DS TS Project

**1. Artificial Data Generation**

Objective:

Create a comprehensive dataset of artificial time series data to train and test your model.

Steps:

* *Generate Time Series*: Use statistical models such as AR (AutoRegressive), MA (Moving Average), and ARMA (AutoRegressive Moving Average) to generate synthetic data. Incorporate variations by adding trends and seasonal components to some of the series.
* *Labeling*: Clearly label each series with its respective characteristics (e.g., type of model, presence of trend, seasonality).
* *Data Split*: Split the data into training, validation, and test datasets to ensure robust model evaluation. Typically, a split of 70% training, 15% validation, and 15% test can be used.
* *Storage*: Efficiently store data in a structured format, possibly using databases or data frames in Python, to facilitate easy access and manipulation during model training.

**2. Model Building and Fine Tuning**

Objective:

Develop a predictive model capable of analyzing time series data.

Steps:

* *Model Selection*: Choose between various neural network architectures suitable for time series classification, such as Convolutional Neural Networks (CNNs), Vision Transformers (ViT), or even leveraging Large Language Models (LLMs) adapted for numerical data.
* *Feature Engineering*: Develop or automatically extract features that effectively capture the underlying patterns in the time series data.
* *Training and Validation*: Train your model using the training dataset and fine-tune hyperparameters using the validation set to avoid overfitting and maximize performance.
* *Evaluation Metrics*: Determine appropriate metrics (e.g., accuracy, F1 score, RMSE) to assess model performance during the tuning stage.

**3. Testing on Real Data**

Objective:

Validate the predictive accuracy and generalizability of the model on unseen, real-world data.

Steps:

* *Data Acquisition*: Collect real-world time series data from relevant sources that align with the type of data your model is intended to analyze.
* *Preprocessing*: Ensure this data is preprocessed similarly to your training data to maintain consistency in input to the model.
* *Comparison with Traditional Models*: Compare the model’s predictions with outcomes obtained through traditional time series analysis methods to validate its effectiveness.

**4. Dashboard Creation**

Objective:

Develop a user-friendly interface where users can upload time series data and receive predictions.

Steps:

* *Dashboard Design*: Design a simple and intuitive web-based dashboard. Tools like Dash or Streamlit can be very effective for this purpose.
* *Functionality Implementation*: Implement the functionality to upload time series data, preprocess it according to the model’s requirements, and display predictions.
* *User Testing and Feedback*: Before full deployment, conduct user testing to gather feedback and make necessary adjustments to improve usability and functionality.
* *Deployment*: Deploy the dashboard on a web server or a cloud platform to make it accessible to users.

**Additional Considerations**

Documentation:

Ensure to document each part of your project thoroughly for future reference and for users who may need to understand or modify the dashboard.

Scalability and Maintenance:

Plan for potential scalability needs and regular maintenance of the system to handle increased user load or data volume.