes that is n, s, and t. n stands for non-termination, s stands for self-termination. And t stands for Immediate termination.
3. **ItalyPowerDemand :** The data was derived from twelve monthly electrical power demand time series from Italy. This is univariate series where length of dataset is 24 and class of this dataset is 2.

Task 1

In this task I build four models such as one directional RNN, 1D-CNN, 1D-CNN + GRU and fully connected network to test three datasets such as PowerCons, Strawberry and BeetleFly datasets. Performance of these models shown below.

Performance table:

Model	BasicMotions	ItalyPowerDemand	AtrialFibrillation	Average test score.
FCN	0.6	0.94	0.4	0.6
1D-CNN	0.77	0.85	0.26	0.62
1-D RNN	0.65	0.95	0.4	0.67
1D-CNN + GRU	1.0	0.92	0.26	0.73

Best model Highlight: CNN+GRU model is the best model with the average test accuracy of 0.73

Model Architecture:

Fully connected Network: The model constructed using tensorflow's keras API where used a flatten Layer and hidden five dense layer. Also experimented with different activation functions and regularization technique to enhance the better performance. During training, I employed techniques like early stopping and use LearningRateScheduler to enhance better performance.

1D-CNN architecture: The model constructed using tensorflow's keras API where used conv1d, Maxpooling1d Flatten layer and finally two hidden dense layer. Also experimented with different activation functions and regularization technique.

1D-RNN architecture: The model constructed using tensorflow's keras API Where used a simpleRNN Layer and three hidden dense layer. Also experimented by different activation function (i.e. relu and softmax).

1D-CNN plus GRU: The model constructed using tensorflow's keras API Where used 1D CNN , two GRU layer and couple of dense layer . Also experimented by different activation functions (is.e. relu and softmax).

Task 2

In this task I build two models and one CNNclassifier on three datasets such as PowerCons, Strawberry and BeetleFly datasets. Performance of these models shown below

Model	BasicMotions	ItalyPowerDemand	AtrialFibrillation	Average test score.
FCN	0.25	0.51	0.4	0.39
MLP	0.67	0.97	0.3	0.66
CNN Classifier	1.0	0.94	0.4	0.79

From this table CNNClassifier is the first rank with 0.79 then MLP then FCN.

Model Architecture Verification:

MLP: These model designed according to research paper information. It is constructed using tensorflow keras API where used a flatten layer and four dense layer. Dropout layer also used to reduces the dimensionality. Experimented with different activation functions to enhance model performance.

FCN: These model designed according to research paper information. It is constructed using tensorflow keras API where used there cov1d layer and a dense layer. Also used BatchNormalization in every layer to improve the model performance. GlobalAveragePooling used before final layer.

Task 3

For the time series data I chose LSTM neural networks and CNN network separately then concatenate both. Also chose two sktime transformers that's ExponentTransformer and Differencer. To improve model's performance I use early stopping and LearningRateScheduler callbacks.

For evaluation I tested the model on three datasets: BasicMotions, ItalyPowerDemand, and AtrialFibrillation. On the BasicMotions dataset, the model achieved an accuracy of 100%. On ItalyPowerDemand dataset, it achieved classification accuracy of 96%. on the AtrialFibrillation dataset, it achieved an accuracy of 26%.

Task 4

Choice of classifier and transformer pair: I choose to use KNeighbors Classifier paired with exponent Transformer and Differencer as transformer. KNeighborsClassifier is a simple effective algorithm for classification tasks. With 'Exponent transformer' apply element wise exponentiation transformation to time series. Then difference use to difference find the difference element wise.

Parameter to Optimize: I focused on optimizing the hyperparametes of the KNeighbors Classifier, including the number of neighbors (n_neighors). Also I optimize the data using exponent transformer then difference transformer. And difference apply iterative differences to a timeseries.

Best Score and Estimator: The best score and estimator from the randomized search instance were as follows:

- Best Score: 0.325
- Best Estimator : KNeighborsClassifier and exponentTransformer with power=0.7 and Differencer with lags=2.

Test Score of the best model: After training the best estimator on the entire training dataset, I evaluated it's performance on the test dataset. The test score of the best model was 0.325

Comparison with default settings: The default setting of the Kneighbors classifier typically yield a lower performance compared to the optimize model. In my case, the default settings achieved a test score of 0.25.

Task 5

In this report, I compare the performance of two chosen classifiers, namely RestNet, Rocket against a deep learning model on a multivariate dataset. The objective is to identify the best performance model based on the test score. Below are the findings along with explanations of how the multivariate dataset was handled for each model

Encoding categorical variable: Applied label encoding to convert categorical variables into numerical representations.

Model Training: Trained RestNet and rocket classifiers on the preprocessed dataset using the sktime library.

Model Training: Trained the deep learning model on the preprocessed Epilepsy dataset using appropriate training algorithms and loss functions.

Best Model:

Performance table

Model	**Test Score**
Rocket	0.97
ResNet	0.28
Deep learning model	0.95

From the above table, Rocket is the best model with the test accuracy is 0.97.