**Summary of Feature Extraction & Model Training Approaches**

**Machine Learning Approach with Feature Extraction (SVM)**

This approach involves **feature extraction** from inertial sensor data using **TSFEL** and then training an **SVM (Support Vector Machine)** classifier. The goal is to compare **TSFEL-generated features** with **predefined features from X\_train.txt**.

**Steps**

1. **Load Raw Inertial Signals**
   * Reads accelerometer (body\_acc\_x, body\_acc\_y, body\_acc\_z) and gyroscope (body\_gyro\_x, body\_gyro\_y, body\_gyro\_z) data.
   * Stores them as a **3D NumPy array**: (samples, time\_steps, channels).
2. **Feature Extraction using TSFEL**
   * Uses **TSFEL (Time Series Feature Extraction Library)** to generate **statistical, temporal, and frequency-domain features**.
   * Each extracted feature set is **flattened** into a 1D array: (samples, features).
3. **Train an SVM Model on TSFEL Features**
   * **Normalizes** extracted features using StandardScaler().
   * Trains an **SVM classifier with RBF kernel** on the TSFEL-generated features.
4. **Train Another SVM Model on Predefined Features**
   * Loads the **predefined features** (X\_train.txt, X\_test.txt).
   * **Normalizes and trains** a separate SVM classifier on this feature set.
5. **Compare Model Performance**
   * Compares the test accuracy of:
     + **SVM on TSFEL features**
     + **SVM on Predefined Features**

**Deep Learning Approach (1D CNN)**

This approach directly trains a **1D Convolutional Neural Network (CNN)** on **raw signals** and also on **predefined features** for comparison.

**Steps**

1. **Load Raw Inertial Signals**
   * Reads sensor data and structures it into a **3D format**: (samples, time\_steps, channels), which is necessary for CNN input.
2. **Train a 1D CNN on Raw Signals**
   * The CNN consists of:
     + **Convolutional Layers (Conv1D)** to extract spatial-temporal patterns.
     + **Batch Normalization** for stable learning.
     + **MaxPooling Layers** to reduce dimensionality.
     + **Dense Layers** for classification.
   * Trains on raw signals **without manual feature extraction**.
3. **Train an MLP (Fully Connected Model) on Predefined Features**
   * Instead of a CNN, a **Multi-Layer Perceptron (MLP)** is trained on the predefined feature set.
4. **Compare Model Performance**
   * Compares the accuracy of:
     + **CNN on Raw Signals**
     + **MLP on Predefined Features**

**Final Takeaways**

* **SVM works well when meaningful features are extracted** (TSFEL or predefined).
* **CNN performs better when trained on raw signals** because it can **automatically extract spatial-temporal features**.