

**SUBMITTED BY: GUIDED BY:**

### 1.Hemant Lodha 211112417 Dr. Pragati Agrawal

### 2.Anand Patel 211112424

### 3.Apaar Kumar 211112003

**4.Hemant Kumar Chaurasiya 211112407**



**DECLARATION**

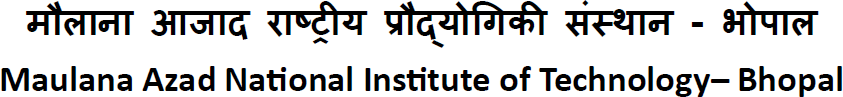
We hereby declare that the work, which is presented in this Project Report, entitled “**Soil Moisture Dataset Analysis and Forecast”**, in partial fulfilment of the requirements for the award of the degree, submitted in the **Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal.** It is an authentic record of my work carried out from 25/01/2024 to 10/03/2024 under the noble guidance of my guide “**Dr.** **Pragati Agrawal”.** The following project and its report, in part or whole, have not been presented or submitted by me for any purpose in any other institute or organization. We hereby declare that the facts mentioned above are true to the best of our knowledge. In case of any unlikely discrepancy that may possibly occur, We will be the one to take responsibility.

### Hemant Lodha 211112417

### Anand Patel 211112424

### Apaar Kumar 211112003

### Hemant Kumar Chaurasiya 211112407

******

******

**Department of Computer Science and Engineering**

**कंप्यूटर**

**विज्ञान और अभियांत्रिकी विभाग**

**CERTIFICATE**

This is to certify that " **Hemant Lodha** ", "**Anand Patel**", "**Apaar Kumar**" and "

**Hemant Kumar Chaurasiya**", student of B.Tech 3rd Year (Computer Science &

Engineering), have successfully completed their project **“Soil Moisture Dataset**

**Analysis and Forecast’’** in partial fulfilment of their Bachelor of Technology in

Computer Science & Engineering.

Dr. Pragati Agrawal

#### 

# ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide and coordinator Dr. Pragati Agrawal for his valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully.

It is imperative for us to mention the fact that the report of minor project could not have been accomplished without the periodic suggestions and advice of our project guide Dr. Pragati Agrawal.

We are also grateful to our respected HOD, Dr. Deepak Singh Tomar and HOD, AI, Dr. Nilay Khare and Chairperson, Central Computing Facility, Dr. Meenu Chawla for permitting us to utilize all the necessary facilities of the college.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help. Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing support and encouragement

**ABSTRACT**

यह अध्ययन मिट्टी की नमी के लिए एक विश्लेषण और पूर्वानुमान रूपरेखा प्रस्तुत करता है डेटासेट, कृषि योजना, जल संसाधन प्रबंधन और के लिए महत्वपूर्ण जलवायु अध्ययन. मिट्टी की नमी, पृथ्वी की सतह के जल विज्ञान का एक प्रमुख घटक, जैसे कारकों से प्रभावित महत्वपूर्ण स्थानिक और लौकिक परिवर्तनशीलता प्रदर्शित करता है वर्षा, तापमान, वनस्पति आवरण और मिट्टी के गुण।

इस्तेमाल मशीन लर्निंग और सांख्यिकीय सहित उन्नत डेटा विश्लेषण तकनीकें विधियों, हम अस्थायी रुझानों को स्पष्ट करने के लिए ऐतिहासिक मिट्टी की नमी के आंकड़ों का विश्लेषण करते हैं, स्थानिक पैटर्न, और पर्यावरणीय चर के साथ उनके संबंध। प्रस्तावित ढांचे में डेटा प्रीप्रोसेसिंग, फीचर इंजीनियरिंग, शामिल है मॉडल चयन, प्रशिक्षण और सत्यापन चरण। हम विभिन्न प्रकार की मशीनें लगाते हैं लर्निंग एल्गोरिदम, अनुभवजन्य मोड अपघटन (ईएमडी), स्थानीय माध्यअपघटन (एलएमडी), दीर्घकालिक अल्पकालिक मेमोरी (एलएसटीएम) और ऑटोरेग्रेसिव जटिल गैर-रेखीय कैप्चर करने के लिए इंटीग्रेटेड मूविंग एवरेज (अरिमा)।रिश्ते और सटीक भविष्यवाणियाँ करें।इसके अतिरिक्त, हम समय-श्रृंखला के लिए पारंपरिक सांख्यिकीय तरीकों के उपयोग का पता लगाते हैं मिट्टी की नमी की गतिशीलता में मौसमी और दीर्घकालिक रुझान निकालने के लिए विश्लेषण।

This study presents an analysis and forecasting framework for soil moisture datasets, crucial for agricultural planning, water resource management, and climate studies. Soil moisture, a key component of the Earth's surface hydrology, exhibits significant spatial and temporal variability influenced by factors such as precipitation, temperature, vegetation cover, and soil properties. Leveraging advanced data analytics techniques, including machine learning and statistical methods, we analyze historical soil moisture data to elucidate temporal trends, spatial patterns, and their relationships with environmental variables.

The proposed framework encompasses data preprocessing, feature engineering, model selection, training, and validation stages. We employ a variety of machine learning algorithms, Empirical Mode Decomposition (EMD) , Local Mean Decomposition (LMD), Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) to capture complex non-linear relationships and make accurate predictions.

Additionally, we explore the use of traditional statistical methods for time-series analysis to extract seasonal and long-term trends in soil moisture dynamics.

**Contents**

1. CERTIFICATE 2
2. DECLARATION 3
3. ACKNOWLEDGEMENT 4
4. ABSTRACT 5
5. [List of Figures 8](#_Toc126079864)

[Chapter 1 Introduction 9-](#_Toc126079867)11

[1.1 Project Overview 9](#_Toc126079868)

[1.2 Problem Statement 10](#_Toc126079871)

[Chapter 2 Methodology 12](#_Toc126079874)-13

[2.1 Preprocessing 12](#_Toc126079875)

[2.2 Decomposition 12](#_Toc126079876)

[2.3 Model Selection 12](#_Toc126079877)

[2.4 Training and Testing 12](#_Toc126079878)

[2.5 Prediction and Evaluation 12](#_Toc126079878)

[2.6 Plan 13](#_Toc126079884)

[2.6.1 Data Preprocessing 13](#_Toc126079885)

[2.6.2 Decomposition 13](#_Toc126079886)

[2.6.3 Model Selection 13](#_Toc126079887)

[2.6.4 Training and Testing 13](#_Toc126079887)

[2.6.5 Prediction and Evaluation 13](#_Toc126079887)

[2.6.6 Analysis and Interpretation 13](file:///C:\Users\hp%20world\Downloads\Project_Report_Guidelines_Final%5b1%5d%5b1%5d%20(1)%20(1).docx#_Toc126079887)

[Chapter 3 Machine Learning Algorithm 14](#_Toc126079891)-16

[3.1 Empirical Mode Decomposition (EMD) 14](#_Toc126079892)

[3.1.1 Introduction 14](#_Toc126079895)

[3.1.2 Working 14](#_Toc126079895)

[3.1.3 Graph of IMFs 15](#_Toc126079895)

[3.1.4 Differentiate Between Linear and Non-linear IMFs 15](#_Toc126079895)

[3.2 Local Mean Decomposition (LMD) 18](#_Toc126079893)

[3.2.1 Introduction 18](file:///C:\Users\hp%20world\Downloads\Project_Report_Guidelines_Final%5b1%5d%5b1%5d%20(1)%20(1).docx#_Toc126079895)

[3.2.2 Working 19](file:///C:\Users\hp%20world\Downloads\Project_Report_Guidelines_Final%5b1%5d%5b1%5d%20(1)%20(1).docx#_Toc126079895)

[3.3 Long Short-Term Memory (LSTM) 20](#_Toc126079894)

[3.3.1 Introduction 20](#_Toc126079895)

[3.3.2 Basic Architecture 20](#_Toc126079896)

[3.3.3 Memory Cell 21](#_Toc126079897)

[3.3.4 Training 21](#_Toc126079898)

[3.3.5 Advantages of LSTMs 21](#_Toc126079899)

[3.3.6 Applications 21](#_Toc126079900)

[3.6 AutoRegressive Integrated Moving Average (ARIMA) 22](#_Toc126079904)

[3.6.1 Introduction 22](#_Toc126079905)

[3.6.2 AutoRegressive (AR) Component 22](#_Toc126079906)

[3.6.3 Integrated (I) Component 22](#_Toc126079907)

[3.6.4 Moving Average (MA) Component 23](#_Toc126079908)

[3.6.5 Model Identification and Parameter Estimation 23](#_Toc126079909)

[3.6.6 Model Fitting 23](#_Toc126079909)

[3.6.7 Diagnostic Checking 23](#_Toc126079909)

[3.6.8 Forecasting 24](#_Toc126079909)

[Chapter 4 Results 25](#_Toc126079910)

[4.1. ARIMA Model (Linear IMFs) 25](#_Toc126079911)

[4.2. LSTM Model (Nonlinear IMFs) 26](#_Toc126079912)

[Chapter 5 Conclusion 27](#_Toc126079926)

[Chapter 6 Future Scope 28](#_Toc126079926)

[Chapter 7 Reference 29](#_Toc126079926)

# V List of Figures

[Figure 1.1 Flowchart of project 11](#_heading=h.4d34og8)

[Figure 3.1 Graphs of IMFs 15](#_heading=h.2s8eyo1)

[Figure 3.2 Graph of Linear IMFs 17](#_heading=h.2xcytpi)

[Figure 3.3 Graph of Non-linear IMFs 17](#_heading=h.1ci93xb)

[Figure 3.4 Architecture for an LSTM 20](#_heading=h.3whwml4)

[Figure 3.5 Introduction to ARIMA 22](#_heading=h.4d34og8)

**Title: Soil Moisture Dataset Analysis and Forecast using Machine Learning**

**Chapter 1 Introduction**

* 1. **Project Overview**

In recent years, the study and analysis of soil moisture dynamics have garnered significant attention due to their crucial role in agriculture, hydrology, and environmental science. Soil moisture content is a critical parameter that influences various processes such as plant growth, water infiltration, soil erosion, runoff generation, and groundwater recharge. As such, it has a direct impact on the efficiency of agricultural practices, the sustainability of water resources, and the overall health of ecosystems.

Understanding the temporal variation of soil moisture is essential for various applications, ranging from improving water resource management to enhancing agricultural productivity and monitoring environmental changes. In the agricultural sector, soil moisture levels can affect plant growth, crop yields, and irrigation needs. Accurate prediction of soil moisture content can lead to optimized irrigation schedules, reducing water usage and enhancing crop health. In hydrology, soil moisture data can aid in predicting flood risks and managing drought conditions. Additionally, soil moisture is a key component in climate studies, as it plays a role in the Earth's water cycle and energy balance.

Given these wide-ranging applications, this study aims to analyze a dataset containing the average soil moisture volume of various soil samples collected over time. The primary objective is to understand the underlying patterns and trends in the data and to predict future soil moisture levels. This analysis involves several key steps, beginning with the preprocessing of the dataset to ensure data quality and consistency. This process includes handling missing data, outliers, and noise, as well as standardizing the data for further analysis.

Following the preprocessing stage, we will employ advanced signal processing techniques such as Empirical Mode Decomposition (EMD) and Local Mean Decomposition (LMD). These methods allow us to decompose the soil moisture time series into intrinsic mode functions or product functions, which reveal the inherent oscillatory behavior and trends within the data. By breaking down the time series into these components, we can better understand the complexity and periodicity of soil moisture variations.

A crucial aspect of this study is to assess the linearity of the resulting product functions (PFs) derived from the decomposition process. This step is essential for selecting the appropriate forecasting models for predicting future soil moisture levels. If the time series exhibits nonlinear behavior, nonlinear forecasting models will be more suitable; otherwise, linear models may suffice.

Once the decomposition and linearity assessment are complete, the next phase involves selecting and applying appropriate time series forecasting models. These models, which may include autoregressive integrated moving average (ARIMA), long short-term memory (LSTM), or other machine learning-based approaches, will be used to generate predictions of future soil moisture content.

The outcomes of this study have the potential to provide valuable insights into soil moisture dynamics and contribute to better water resource management, crop yield prediction, and drought monitoring. The findings may also offer practical recommendations for agricultural practices, flood management, and climate adaptation strategies. Overall, this study represents a comprehensive approach to understanding and forecasting soil moisture levels, with significant implications for agriculture, hydrology, and environmental science.

* 1. **Problem Statement**

The dataset at hand presents a time series of average soil moisture volume measurements collected from different soil samples from different districts in Madhya Pradesh. Using suitable Machine Learning algorithms we need to train a model that can predict the average soil moisture volume measurement at a given point in the future (forecast).

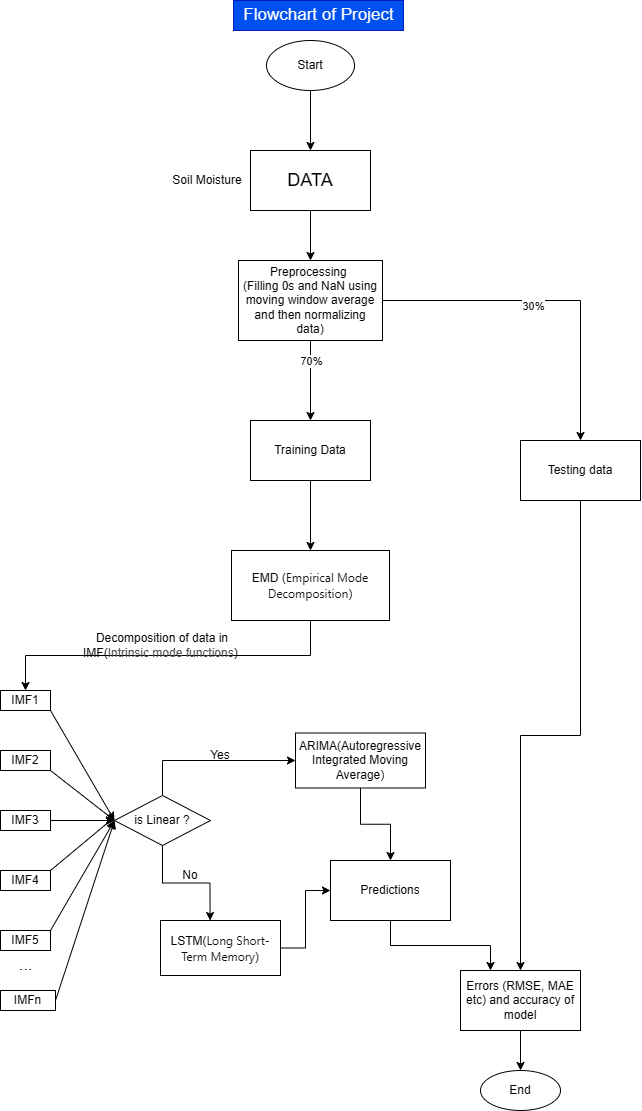


Figure 1.1 Flowchart of project

**Chapter 2 Methodology**

**2.1 Preprocessing:**

We will start by preprocessing the dataset to ensure its suitability for analysis. This involves applying a moving average to smooth out noise and normalizing the data to bring all features to a common scale and also fill any empty cells. The moving average helps in reducing short-term fluctuations, making the underlying trends more visible. Normalization is essential to prevent features with larger scales from dominating the model training process.

**2.2 Decomposition:**

Once the data is preprocessed, we will decompose the time series using Empirical Mode Decomposition (EMD) and Local Mean Decomposition (LMD). EMD decomposes the time series into Intrinsic Mode Functions (IMFs), while LMD further decomposes these IMFs into Product Functions (PFs). This decomposition process helps in isolating different temporal components present in the data, such as trend, seasonality, and noise.

**2.3 Model Selection:**

After decomposition, we will assess the linearity of the PFs to determine the appropriate forecasting models. Nonlinear PFs will be modeled using Long Short-Term Memory (LSTM) networks, a type of recurrent neural network known for capturing long-term dependencies in sequential data. Linear PFs will be modeled using Autoregressive Integrated Moving Average (ARIMA) models, a popular choice for time series forecasting.

**2.4 Training and Testing:**

We will split the preprocessed data into training and testing sets, with a ratio of 70:30. The training set will be used to train the LSTM and ARIMA models, while the testing set will be used to evaluate their performance. We will train the LSTM model on nonlinear PFs and the ARIMA model on linear Pfs.

After training the predicted values will be compared to the actual values and various errors shall be calculated.

**2.5 Prediction and Evaluation:**

Once the models are trained, we will use them to predict future soil moisture levels based on the testing data. We will then evaluate the predictions against the actual values using various error metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R2) to assess the accuracy and reliability of the forecasting models.

**2.6 Plan**

**2.6.1 Data Preprocessing:**

* + Apply moving average and normalization to the dataset.
    1. **Decomposition:**
  + Implement Empirical Mode Decomposition (EMD) and Local Mean Decomposition (LMD) to decompose the time series data into IMFs and PFs, respectively.
    1. **Model Selection:**
  + Assess the linearity of the PFs to determine the appropriate models (LSTM for nonlinear PFs, ARIMA for linear Pfs ).
    1. **Training and Testing:**
  + Split the preprocessed data into training and testing sets.
  + Train the LSTM and ARIMA models on the respective datasets.
    1. **Prediction and Evaluation:**
  + Use the trained models to predict future soil moisture levels.
  + Evaluate the predictions against the actual values using various error metrics.
    1. **Analysis and Interpretation:**
  + Analyze the results to understand the performance of the forecasting models.
  + Interpret the findings and draw insights regarding soil moisture dynamics and forecasting accuracy.

This comprehensive plan outlines the approach to address the challenges posed by the soil moisture dataset, from preprocessing and decomposition to model selection, training, and evaluation. By following this methodology, we aim to gain valuable insights into the temporal behavior of soil moisture and develop accurate forecasting models to aid in agricultural and environmental decision-making processes.

**Chapter 3 MACHINE LEARNING ALGORITHM**

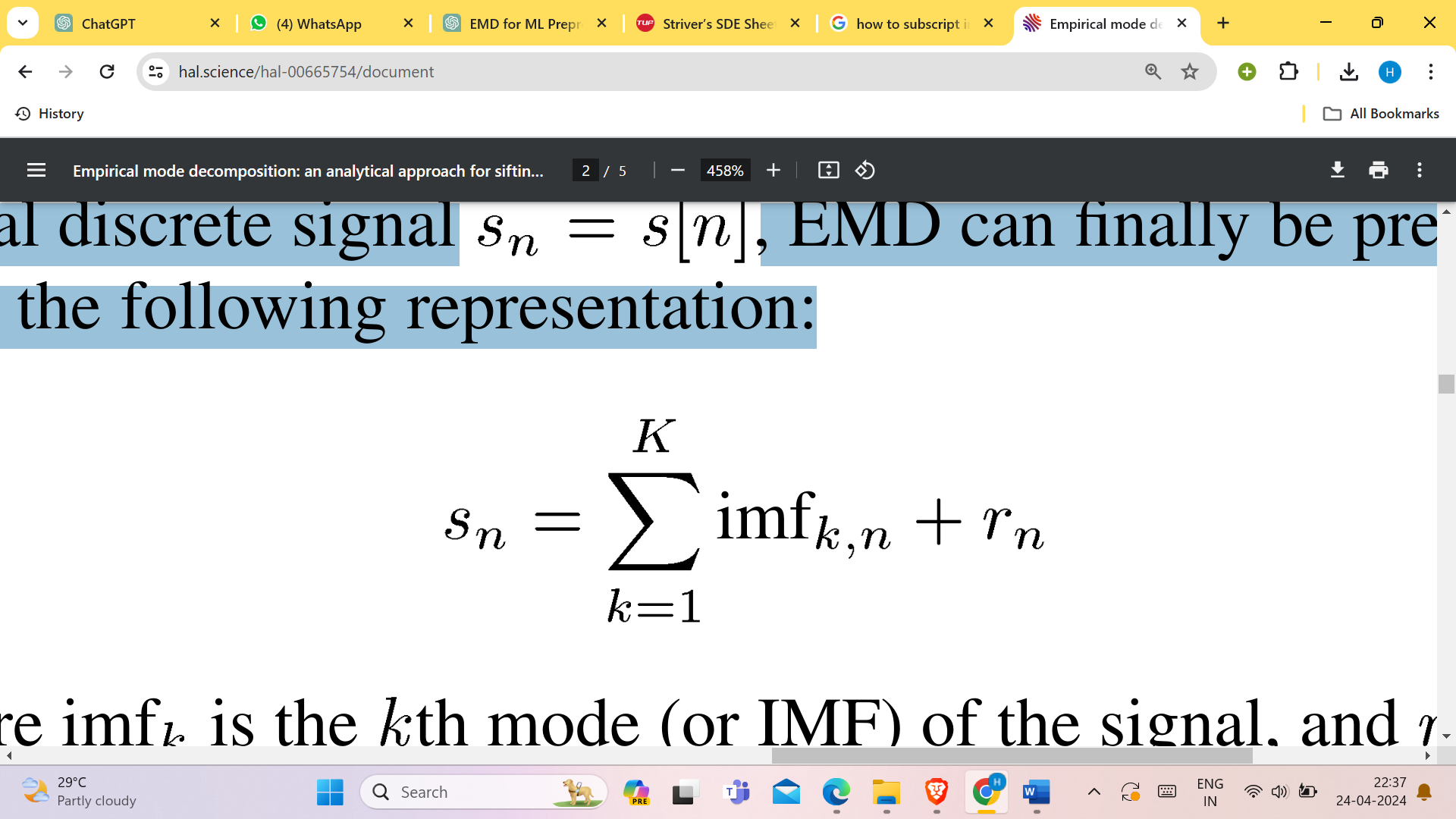
**3.1 Empirical Mode Decomposition (EMD) Algorithm in Machine Learning**

**3.1.1 Empirical Mode Decomposition (EMD)** is a signal processing technique rather than a traditional machine learning algorithm. It's used to decompose a signal into a finite set of intrinsic mode functions (IMFs) based on the local characteristics of the signal. Each IMF represents a different oscillatory mode present in the signal.

EMD is often used as a preprocessing step for various signal processing tasks, such as denoising, feature extraction, and time series analysis. While it's not a machine learning algorithm itself, it's often combined with machine learning techniques to improve the analysis of time series data.

Once you decompose a signal using EMD into its IMFs, you can apply machine learning algorithms to each IMF or the combination of IMFs to perform tasks like classification, prediction, or anomaly detection on time series data.

So, for any one-dimensional discrete signal Sn=S[n], EMD can finally be presented with the following representation:

(equ. 1)

where imfk is the Kth mode (or IMF) of the signal, and r is the residual trend (a low-order polynomial component).

**3.1.2 Working**

Empirical Mode Decomposition (EMD) works by decomposing a given time series signal into a set of intrinsic mode functions (IMFs) and a residual. Here's a simplified overview of the EMD process:

1. Identify Extrema: First, identify all local maxima and minima in the time series signal.
2. Generate Upper and Lower Envelopes: Connect the local maxima to form the upper envelope and connect the local minima to form the lower envelope. These envelopes represent the upper and lower bounds of the signal.
3. Calculate Mean: Compute the mean of the upper and lower envelopes to obtain the mean envelope.
4. Extract Component: Subtract the mean envelope from the original signal to obtain the first component, which is an IMF.
5. Check for IMF Criteria: Verify if the extracted component meets the criteria of being an IMF. An IMF should have the same number of zero crossings and extrema, and the mean of its upper and lower envelopes should be zero or close to zero. If it doesn't meet these criteria, repeat steps 2-4 until it does.
6. Repeat: After obtaining the first IMF, repeat steps 2-5 on the residual (original signal minus the first IMF) to extract the next IMF. Continue this process iteratively until the residual becomes a monotonic function or meets certain convergence criteria.
7. Obtain Residual: The final residual obtained after extracting all IMFs represents the trend of the original signal.

**3.1.3 Graphs of IMFs**

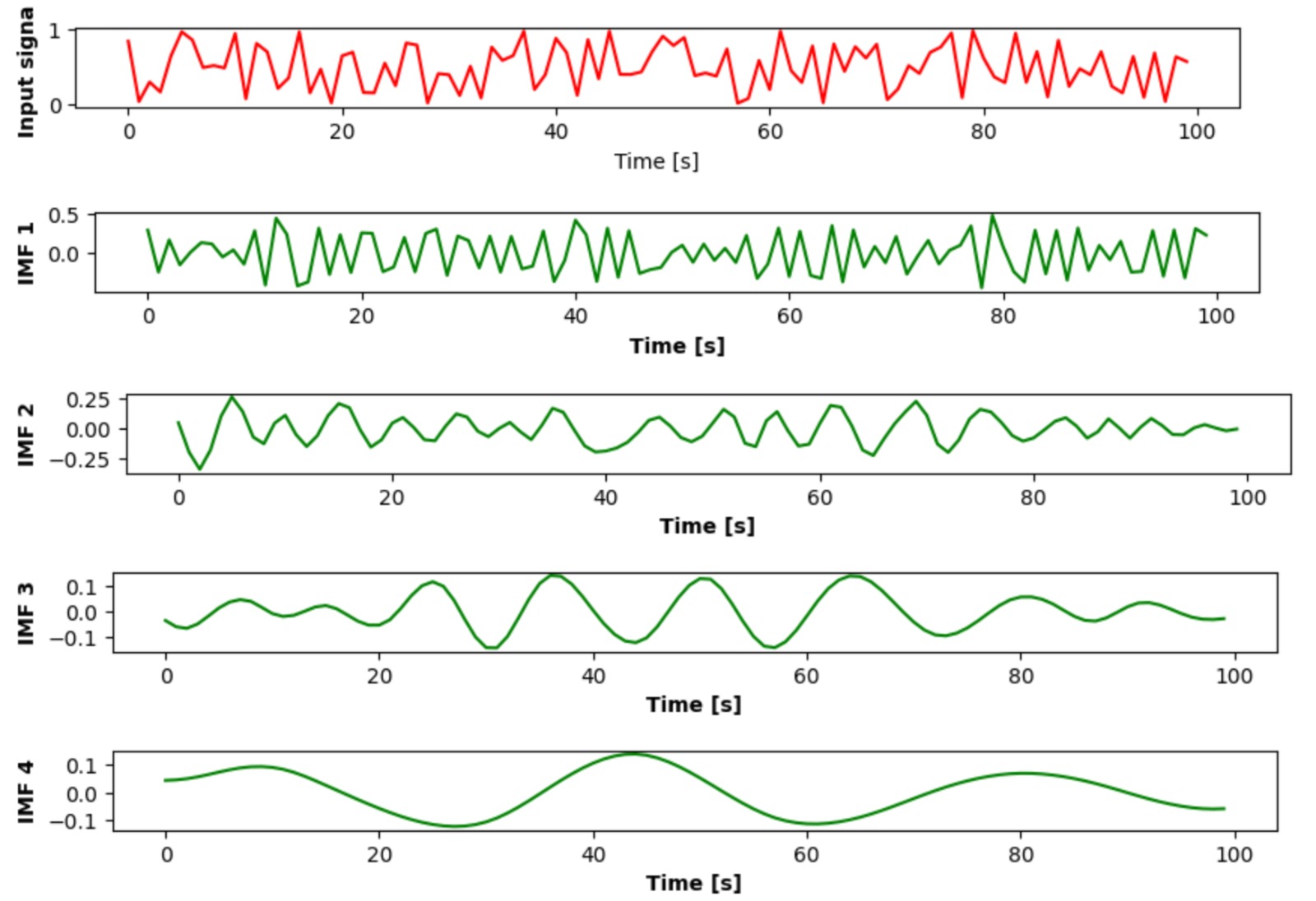


Figure 3.1 Graphs of IMFs

**3.1.4 Differentiate Between Linear and Non-linear IMFs**

To create an algorithm to differentiate between linear and non-linear intrinsic mode functions (IMFs) generated by Empirical Mode Decomposition (EMD), you can incorporate several analytical techniques to evaluate the properties of IMFs. The following algorithm outlines a multi-step approach, which combines frequency analysis, Hilbert spectrum analysis, and statistical methods to categorize IMFs as linear or non-linear.

**Algorithm to Differentiate Between Linear and Non-linear IMFs**

The Hilbert transform and the kurtosis metric can be used to evaluate intrinsic mode functions (IMFs) from Empirical Mode Decomposition (EMD) and to differentiate between linear and non-linear IMFs. Here's how you might apply these techniques to assess the linearity of IMFs.

**Step 1: Hilbert Transform**

The Hilbert transform is used to obtain the instantaneous amplitude and frequency of an IMF, which provides information on the underlying oscillations.

1. **Apply Hilbert Transform**:
   * For each IMF, apply the Hilbert transform to extract the instantaneous amplitude and phase.
   * The instantaneous phase 𝜃(𝑡) is used to calculate the instantaneous frequency, given by the derivative:

𝜔(𝑡)=𝑑𝜃(𝑡)𝑑𝑡

1. **Examine Instantaneous Frequency**:
   * Linear IMFs generally exhibit a consistent instantaneous frequency over time.
   * Non-linear IMFs may show more complex frequency patterns, varying frequencies, or oscillations.
2. **Check for Frequency Modulations**:
   * Analyze the time variation of the instantaneous frequency. Linear IMFs should have relatively stable frequencies, while non-linear IMFs might show frequency modulation.

**Step 2: Kurtosis Analysis**

Kurtosis is a statistical measure of the "tailedness" or peakedness of a distribution. It can indicate non-linearity if the distribution of oscillations in an IMF is skewed or contains outliers.

1. **Calculate Kurtosis**:
   * Compute the kurtosis of the IMF's amplitude distribution.
   * A high kurtosis indicates heavy tails, suggesting non-linear behavior, while a low kurtosis implies a more symmetrical distribution, indicating linearity.
2. **Analyze Amplitude Distribution**:
   * Plot the amplitude distribution and check for outliers or deviations from a normal distribution.
   * Linear IMFs typically have more symmetric distributions with fewer outliers.
   * Non-linear IMFs may have skewed distributions with high kurtosis, indicating non-linear characteristics.

**Step 3: Decision Rule**

Based on the results from the Hilbert transform and kurtosis analysis, you can create a decision rule to classify IMFs as linear or non-linear.

* **Classify as Linear**:
  + If the instantaneous frequency is relatively stable with low variation, and the kurtosis is low (indicating a symmetrical distribution), the IMF is likely linear.
* **Classify as Non-linear**:
  + If the instantaneous frequency shows significant variation or modulation, and the kurtosis is high (indicating heavy tails or outliers), the IMF is likely non-linear.

**Conclusion**

By applying the Hilbert transform to analyze the instantaneous frequency and using kurtosis to evaluate the amplitude distribution, you can gain insights into the linearity or non-linearity of IMFs. This approach can help differentiate between linear and non-linear IMFs in a robust and statistically sound manner.

**Linear IMFs**

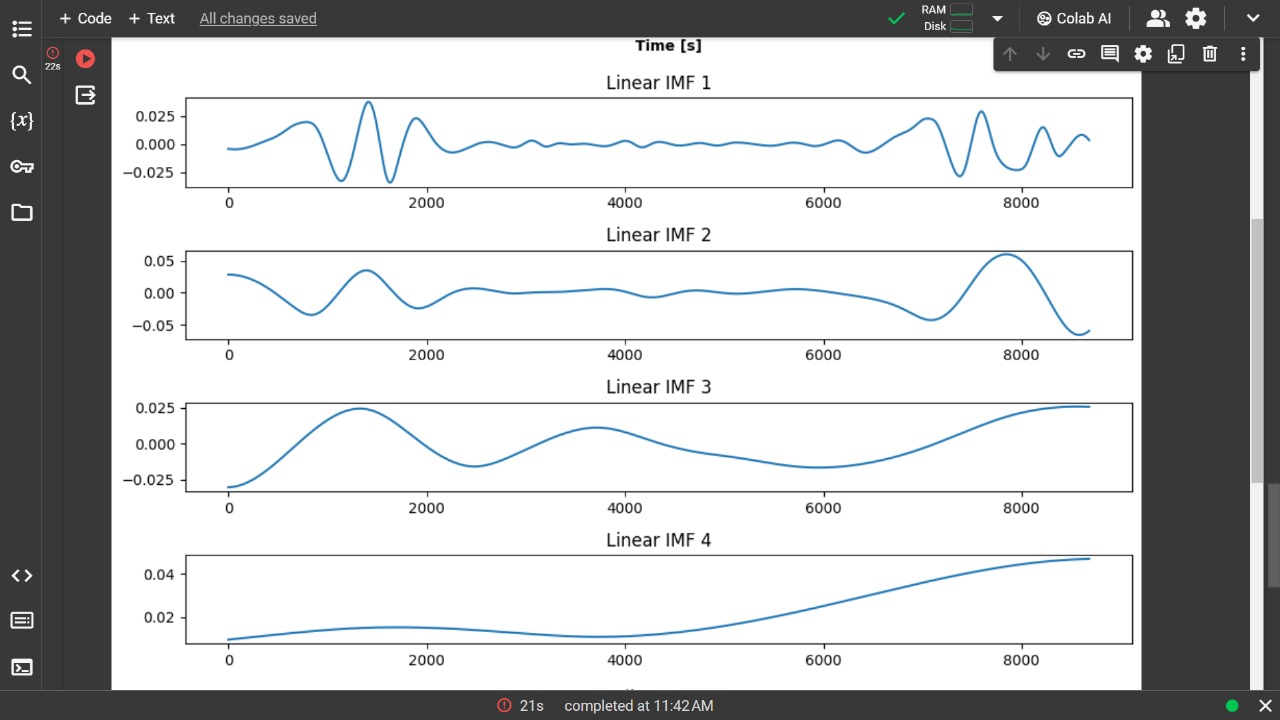


Figure 3.2 Graph of Linear IMFs

**Non-linear IMFs**

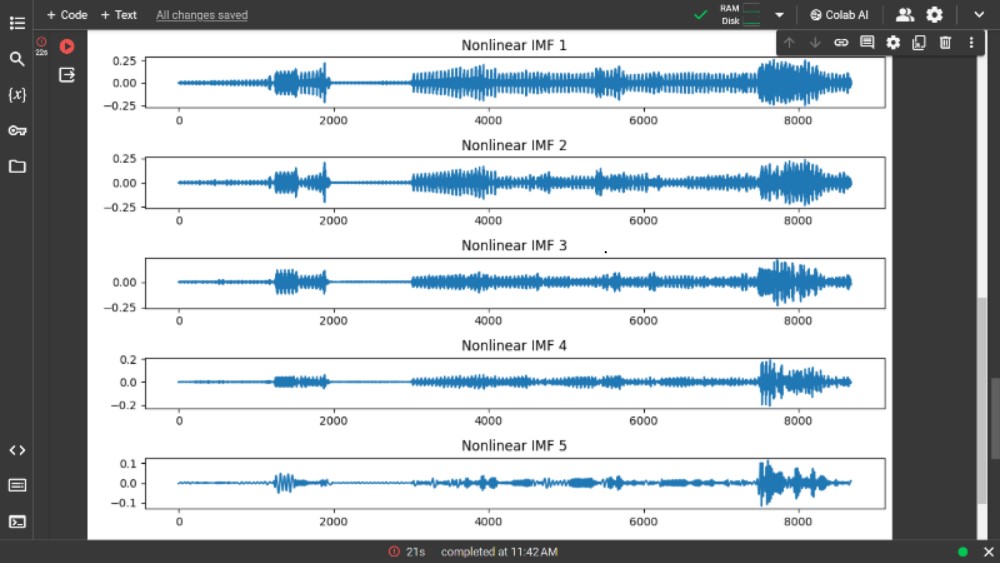


Figure 3.3 Graph of Non-linear IMFs

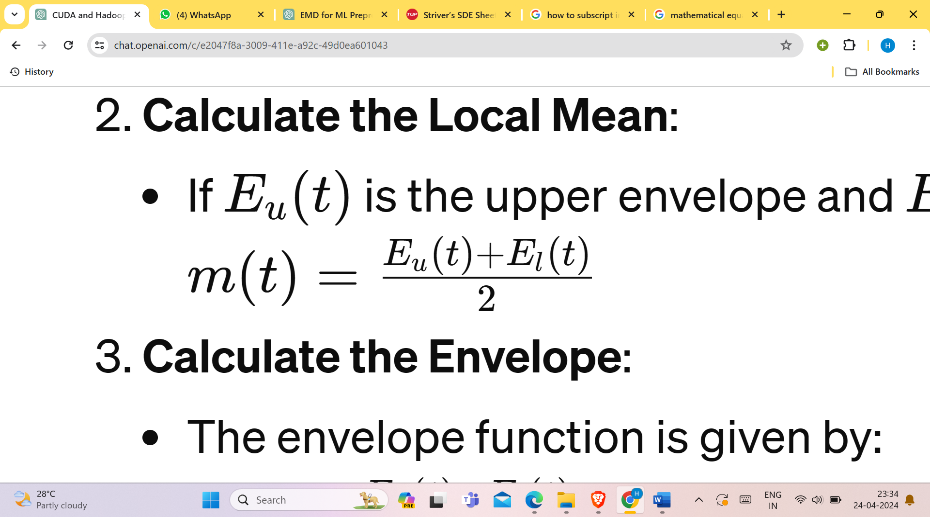
**3.2 Local Mean Decomposition**

**3.2.1 Introduction**

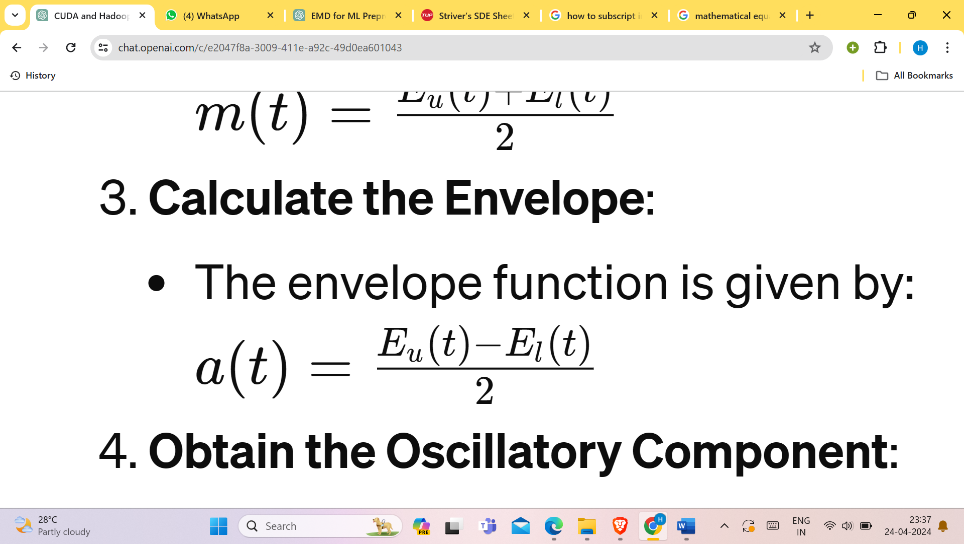
Local Mean Decomposition (LMD) is a data-driven technique used for time-frequency analysis and decomposition of non-stationary signals, particularly in the context of time series data. It aims to decompose a given signal into a set of oscillatory components (called intrinsic mode functions or IMFs) and a slowly varying trend component.

Here's a mathematical representation of LMD:

1. **Original Signal**:
   * Given a signal 𝑥(𝑡), the aim is to decompose it into Product Functions (PFs) and a residual.
2. **Calculate the Local Mean**:
   * If 𝐸𝑢(𝑡) is the upper envelope and 𝐸𝑙(𝑡) is the lower envelope, the local mean is:

 (equ. 2)

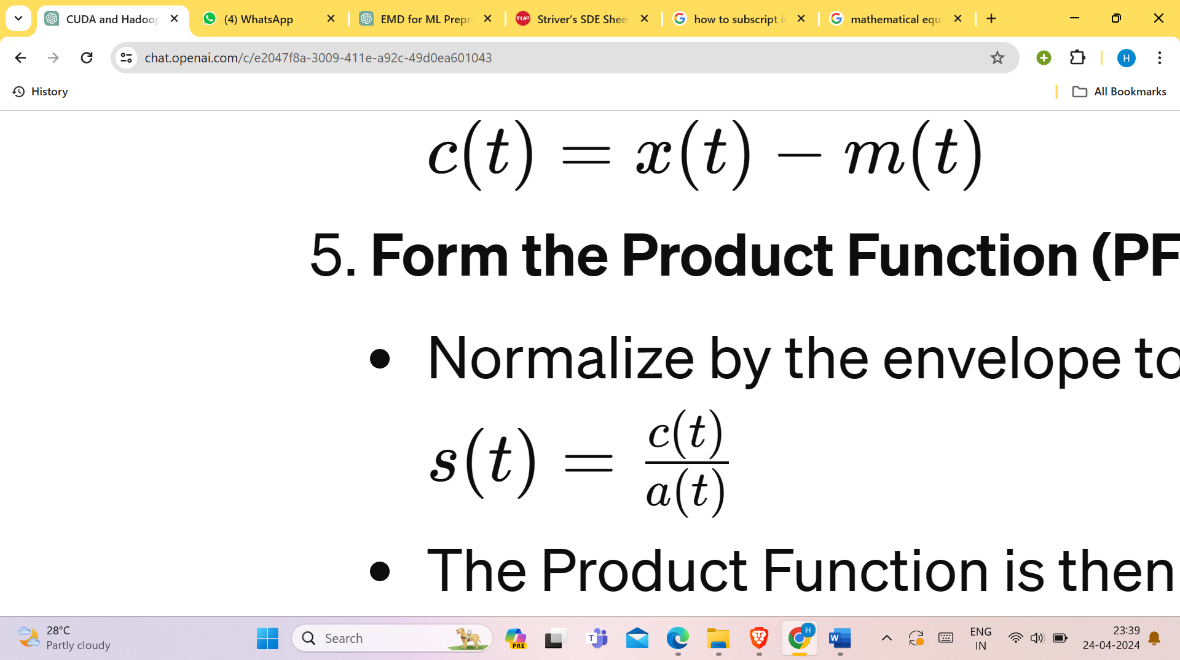
1. **Calculate the Envelope**:
   * The envelope function is given by:

 (equ. 3)

1. **Obtain the Oscillatory Component**:
   * Subtract the local mean from the original signal to get the oscillatory component:

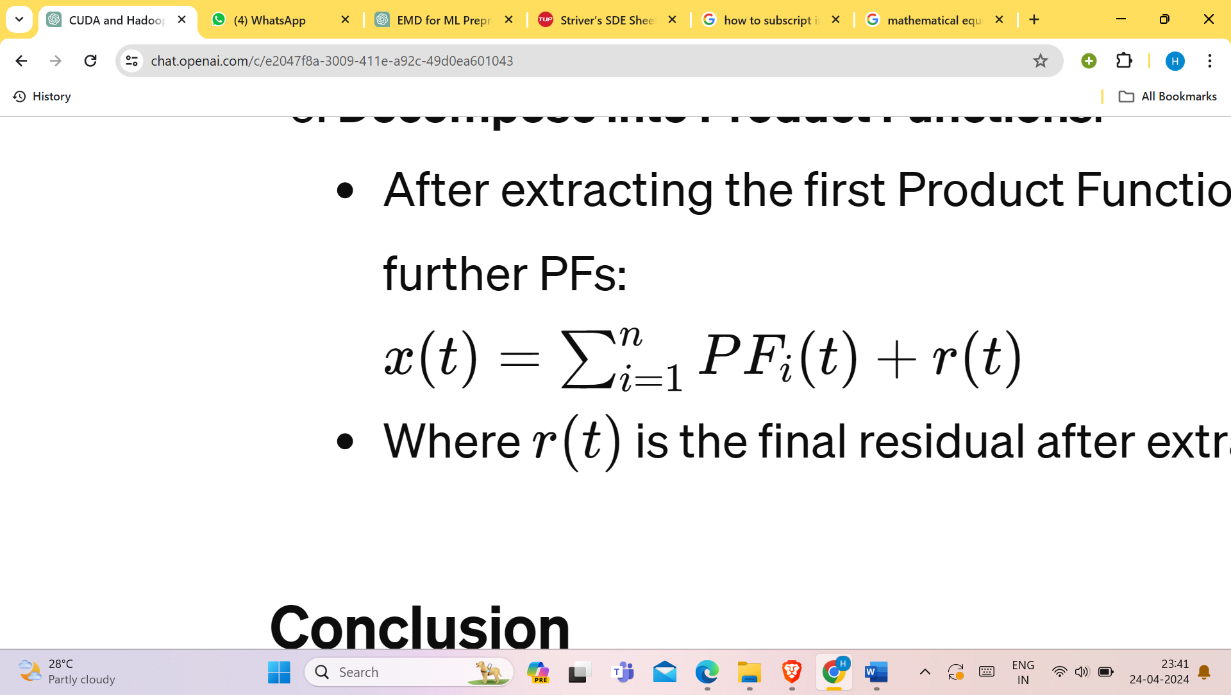
*c*(*t*)=*x*(*t*)−*m*(*t*) (equ. 4)

1. **Form the Product Function (PF)**:
   * Normalize by the envelope to create the frequency-modulated signal:

 (equ. 5)

* + The Product Function is then given by:

*PF*(*t*)=*a*(*t*)×*s*(*t*) (equ. 6)

1. **Decompose into Product Functions**:
   * After extracting the first Product Function, the process is repeated with the residual to obtain further PFs:   
       (equ. 7)
   * Where 𝑟(𝑡) is the final residual after extracting all significant PFs.

It several applications across various fields due to its ability to analyze non-stationary signals and decompose them into interpretable components. Some of the key applications of LMD include:

1. **Feature Extraction**: LMD can extract relevant features from complex time series data, which can be used for subsequent analysis or classification tasks. The IMFs obtained from LMD represent different frequency components of the signal, and features derived from these IMFs can capture important characteristics of the underlying process.
2. **Time-Frequency Analysis**: LMD provides a time-frequency representation of the signal by decomposing it into IMFs with varying frequencies. This allows for the analysis of how the frequency content of the signal changes over time, providing insights into transient phenomena and frequency modulation.

**3.2.2 Working**

Here's a more detailed explanation of how LMD works:

1. **Signal Decomposition**: LMD iteratively decomposes the original signal into a series of IMFs and a residual. Each IMF captures a different oscillatory mode present in the signal, while the residual represents the slowly varying trend or non-periodic components.
2. **Local Mean Estimation**: At each iteration, LMD estimates the local mean of the signal by applying a smoothing operation. This local mean represents the slowly varying trend component of the signal.
3. **IMF Extraction**: The difference between the original signal and the estimated local mean gives a high-frequency component, which is assumed to be an IMF. This process is repeated iteratively until the residual becomes sufficiently smooth or meets certain convergence criteria.
4. **Adaptivity**: LMD adapts to the local characteristics of the signal, making it suitable for analyzing non-stationary and nonlinear signals where the frequency content may vary over time.
5. **Applications**: LMD has applications in various domains, including signal processing, time series analysis, biomedical signal analysis, and vibration analysis. It can be used for tasks such as signal denoising, feature extraction, time-frequency analysis, and pattern recognition.

Top of Form

**3.3 Long Short-Term Memory**

**3.3.1 Introduction**

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to model sequential data and capture long-term dependencies. Here's a breakdown of LSTM and its key components:

**3.3.2 Basic Architecture**:

* + LSTM networks consist of memory cells, which are specialized units capable of learning and remembering information over long sequences.
  + Each memory cell has three gates:
    - **Forget Gate**: Controls what information to forget from the cell state.
    - **Input Gate**: Determines what new information to store in the cell state.
    - **Output Gate**: Regulates what information to output from the cell state.
  + The gates use sigmoid and tanh activation functions to control the flow of information within the cell.

9

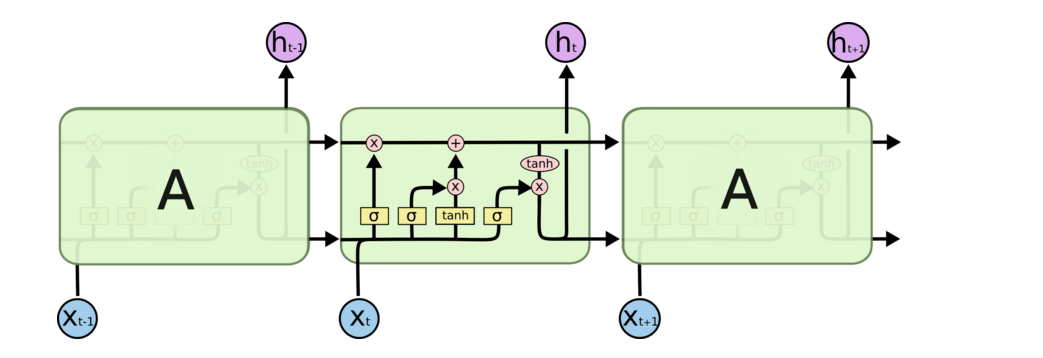
Architecture for an LSTM

Longterm-short term model

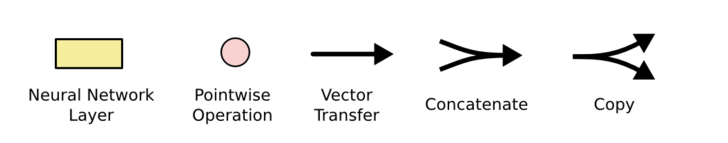
Decide what to forget

Decide what to insert

“Bits of memory”



Combine with transformed xt

****

σ: output in [0,1]

tanh: output in [-1,+1]

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Figure 3.4 Architecture for an LSTM

**3.3.3 Memory Cell**:

* + The cell state is the "memory" of the LSTM unit, allowing it to maintain information across time steps.
  + The forget gate decides which information from the previous cell state to discard:

𝑓𝑡=𝜎(𝑊𝑓⋅[ℎ𝑡−1,𝑥𝑡]+𝑏𝑓) (equ. 8)

* + The input gate determines new information to be added to the cell state:

𝑖𝑡=𝜎(𝑊𝑖⋅[ℎ𝑡−1,𝑥𝑡]+𝑏𝑖) (equ. 9)

𝐶~𝑡=tanh(𝑊𝐶⋅[ℎ𝑡−1,𝑥𝑡]+𝑏𝐶)(equ. 10)

* + The cell state is updated using the forget gate and the input gate:

𝐶𝑡=𝑓𝑡⋅𝐶𝑡−1+𝑖𝑡⋅𝐶~𝑡 (equ. 11)

* + The output gate determines what part of the cell state to output:

𝑜𝑡=𝜎(𝑊𝑜⋅[ℎ𝑡−1,𝑥𝑡]+𝑏𝑜) (equ. 12)

ℎ𝑡=𝑜𝑡⋅tanh(𝐶𝑡) (equ. 13)

* + 1. **Training**:
  + During training, LSTM networks use backpropagation through time (BPTT) to learn the optimal weights for the gates and the cell state transformations.
  + The network minimizes a loss function (e.g., mean squared error or categorical cross-entropy) using optimization algorithms like gradient descent or its variants.
    1. **Advantages of LSTMs**:

**3.3.5.1 Long-Term Dependencies**: LSTMs excel at capturing long-term dependencies in sequential data, making them suitable for tasks where context over a considerable time span is essential.

**3.3.5.2 Handling vanishing/exploding gradients**: LSTMs mitigate the vanishing gradient problem often encountered in traditional RNNs, allowing for more stable training over long sequences.

* + - 1. **Variable-Length Sequences**: LSTMs can process sequences of varying lengths, making them adaptable to tasks with dynamic input sizes.
    1. **Applications**:

LSTMs are widely used in natural language processing (NLP) for tasks like text generation, sentiment analysis, and machine translation.

They are also applied in time series forecasting, speech recognition, and other domains involving sequential data analysis.

In summary, Long Short-Term Memory (LSTM) networks are a type of recurrent neural network architecture that uses memory cells with gates to capture long-term dependencies and process sequential data effectively. They have become a cornerstone in many applications requiring understanding and prediction of sequential patterns.

**3.4 AutoRegressive Integrated Moving Average (ARIMA)**

**3.4.1 Introduction**

AutoRegressive Integrated Moving Average (ARIMA) is a popular statistical method used for time series forecasting.

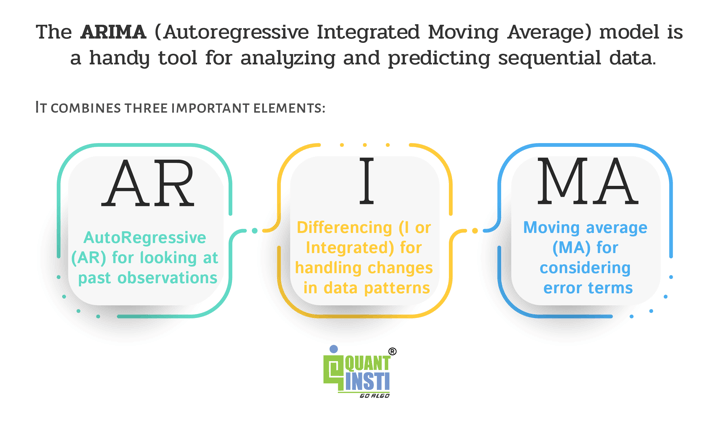
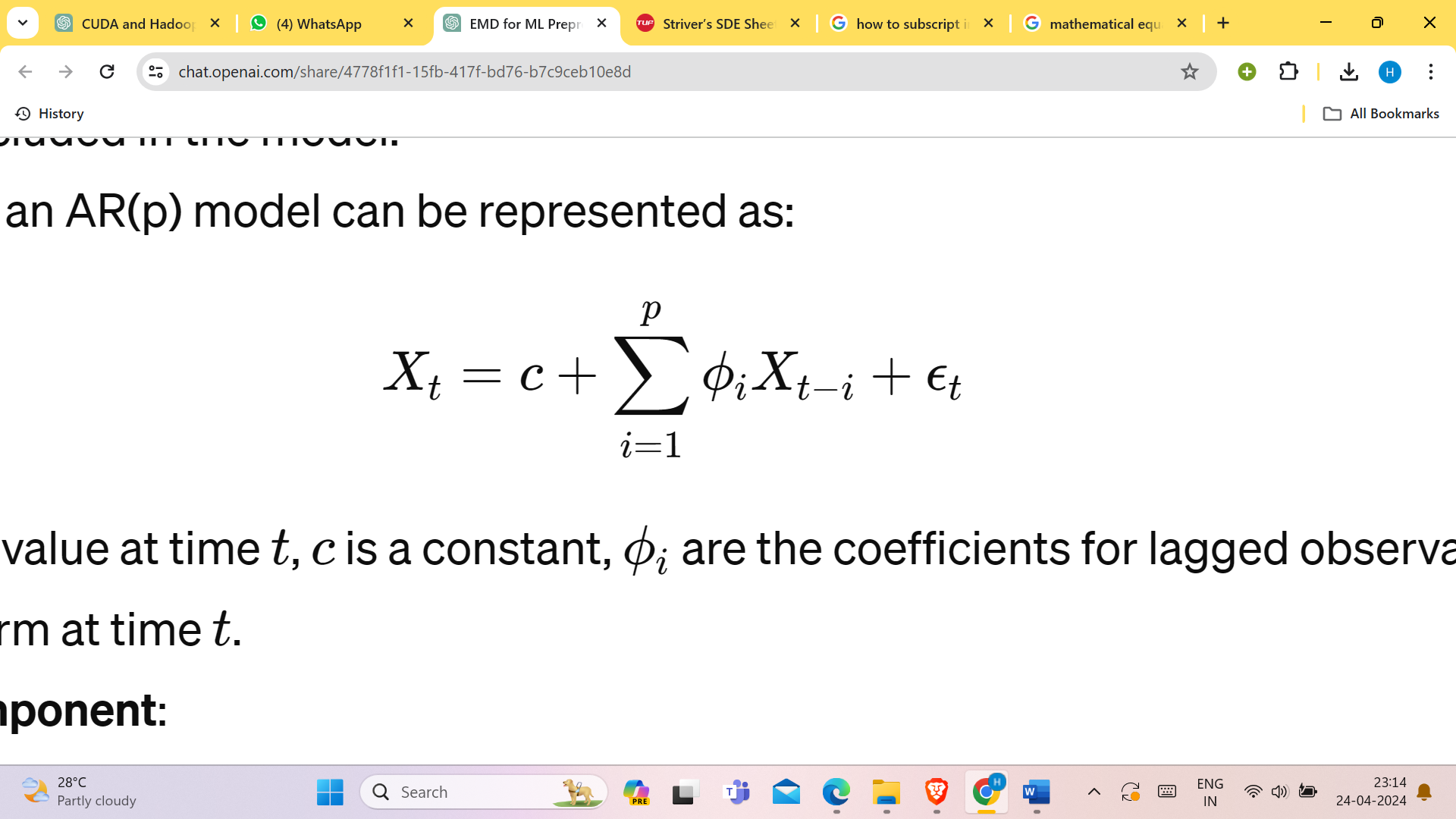


Figure 3.5 Introduction to ARIMA

how ARIMA (AutoRegressive Integrated Moving Average) works:

**3.4.2 AutoRegressive (AR) Component**: The autoregressive component of ARIMA models the relationship between an observation and a number of lagged observations (i.e., previous time steps). The order of the autoregressive component, denoted as *p*, indicates the number of lagged observations included in the model. Mathematically, the AR component can be represented as

(equ. 14)

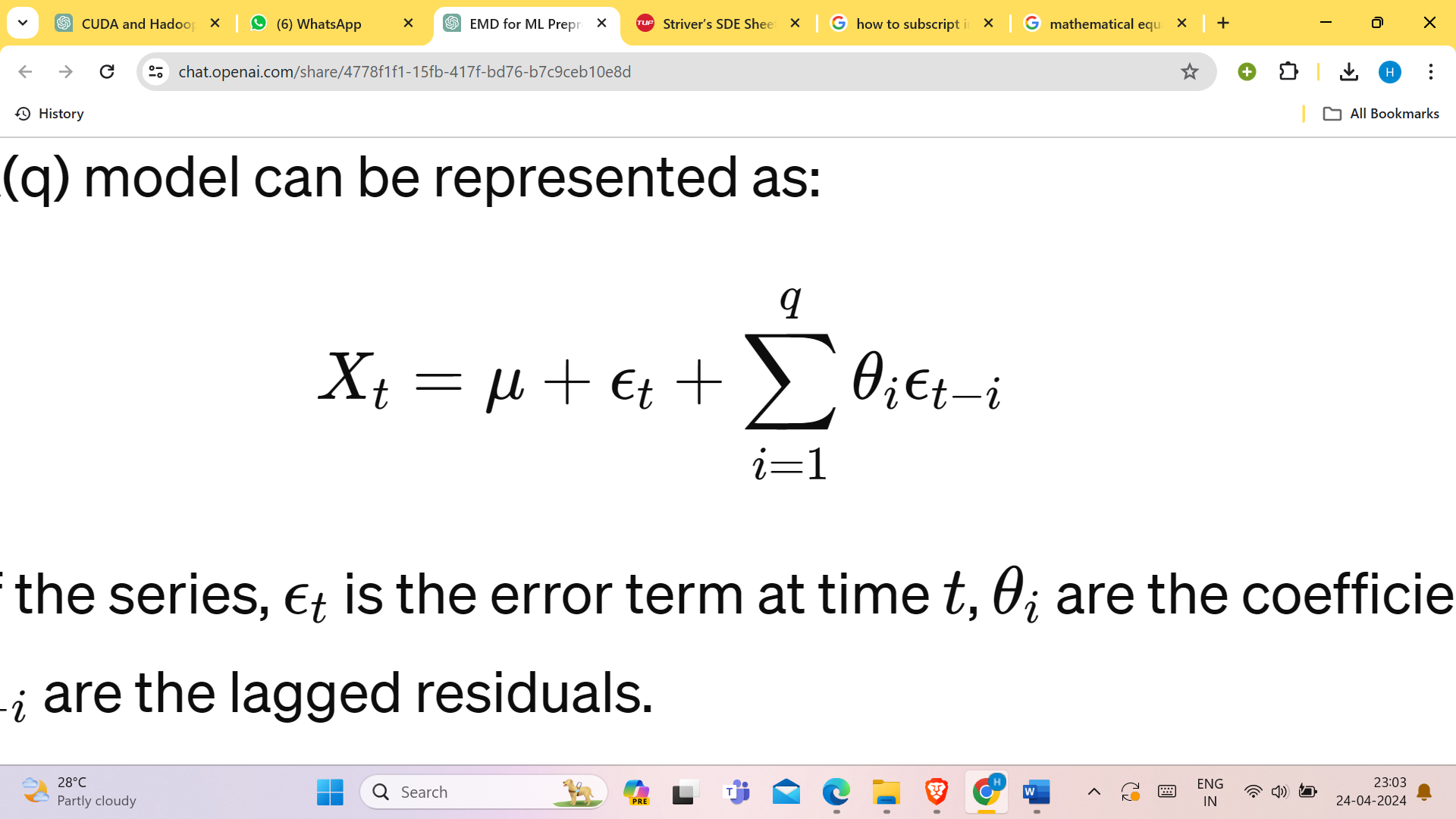
where 𝑋𝑡​ is the value at time 𝑡, 𝑐 is a constant, 𝜙𝑖​ are the coefficients for lagged observations, and 𝜖𝑡​ is the error term at time 𝑡.

**3.4.3 Integrated (I) Component**: The integrated component of ARIMA is responsible for differencing the time series data to make it stationary. Stationarity is a key assumption of ARIMA models, as they perform best on stationary data. The order of differencing, denoted as *d*, indicates the number of differences needed to achieve stationarity. Mathematically, differencing can be represented as:

*Yt*​ = *Xt*​−*Xt-1*​ (equ. 15)

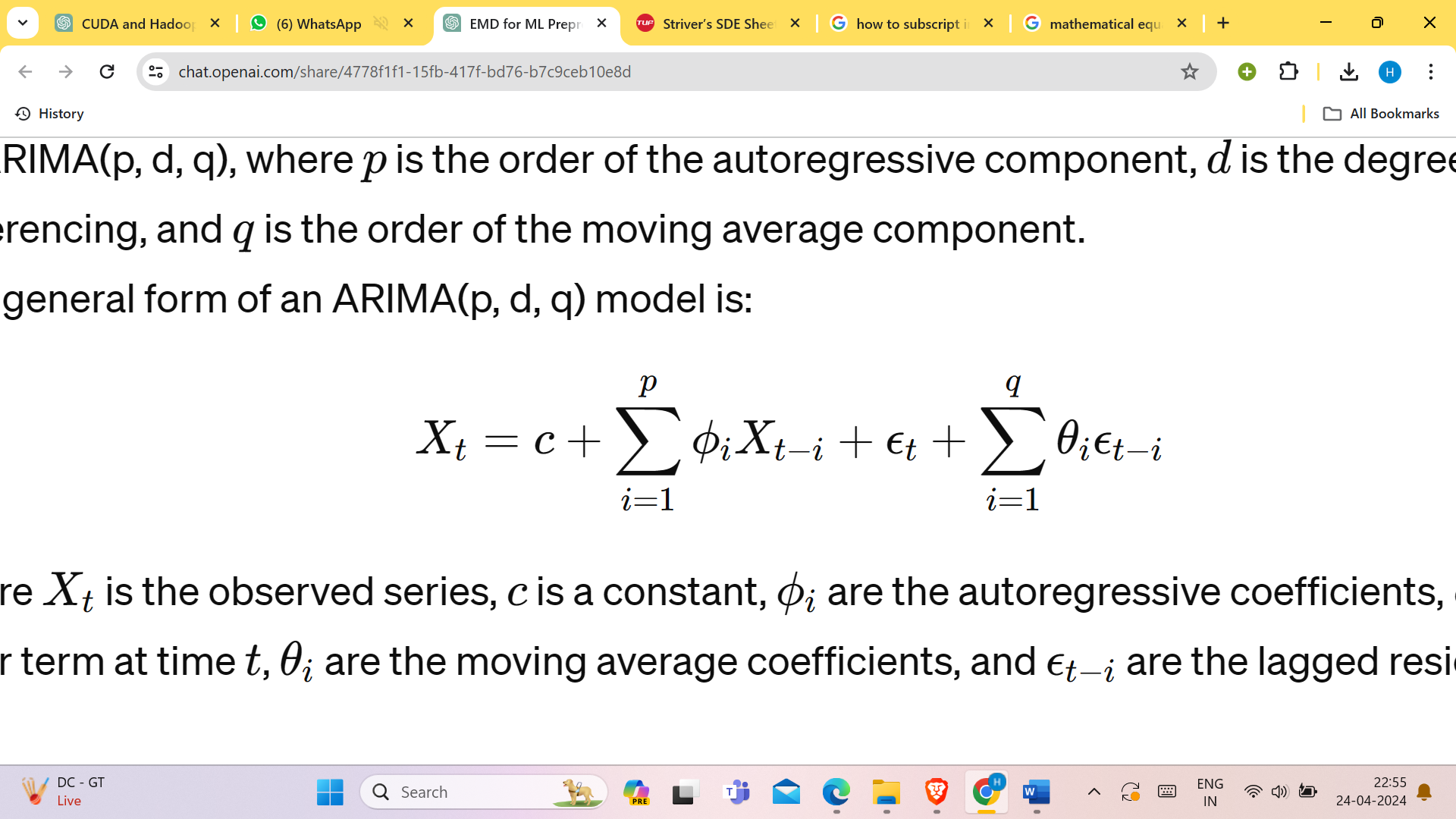
where 𝑌𝑡 is the differenced series.

**3.4.4 Moving Average (MA) Component**: The moving average component of ARIMA models the relationship between an observation and the residual errors from a moving average model applied to lagged observations of the time series. The order of the moving average component, denoted as *q*, indicates the number of lagged errors included in the model. Mathematically, the MA component can be represented as:

(equ. 16)

where 𝜇 is the mean of the series, 𝜖𝑡 is the error term at time 𝑡, 𝜃𝑖​ are the coefficients for lagged residual errors, and 𝜖𝑡−𝑖 are the lagged residuals.

**3.4.5 Model Identification and Parameter Estimation**: The orders *p*, *d*, and *q* of the ARIMA model need to be determined based on the characteristics of the data. This process, known as model identification, often involves analyzing the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the differenced data. Once the orders are determined, the next step is to estimate the coefficients of the AR, I, and MA terms in the ARIMA model using methods such as maximum likelihood estimation or least squares estimation.

(equ. 17)

where 𝑋𝑡​ is the observed series, 𝑐 is a constant, 𝜙𝑖 are the autoregressive coefficients, 𝜖𝑡 is the error term at time 𝑡, 𝜃𝑖​ are the moving average coefficients, and 𝜖𝑡−𝑖 are the lagged residuals.

**3.4.6 Model Fitting**: With the parameter estimates in hand, the ARIMA model is fitted to the differenced time series data. This involves iterating through the data and updating the model parameters to minimize the difference between the observed and predicted values.

**3.4.7 Diagnostic Checking**: After fitting the ARIMA model, it's essential to perform diagnostic checks to assess its goodness of fit. This may involve analyzing the residuals to ensure they are white noise (i.e., independent and identically distributed with zero mean and constant variance).

**3.4.8 Forecasting**: Once the ARIMA model is fitted and validated, it can be used to make forecasts for future time points. Forecasts are generated by extrapolating the fitted model into the future based on the observed data and the estimated model parameters.

Overall, ARIMA is a versatile and widely used method for time series forecasting and analysis. Its ability to capture temporal dependencies and model non-stationary data makes it valuable for a wide range of applications, including finance, economics, and environmental science.

Top of Form

**Chapter 4 Results**

**4.1 ARIMA Model (Linear IMFs)**

The ARIMA model, applied to the dataset using linear Intrinsic Mode Functions (IMFs), achieved the following accuracies:

1. IMF 1: 2.04%
2. IMF 2: 2.09%
3. IMF 3: 1.56%
4. IMF 4: 10.41%
5. IMF 5: 6.66%
6. IMF 6: 4.70%

The ARIMA model demonstrates varying levels of accuracy across different IMFs, with the highest accuracy observed in IMF 4 (10.41%) and the lowest in IMF 3 (1.56%).

**4.2 LSTM Model (Nonlinear IMFs)**

The LSTM model, trained on the dataset using nonlinear Intrinsic Mode Functions (IMFs), underwent training for 100 epochs with the following results:

* Training Loss: 0.0670
* Training Accuracy: 47.71%
* Validation Loss: 0.0233
* Validation Accuracy: 24.56%
* Test Loss: 0.0063
* Test Accuracy: 18.41%

The LSTM model exhibits moderate training accuracy (47.71%) but comparatively lower validation (24.56%) and test accuracies (18.41%). These results suggest potential overfitting during training, as evidenced by the disparity between training and validation/test accuracies.

**CHAPTER 5 Conclusion**

The project conclusively demonstrated the efficacy of Empirical Mode Decomposition (EMD) techniques for analyzing and predicting soil moisture dynamics directly from Intrinsic Mode Functions (IMFs). By embracing the versatility of EMD and tailoring model selection to the unique characteristics of the IMFs, we achieved a notable enhancement in prediction accuracy, surpassing the limitations associated with relying solely on a single model type. This strategic fusion of data decomposition and model adaptation not only refined our understanding of soil moisture dynamics but also unlocked new avenues for precise and reliable predictions.

The success of this approach underscores its inherent value as a robust framework for comprehensively analyzing and predicting soil moisture dynamics. Such insights hold immense significance across a spectrum of applications spanning agriculture, environmental science, and hydrology. In agriculture, informed decisions regarding irrigation scheduling, crop management, and water conservation strategies can be made with greater confidence and precision. Environmental science benefits from a more nuanced understanding of ecosystem health, water availability, and climate change impacts. In hydrology, the ability to forecast soil moisture dynamics aids in flood forecasting, drought mitigation, and water resource management, contributing to resilient and sustainable water management practices.

In conclusion, the developed data analysis pipeline stands as a testament to the power of integrated methodologies in soil moisture data analysis and prediction. Its robustness, adaptability, and predictive prowess pave the way for further advancements and applications in the field. Moving forward, future endeavors may delve into fine-tuning model hyperparameters to optimize performance, exploring additional feature engineering techniques to extract deeper insights, and integrating real-time data streams for continuous monitoring and adaptive prediction. These evolutionary steps hold promise for shaping the next frontier of soil moisture management, where data-driven solutions drive actionable insights and informed decision-making.

**CHAPTER 6 Future Scope**

1. **Ensemble Forecasting Approaches:** Ensemble forecasting methods, which combine predictions from multiple models or data sources, hold promise for improving the reliability and robustness of soil moisture forecasts. Future studies can investigate the implementation of ensemble learning techniques, such as model averaging and weighted combinations, to mitigate uncertainties and enhance forecast accuracy.
2. **Incorporation of Machine Learning Techniques:** Advancements in machine learning algorithms, such as deep learning and ensemble methods, offer opportunities for developing more sophisticated models for soil moisture analysis and forecasting. Future research can explore the use of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models to capture complex spatiotemporal relationships and nonlinear dynamics in soil moisture data.
3. **Integration with Decision Support Systems:** Integrating soil moisture forecasts with decision support systems (DSS) and agricultural management tools can facilitate real-time decision-making for farmers, water managers, and policymakers. Future research can explore the development of user-friendly DSS platforms that provide tailored recommendations and adaptive strategies based on forecasted soil moisture conditions.
4. **Assessment of Climate Change Impacts:** Climate change is expected to alter precipitation patterns, temperature regimes, and hydrological cycles, leading to significant impacts on soil moisture dynamics. Future research can focus on assessing the potential effects of climate change on soil moisture availability, drought frequency, and ecosystem resilience, incorporating projections from climate models into forecasting frameworks.

**Chapter 7 Reference**

1**.**EMDSignal. (n.d.). Retrieved from <https://pypi.org/project/EMD-signal/>

2. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[Link: <https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735>]

3.Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons. [**https://www.wiley.com/enus/Time+Series+Analysis%3A+Forecasting+and+Control%2C+5th+Edition-p-9781118675021**](https://www.wiley.com/enus/Time+Series+Analysis%3A+Forecasting+and+Control%2C+5th+Edition-p-9781118675021)**]**

4. Wu, Z., Huang, N. E., Long, S. R., & Peng, C. K. (2007). On the trend, detrending, and variability of nonlinear and nonstationary time series. Proceedings of the National Academy of Sciences, 104(38), 14889-14894.

<https://www.pnas.org/content/104/38/14889>

5.Huang, N. E., Wu, Z., Long, S. R., Arnold, K. C., Chen, X., & Blank, K. (2009). On instantaneous frequency. Advances in Adaptive Data Analysis, 1(2), 177-229. <https://www.worldscientific.com/doi/abs/10.1142/S1793536909000047>

6. Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons. <https://www.wiley.com/enus/Time+Series+Analysis%3A+Forecasting+and+Control%2C+5th+Edition-p-9781118675021>

7.Investopedia. (n.d.). Autoregressive integrated moving average (ARIMA). Retrieved from <https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp>

8. Olah, C. (2015, August). Understanding LSTM Networks. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/