



GRADUATE SCHOOL  
EAST TENNESSEE STATE UNIVERSITY

East Tennessee State University  
Digital Commons @ East  
Tennessee State University

---

Electronic Theses and Dissertations

Student Works


---

12-2023

## Implementation of Hierarchical and K-Means Clustering Techniques on the Trend and Seasonality Components of Temperature Profile Data

Emmanuel Ogedegbe  
*East Tennessee State University*

Follow this and additional works at: <https://dc.etsu.edu/etd>

 Part of the [Applied Mathematics Commons](#), [Computer Sciences Commons](#), [Data Science Commons](#),  
and the [Statistics and Probability Commons](#)

---

### Recommended Citation

Ogedegbe, Emmanuel, "Implementation of Hierarchical and K-Means Clustering Techniques on the Trend and Seasonality Components of Temperature Profile Data" (2023). *Electronic Theses and Dissertations*. Paper 4270. <https://dc.etsu.edu/etd/4270>

This Thesis - embargo is brought to you for free and open access by the Student Works at Digital Commons @ East Tennessee State University. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ East Tennessee State University. For more information, please contact [digilib@etsu.edu](mailto:digilib@etsu.edu).

Implementation of Hierarchical and K-Means Clustering Techniques on the Trend  
and Seasonality Components of Temperature Profile Data

---

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

---

by

Emmanuel Aigbokhavbo Ogedegbe

December 2023

---

Michele Lynn Joyner, Ph.D., Chair

Jeff Randall Knisley, Ph.D.

Mostafa Zahed, Ph.D.

Keywords: Time series data, K-Means Clustering, Hierarchical Clustering, Time  
Series Decomposition. Dynamic Time Warping

## ABSTRACT

Implementation of Hierarchical and K-Means Clustering Techniques on the Trend  
and Seasonality Components of Temperature Profile Data

by

Emmanuel Aigbokhavbo Ogedegbe

In this study, time series decomposition techniques are used in conjunction with K-means clustering and Hierarchical clustering, two well-known clustering algorithms, to climate data. Their implementation and comparisons are then examined. The main objective is to identify similar climate trends and group geographical areas with similar environmental conditions. Climate data from specific places are collected and analyzed as part of the project. The time series is then split into trend, seasonality, and residual components. In order to categorize growing regions according to their climatic inclinations, the deconstructed time series are then submitted to K-means clustering and Hierarchical clustering with dynamic time warping. In order to understand how different regions' climates compare to one another and how regions cluster based on the general trend of the temperature profile over the course of the full growing season as opposed to the seasonality component for the various locations, the created clusters are evaluated.

Copyright by Emmanuel Aigbokhavbo Ogedegbe 2023

All Rights Reserved

## DEDICATION

I dedicate this thesis to my beloved family whose enduring love and encouragement made the completion of this academic journey possible and a success. I am most grateful to Augustine Ogedegbe for his continued guidance and support and for being a worthy model and mentor to me in the course of this research, and to my parents, Late Engr. Francis Ogedegbe and Rosemary Ogedegbe, for teaching me resilience and patience. I love you all.

## ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Dr. Michele Joyner for her patience and guidance through all the stages of researching and writing this thesis. Her comforting words of encouragement saw me through moments of despair and utter confusion, and thanks to her influence as a teacher and a mother, I have better clarity on my career aspirations and look forward to a thriving career. To my committee members, Dr. Jeff Knisley, and Dr. Mostafa Zahed, and all the professors who have contributed to my academic maturity these last two years, I am immensely grateful. I am thankful to friends and colleagues for challenging me in worthwhile ways and for their unwavering support. I want to acknowledge and appreciate these individuals for their moral and academic support throughout my academic journey in the past two years at ETSU; Peter Ogunmefun, Mark-Daniels Tamakloe, Dayo Victor, to mention but a few. Special appreciation to Cecilia Ashibel for her love, matchless kindness, and encouragement, and to Janet Kireta for being a friend indeed. Above all, I dedicate this thesis to God Almighty in appreciation for all that has been, all that is, and all that is yet to be and most especially for His mercy and grace.

## TABLE OF CONTENTS

ABSTRACT . . . . .	2
DEDICATION . . . . .	4
ACKNOWLEDGMENTS . . . . .	5
LIST OF TABLES . . . . .	9
LIST OF FIGURES . . . . .	11
1 INTRODUCTION . . . . .	12
2 DATA . . . . .	16
2.1 Overview . . . . .	16
2.2 Pre-processing of Data . . . . .	17
3 Time Series Decomposition . . . . .	22
3.1 Overview of Algorithm . . . . .	23
3.2 Implementation . . . . .	25
4 Dynamic Time Warping . . . . .	36
4.1 A toy example on DTW . . . . .	38
4.2 The advantages of DTW over Euclidean distance for time series data . . . . .	45
5 Clustering . . . . .	49
5.1 Hierarchical Clustering . . . . .	49
5.2 K-means Clustering . . . . .	53
6 Results . . . . .	56
6.1 Clustering Based on Trend Assuming a Daily Period . . . . .	57
6.2 K-Means Clustering . . . . .	61

6.3	Clustering Based on Trend Assuming a Weekly Period . . . . .	66
6.4	Clustering Based on the Seasonality Component Assuming Daily Seasonality . . . . .	70
7	Summary and Future Work . . . . .	75
	BIBLIOGRAPHY . . . . .	77
	VITA . . . . .	82



## LIST OF TABLES

1	Daily and Weekly lag for each location . . . . .	29
2	DTW distance matrix . . . . .	39
3	DTW distance matrix . . . . .	40
4	DTW distance matrix . . . . .	41
5	DTW distance matrix . . . . .	42
6	Distance Matrix For Daily Trend With The Labels 1 Through 26 Representing The 26 Different Growing Locations Considered In This Study: 1:COH1, 2:DEH1, 3:GAH1, 4:GEH1, 5:IAH1, 6:IAH2, 7:IAH3, 8:IAH4, 9:ILH1, 10:INH1, 11:MIH1, 12:MNH1, 13:MOH1, 14:NCH1, 15:NEH1, 16:NEH2, 17:NYH2, 18:NYH3, 19:OHH1, 20:ONH2, 21:SCH1, 22:TXH1, 23:TXH2, 24:TXH4, 25:WIH1, 26:WIH2. . . . .	46
7	Distance Matrix For Daily Seasonality With The Labels 1 Through 26 Representing The 26 Different Growing Locations Considered In This Study: 1:COH1, 2:DEH1, 3:GAH1, 4:GEH1, 5:IAH1, 6:IAH2, 7:IAH3, 8:IAH4, 9:ILH1, 10:INH1, 11:MIH1, 12:MNH1, 13:MOH1, 14:NCH1, 15:NEH1, 16:NEH2, 17:NYH2, 18:NYH3, 19:OHH1, 20:ONH2, 21:SCH1, 22:TXH1, 23:TXH2, 24:TXH4, 25:WIH1, 26:WIH2. . . . .	47
8	Original DTW Distance Matrix for Example . . . . .	50
9	Updated DTW Distance Matrix for Example . . . . .	51
10	Second Updated DTW Distance Matrix for Example . . . . .	52
11	Clusters using $K=3$ in the K-means Clustering Algorithm . . . . .	65
12	Clustering using $K=5$ in the K-means Clustering Algorithm . . . . .	66

13	Clustering using $K=9$ in the K-means Clustering Algorithm . . . . .	66
----	--	----

## LIST OF FIGURES

1	Map of Growing locations in North America . . . . .	17
2	All the timestamp labels for a daily period for the entire growing season at one growing location (COH1) . . . . .	19
3	The time step labels for one daily period after relabelling the ‘off’ time stamps to the closest half-hour or hourly label for each day of the growing season for location COH1 . . . . .	19
4	Plot showing data for location COH1 with and without missing values	20
5	A Colorado Location Temperature Data with Rolling Mean and STD	24
6	Location COH1 Raw Data Plot . . . . .	27
7	Autocorrelation plot for growing location COH1 . . . . .	27
8	Trend component for location COH1 assuming daily seasonality . . .	31
9	Seasonality component for location COH1 assuming daily seasonality	31
10	Residual component for location COH1 assuming daily seasonality . .	32
11	Plot of the trend component for all growing locations assuming daily seasonality . . . . .	32
12	Seasonality component for all growing locations assuming daily sea- sonality . . . . .	33
13	Trend component for all growing locations assuming weekly seasonality	33
14	Seasonality component for all growing locations assuming weekly sea- sonality . . . . .	34
15	Heat map of accumulated cost matrix for toy example [24] . . . . .	43
16	Heat map of accumulated cost matrix for toy example [24] . . . . .	44

17	Alignment Graph for warped path on simple example . . . . .	45
18	Dendrogram of hierarchical clustering of simple example . . . . .	53
19	Denodogram using hierarchical clustering on the trend component of the time series decomposition when assuming a daily seasonality period.	58
20	Clusters based on a threshold level of 1200 in the hierarchical clustering algorithm on the trend component of the time series decomposition assuming a daily period for seasonality . . . . .	60
21	Clusters based using $K=6$ clusters in the K-means clustering algorithm on the trend component of the time series decomposition assuming a daily period for seasonality . . . . .	63
22	Dendrogram using hierarchical clustering on the trend component of the time series decomposition when assuming a weekly seasonality period	68
23	Clusters based on a threshold level of 1200 in the hierarchical clustering algorithm on the trend component of the time series decomposition assuming a weekly period for seasonality . . . . .	69
24	Dendrogram using hierarchical clustering on the seasonal component of the time series decomposition when assuming a daily seasonality period	71
25	Clusters based on a threshold level of 60 in the hierarchical clustering algorithm on the seasonal component of the time series decomposition assuming a daily period for seasonality . . . . .	74

## 1 INTRODUCTION

In recent years, understanding the intricate dynamics of climate trends has become more and more important recently, particularly in the context of agriculture and food security [4]. In order to optimize agricultural practices, forecast crop yields, and create efficient climate change mitigation plans, it is essential to be able to recognize and assess climate patterns and seasonality across a variety of growing regions [15, 20]. To achieve these goals, clustering algorithms have become important resources for examining patterns in massive data sets in order to accomplish these objectives. In this work, we'll concentrate on putting two popular clustering algorithms, hierarchical clustering and K-means clustering, into practice and contrasting them in order to analyze climatic data across various growth regions utilizing time series decomposition.

Time series, which include various climatic parameters including temperature, precipitation, humidity, and wind patterns throughout time, are widely used to describe climate data. Time series decomposition techniques are necessary in order to study such data properly. We can obtain better insights into long-term climatic trends and recurring patterns that happen throughout particular time periods by untangling essential elements like trends and seasonality.

Unsupervised learning methods that are frequently employed in exploratory data analysis and pattern recognition tasks include hierarchical clustering and K-means clustering [10]. By iterative merging or dividing data points depending on their similarity, hierarchical clustering creates a hierarchy of clusters that eventually forms a dendrogram, which resembles a tree [23]. On the other hand, K-means clustering

partitions data points into  $k$  distinct clusters, aiming to minimize the within-cluster variance [12]. Both algorithms offer unique advantages and have been extensively used in various domains, including climate analysis.

In this research endeavor, our primary objective is to evaluate the differences in using Hierarchical and K-means Clustering in conjunction with the dynamic time warping distance metric on trend or seasonal data found through time series decomposition. By applying time series decomposition techniques, we extract the underlying trends and seasonality from the climate data collected at multiple locations in the United States and Canada. Subsequently, we employ Hierarchical and K-means Clustering algorithms to group these locations based on their climate patterns, aiming to identify clusters with similar trends and seasonality.

Past projects have explored the application of clustering techniques to climate data, highlighting their potential in identifying distinct climate patterns across different geographical regions. For example, Taylor et al [36] utilized cubic spline interpolation and K-means clustering to identify which geographic growing locations are most comparable based on their climates throughout the growing season for maize. They utilized cubic spline interpolation to smooth the data and create data at the same time point for comparison. In section 3 we address some of the limitations inherent in using the cubic spline interpolation which we hope to improve upon using time series decomposition. Additionally in section 4 we discuss the dynamic time warping distance metric which relieves the restrictions of comparing time series at the same time points. In summary, in this thesis, we intend to seek the underlying trend or seasonality within the data and use dynamic time warping to obtain the ‘best match’

between the growing locations, allowing for groupings based on the growing pattern even if similar growing pattern occur during offset periods of the growing season.

Taylor et al [35] sought to cluster maize growing locations to control for climate prior to implementing machine learning techniques to predict over-performing hybrid maize plants based on the genetic composition of the plants in these climate-controlled locations. This thesis seek to address the initial clustering portion to more effectively control for climate prior to the implementation of machine learning to predict crop yield based on genetic factors. While these previous studies have provided valuable insights, there is a need for more in-depth exploration of clustering algorithms for climate pattern analysis. The implementation and comparison of Hierarchical clustering and K-means clustering techniques using dynamic time warping and time series decomposition, specifically in the context of climate data across growing locations, can provide the proof of concept for combining these techniques for clustering climate data. By leveraging time series decomposition methods to extract trends and seasonality, these clustering approaches can provide a deeper understanding of long-term climate patterns and recurring trends, which are crucial for optimizing agricultural practices and developing climate change adaptation strategies. Moreover, the exploration of advanced clustering techniques for climate data can greatly benefit machine learning models used in previous projects. By clustering climate data based on similar patterns, the resulting clusters can serve as informative features or inputs for machine learning algorithms. This strategy may improve the effectiveness of predictive models, allowing for more precise forecasts of crop yields, ideal planting periods, and dangers associated to the environment. The resulting clusters can also be used

to direct the creation of regional models and focused interventions, because various clusters may call for different agricultural approaches and adaptation techniques.

Finally, the motivation for exploring and comparing hierarchical and K-means is to better comprehend complicated climatic dynamics and their implications for agriculture. Clustering techniques for studying climate trends across growing locations employing dynamic time warping and time series decomposition were developed. We can extract critical elements from time series data, uncover related climatic patterns, and help machine learning models become more precise and localized by utilizing clustering methods. A detailed description of the project will be given in the parts that follow. We'll start by giving a summary of data processing techniques in section 2. We give a general review of time series decomposition in section 3 before going into great detail into dynamic time warping in section 4. The two different clustering algorithms are the main topic of Section 5. Section 6 presents the findings, while Section 7 provides a summary and suggestions for further work.



## 2 DATA

### 2.1 Overview

The data set used in this study was obtained from the “Genomes to Fields” project which is a research project that aims to improve plant breeding and crop productivity by using genomic data [26, 31]. It involves collaboration between researchers and farmers to collect data on how different varieties of crops perform in different environments. This data is then used to develop better crop varieties that are more resilient to various stressors, such as drought, disease, or pests. The project involves sequencing the genomes of different crop varieties and using that information to develop genetic markers that can be used to identify desirable traits. These markers can then be used to breed crops with those traits more efficiently.

In this thesis we analyze the 2019 data as it was the most recent data prior to the 2020 disruption due to the pandemic and thus most complete. The data set consists of information on 26 different growing locations across the US, Canada and Germany with locations including Colorado, Delaware, Georgia, Iowa, Illinois, Indiana, Michigan, Minnesota, Missouri, North Carolina, Nebraska, New York, Ohio, Ontario, South Carolina, Texas, and Wisconsin with a total of 227278 observations (See Figure 1 for a map of locations included in this study). There are 23 variables in the data set, including field locations, station ID, NWS Network, NWS Station, Date key, Month, Day, Year, Time, Temperature, Dew Point, Relative Humidity, Solar Radiation, Rainfall, Wind Speed, Wind Direction, Wind Gust, Soil Temperature, Soil Moisture, Soil EC, UV Light, PAR, and CO<sub>2</sub>. The data set is in a comma-

separated values (CSV) file format and is stored on a local computer. The data was incomplete with missing values, so data cleaning was carried out on the data set prior to analysis.

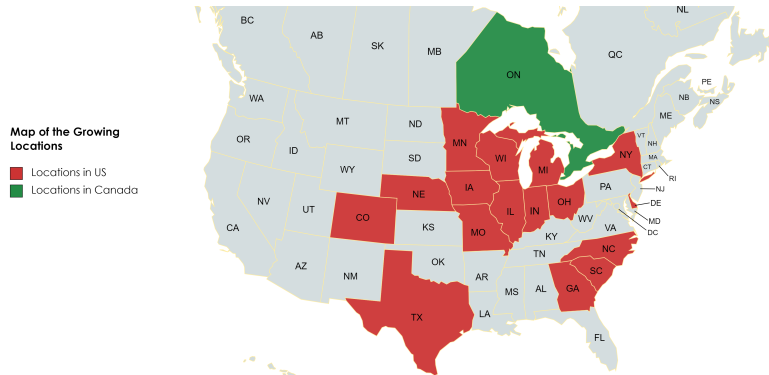


Figure 1: Map of Growing locations in North America

## 2.2 Pre-processing of Data

During the data cleaning process, several steps were undertaken to ensure the accuracy and consistency of the dataset. Since the objective of clustering was to group locations together to control for climate, we initially examined all of the climate data. Unfortunately, most of the climate variables have more than half of the data missing across all growing locations. Therefore, although precipitation, humidity and a variety of other factors should ideally be included in a climate study, in this thesis, we focus on temperature data due to availability. Therefore, the objective was to analyze the temperature measurements of the growing locations across the 2019 growing season.

To most effectively utilize time series decomposition techniques, we needed to have uniform time measurements from one day to the next, i.e. the same number of measurements for each day within one growing season, to determine the periodicity

of the series. As such we took the following steps to create a uniform time series for each of the temperature profiles for each of the growing locations.

- **Identifying Missing Days:** It was crucial to check for missing days in the dataset. We extracted the unique dates in the data and sorted in ascending order. In this particular dataset, no missing days were found, except for one dataset in which there were only sporadic measurement during the start of the growing season. For this dataset, we restricted the time period to when the measurements were more consistent which no longer included any missing days.
- **Handling Different Time Labels:** Most datasets contained different time labels for what we assume were similar ‘time-of-day’ measurements. For example, in Figure 2, we see a variety of measurements (96 different labels in all). On one day, temperature measurements were taken at 0:25, 0:54, 1:25, 1:54, etc; whereas, on a different day the label might indicate the temperature was measured at times 0:30, 1:00, 1:30, 2:00 etc. These labels were believed to represent either a half-hour or hourly interval. Inconsistent labeling would eventually cause problems when filling in missing data and in determining periodicity. To ensure consistency, the labels were renamed to be consistent from day to day. We did not change the data, only the labels for the measurements. For example we replace 0:25 with 0:30 and 0:54 with 1:00 to reflect the nearest time interval. The corrected labels are given in Figure 3. Notice that when the labels were modified to account for half-hour and hourly intervals, there were a total of only 48 different time steps for one daily period, as opposed to the 96 different times indicated initially.

```

['0:00', '0:25', '0:30', '0:56', '1:00', '1:25', '1:30', '1:56', '2:00', '2:25', '2:30', '2:56', '3:00', '3:25', '3:30', '3:56', '4:00', '4:25', '4:30', '4:56', '5:00', '5:25', '5:30', '5:56', '6:00', '6:25', '6:30', '6:56', '7:00', '7:25', '7:30', '7:56', '8:00', '8:25', '8:30', '8:56', '9:00', '9:25', '9:30', '9:56', '10:00', '10:25', '10:30', '10:56', '11:00', '11:25', '11:30', '11:56', '12:00', '12:25', '12:30', '12:56', '13:00', '13:25', '13:30', '13:56', '14:00', '14:25', '14:30', '14:56', '15:00', '15:25', '15:30', '15:56', '16:00', '16:25', '16:30', '16:56', '17:00', '17:25', '17:30', '17:56', '18:00', '18:25', '18:30', '18:56', '19:00', '19:25', '19:30', '19:56', '20:00', '20:25', '20:30', '20:56', '21:00', '21:25', '21:30', '21:56', '22:00', '22:25', '22:30', '22:56', '23:00', '23:25', '23:30', '23:56']

```

96

Figure 2: All the timestamp labels for a daily period for the entire growing season at one growing location (COH1)

```

['0:00', '0:30', '1:00', '1:30', '2:00', '2:30', '3:00', '3:30', '4:00', '4:30', '5:00', '5:30', '6:00', '6:30', '7:00', '7:30', '8:00', '8:30', '9:00', '9:30', '10:00', '10:30', '11:00', '11:30', '12:00', '12:30', '13:00', '13:30', '14:00', '14:30', '15:00', '15:30', '16:00', '16:30', '17:00', '17:30', '18:00', '18:30', '19:00', '19:30', '20:00', '20:30', '21:00', '21:30', '22:00', '22:30', '23:00', '23:30']

```

: 48

Figure 3: The time step labels for one daily period after relabelling the ‘off’ time stamps to the closest half-hour or hourly label for each day of the growing season for location COH1

- **Creating a New Data Range:** A new date range was constructed with the new labels for time, date, and temperature. Any missing temperature readings were automatically filled in with N/A.
- **Imputing Missing Values:** Linear interpolation was used to fill in missing values [16]. We first identify which specific times contain missing data. We then determine the nearest known values before and after the missing time point for each missing value. Based on the known values on each side of the missing value, we estimated the missing values using a line connecting the two known values. The assumption behind linear interpolation is that the known values are related linearly. By calculating a weighted average of the nearby known values,

where the weights depend on the relative placements of the missing point and the unknown points, one may approximate the values that are missing.

Let's say we have a missing value at point  $t$  and the nearest available data points are  $(t_1, T_1)$  and  $(t_2, T_2)$ ,

$$t_1 < t < t_2$$

The linear interpolation formula estimates the value  $T$  at time  $t$  as

$$T = T_1 + (t - t_1) \frac{(T_2 - T_1)}{(t_2 - t_1)}$$

where  $T$  represents the temperature at time  $t$ . We follow this procedure for each missing point in the data. It can be easily implemented using the `df.interpolate()` function in Python.

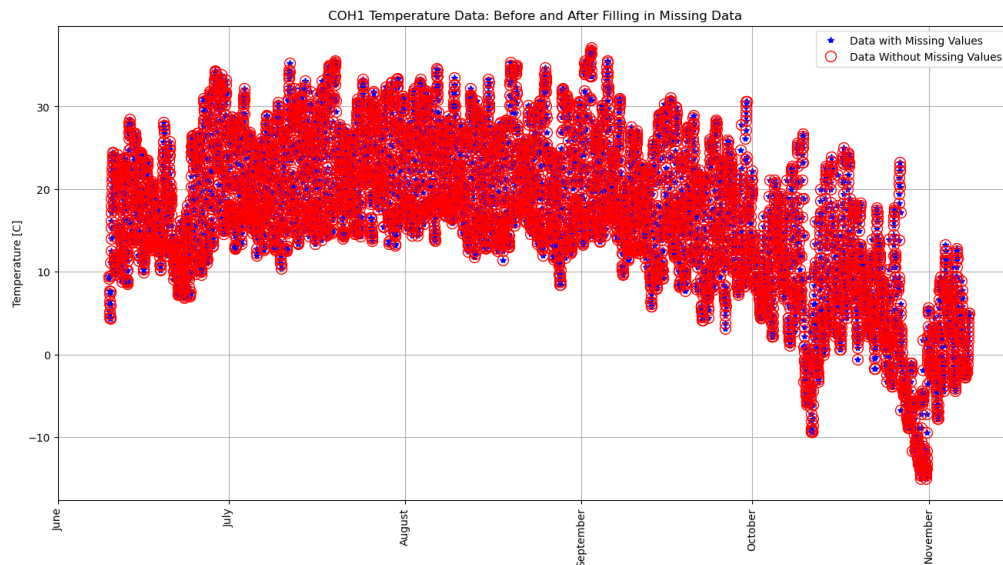


Figure 4: Plot showing data for location COH1 with and without missing values

In Figure 4, we show the raw data on the same graph as the processed data. Notice that we keep the integrity of the raw data while enabling us to create uniform daily measurements with no missing values. The uniform data allows us to determine potential periods to be used in the time series decomposition.

### 3 Time Series Decomposition

Time series decomposition is a methodological approach to smooth out noise, especially in datasets like temperature data that exhibit clear seasonal patterns [8]. By decomposing a time series into trend, seasonality, and residual components, it allows for a better understanding of the underlying patterns and provides a solid foundation for forecasting and analysis. When applied to temperature data, time series decomposition can effectively separate the seasonal variations, long-term trends, and irregular fluctuations present in the dataset.

The process of time series decomposition can be performed using various methods, such as classical decomposition, moving averages, or exponential smoothing [17]. Classical decomposition methods, such as the additive or multiplicative decomposition, involve separating the time series into its components using statistical techniques. Comparing time series decomposition to smoothing splines, both techniques aim to smooth out noise in the data. Smoothing splines are a non-parametric regression method that uses a flexible curve-fitting approach to estimate the underlying trend [37]. By adjusting the degree of smoothing, splines can effectively remove short-term fluctuations while preserving the overall shape of the data. However, time series decomposition offers several advantages over smoothing splines in capturing seasonal patterns and long-term trends.

Since time series decomposition explicitly separates the different components, it provides a clear interpretation of the trend and seasonality effects. Additionally, decomposed components can be easily modeled and forecasted individually, allowing for more accurate predictions. In contrast, smoothing splines may struggle to cap-

ture and separate seasonal variations as effectively as time series decomposition [12]. Splines tend to smooth the entire dataset as a whole, potentially averaging out the seasonal patterns. While it is possible to incorporate seasonal terms in smoothing splines, the explicit decomposition offered by time series techniques is often more intuitive and tailored for capturing seasonality.

### 3.1 Overview of Algorithm

The time series decomposition algorithm typically follows the additive or multiplicative model [30]. The additive model assumes that the observed time series can be decomposed into the *sum* of its components while the multiplicative model assumes that the observed time series can be decomposed into the *product* of its components. Additive models are models in which the variance of data doesn't change over different values of the time series. Multiplicative models are models in which the variance of data increases as the data increases or the seasonal pattern becomes more pronounced.

An additive model is linear in its components. The mathematical equation for an additive model can be represented as

$$Y(t) = T(t) + S(t) + R(t)$$

where

- $Y(t)$  represents the observed value of the time series at time  $t$ .
- $T(t)$  represents the trend component at time  $t$ .
- $S(t)$  represents the seasonality component at time  $t$ .



- $R(t)$  represents the residual (or error) component at time  $t$  [11].

In the multiplicative model, the trend and seasonal components are multiplied. It is non-linear, such as quadratic or exponential, and the trend is represented by a curved line, while the seasonality may have an increasing or decreasing frequency and amplitude over time. The mathematical equation for a multiplicative model can be represented as

$$Y(t) = T(t) \cdot S(t) \cdot R(t)$$

where:

- $Y(t)$  represents the observed value of the time series at time  $t$ .
- $T(t)$  represents the trend component at time  $t$ .
- $S(t)$  represents the seasonality component at time  $t$ .
- $R(t)$  represents the residual (or error) component at time  $t$  [11].

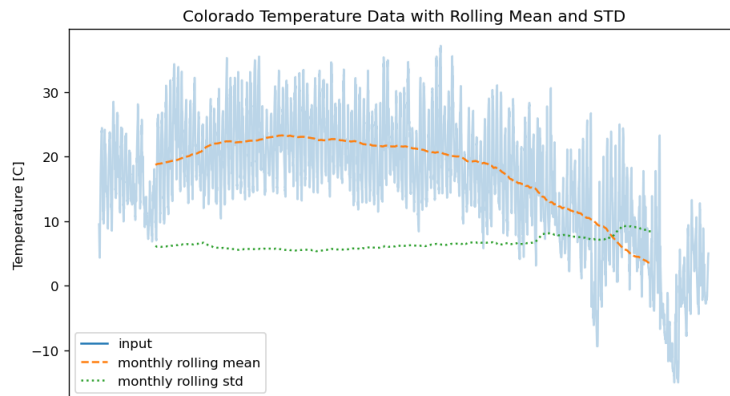


Figure 5: A Colorado Location Temperature Data with Rolling Mean and STD

In this study, we assume that the additive model is acceptable. Figure 5 shows one growing location’s temperature profile as a function of time where the red line gives the monthly rolling mean and the green line gives the rolling standard deviation using the Python command `df.mean()` and `df.std()` respectively. For an additive model, the amount of variation around the mean should remain fairly constant. Figure 5 indicates the standard deviation, and hence variation, over time doesn’t change considerably. In other words, there is little rise or fall in standard deviation with time. We note that all other growing locations display a similar pattern.

In Python, the `statsmodels` library provides the `seasonal_decompose()` function [9] which allows us to decompose a time series into its components. This function requires specifying the model as either “Additive” or “Multiplicative” and specifying the period of the seasonality. The output of the function includes the trend and seasonal components stored in an array, as well as the residuals, which represent the remaining variation after removing the trend and seasonal components. The original observed data is also stored for reference.

### 3.2 Implementation

The time series decomposition process involves the following steps [30]:

- **Trend Extraction:** The trend component represents the long-term pattern or direction of the time series. It captures the overall increasing or decreasing behavior of the data. Common techniques used for trend extraction include moving averages, polynomial regression, or exponential smoothing. The `seasonal_decompose()` python command uses a convolution filter. We refer the reader to

[30] for more details.

- **Seasonality Detection:** The seasonality component captures periodic patterns that repeat over fixed intervals. It represents the systematic variations occurring within a specific time frame. To determine the seasonality in the time series, we utilize the autocorrelation function (ACF) and autocorrelation coefficient. The ACF measures the correlation between a time series and its lagged versions at various time lags [7]. A significant correlation at specific lags suggests the presence of seasonality. In the ACF plot, if there is a sinusoidal-looking curve with peaks occurring at regular intervals, it indicates the existence of seasonality. The coefficient of correlation assumes a lag and measures the strength of the correlation at that particular lag [8, 32]. By analyzing the ACF plot and computing the autocorrelation coefficients, we can identify the seasonality pattern and estimate the lag at which the seasonality occurs.
- **Residual Calculation:** The residual component represents the random or unexplained variation in the time series after trend and seasonality have been accounted for. It contains the irregular or unpredictable fluctuations that cannot be explained by the trend or seasonality. Residuals are calculated by subtracting the trend and seasonality components from the original time series.

To illustrate the results of time series decomposition and the determination of seasonality, let's consider a specific growing location for maize crop yield as given in Figure 6.

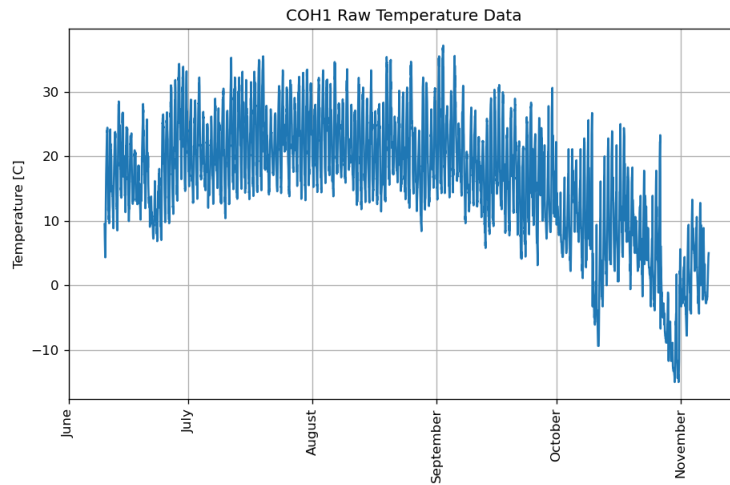


Figure 6: Location COH1 Raw Data Plot

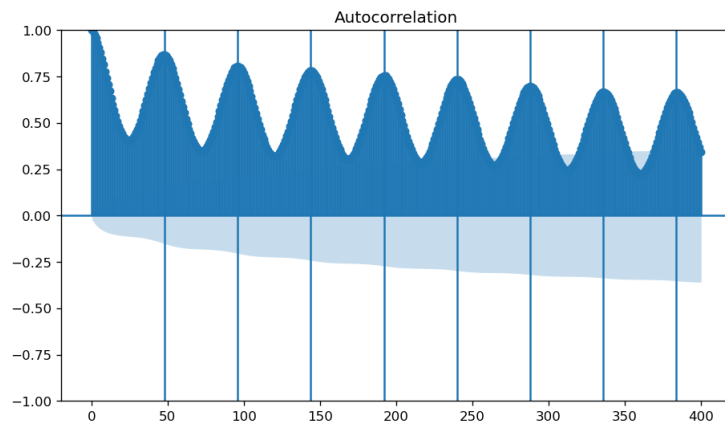


Figure 7: Autocorrelation plot for growing location COH1

We use the python command `df.autocorrelation` in the pandas package [27] to produce the auto correlation plot for our growing locations given in Figure 7. We first note that this figure has a sinusoidal like curve with peaks at uniform intervals. Based on this figure we assume a possible daily as well as weekly seasonality:

- Daily Seasonality: The ACF plot in Figure 7 displays vertical lines every 48

measurements at the peaks, which equates to a daily time period for this growing location. This indicates a seasonality of around 48 time measurements or daily seasonality. Additionally using the `df.autocorrelation` python command with  $lag = 48$  (number of daily time measurements), it is determined that the auto-correlation value for this daily period is 0.88 (88%), demonstrating a strong correlation between successive daily data.

- Weekly Seasonality: There is a distinct weekly trend in addition to the daily seasonality. Although it is not as strong as the daily seasonality, the weekly seasonality's auto-correlation is determined to be 0.72 (72%), showing a strong weekly auto-correlation.

The trends and patterns related to the daily seasonality may be obtained by breaking down the data into its periodic components and analyzing them, especially utilizing the given number of measurements for the daily period for each growing location. The autocorrelation values for both daily and weekly seasonality for each growing location is given in Table 1. Note that the autocorrelation ranges from 0.70 for location NEH1 to 0.88 for locations COH1, GAH1, IAH1, NEH2, NYH2, SCH1, and TXH1 which indicates a fairly strong daily seasonal autocorrelation for all growing locations. The weekly autocorrelation factors are slightly weaker with a range of 0.42 for location ILH1 to 0.81 for location MIH1 but with most values around 0.6 or higher. Nonetheless, we consider the trend and seasonality components assuming both periods for smoothed data upon which we perform clustering.

Using the decomposition algorithm with a periodicity of 48 (daily seasonality), we obtain the trend for this location in Figure 8, the seasonal component in Figure 9

Table 1: Daily and Weekly lag for each location

Locations	Daily	Weekly
COH1	0.88	0.72
DEH1	0.80	0.61
GAH1	0.88	0.76
GEH1	0.85	0.63
IAH1	0.88	0.79
IAH2	0.80	0.63
IAH3	0.82	0.68
IAH4	0.79	0.61
ILH1	0.77	0.42
INH1	0.79	0.59
MIH1	0.87	0.81
MNH1	0.74	0.57
MOH1	0.78	0.55
NCH1	0.81	0.60
NEH1	0.70	0.52
NEH2	0.88	0.73
NYH2	0.88	0.80
NYH3	0.87	0.80
OHH1	0.81	0.67
ONH2	0.87	0.79
SCH1	0.88	0.70
TXH1	0.88	0.73
TXH2	0.87	0.73
TXH4	0.85	0.75
WIH1	0.79	0.60
WIH2	0.79	0.60

and the leftover residual component in Figure 10. We note that the residual plot does still have some discernable pattern (i.e. not totally random) which might indicate the need to consider multiple seasonality periods simultaneously in future studies

Figure 11-14 illustrate the trend and seasonality components of the time series decomposition assuming a daily seasonality period (Figures 11 and 12) versus a weekly

seasonality period (Figure 13 and 14) for all locations. By examining these plots we can notice some similarities and differences between growing locations across the growing season. One thing we can immediately notice in Figure 11 is that location MNH1 has some extreme drops in temperature at several points in the growing season. For this study we did not adjust for any potential outliers, but in the future, one might consider whether there is a need to remove outliers. We also observe variability in the length of the growing season across locations as well as the timing for the start and end of the growing season. For example, the growing season for TXH2 has the earliest start in March while ONH2 doesn't begin until mid-June. Some locations such as GAH1, TXH1 and TXH2 end the season in mid-August while the end of others don't happen until November or December, like GEH1, MNH1, ONH2, and WIH1. There is also much variation in the highs, lows, and average temperatures across the entire season (Figure 11) but we all see a lot of different variation daily about this overall trend (Figure 12). Some growing locations such as GAH1, the temperature does not vary greatly across the day away from the trend (approximate  $6^{\circ}C$  variation). This is different from the daily variations in a location like TXH4 which has a much larger variation of about  $12^{\circ}C$ .

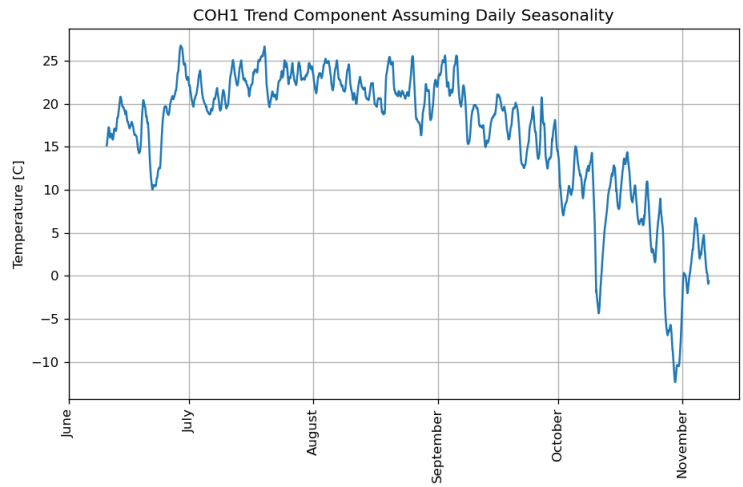


Figure 8: Trend component for location COH1 assuming daily seasonality

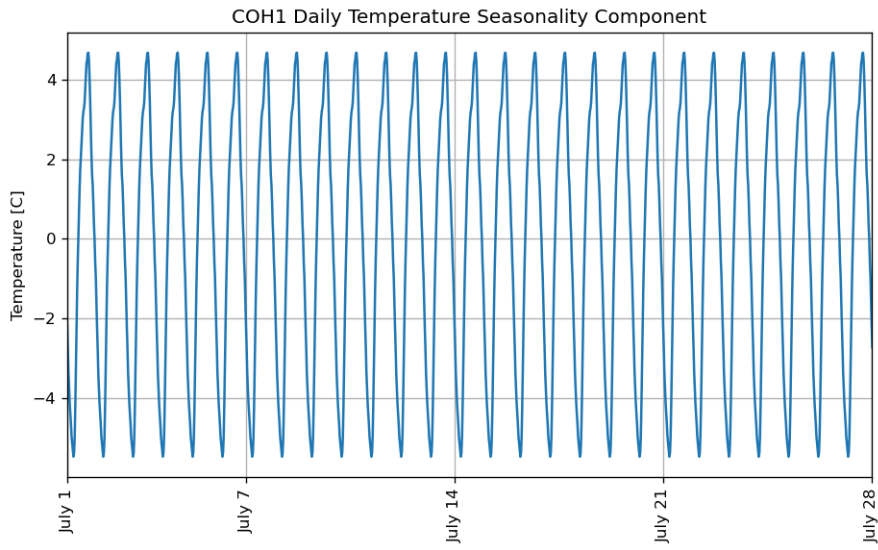


Figure 9: Seasonality component for location COH1 assuming daily seasonality



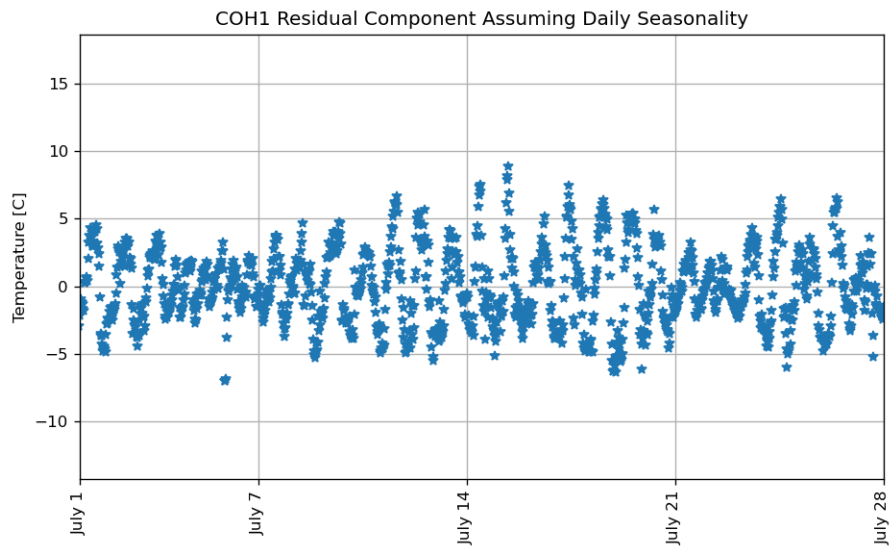


Figure 10: Residual component for location COH1 assuming daily seasonality

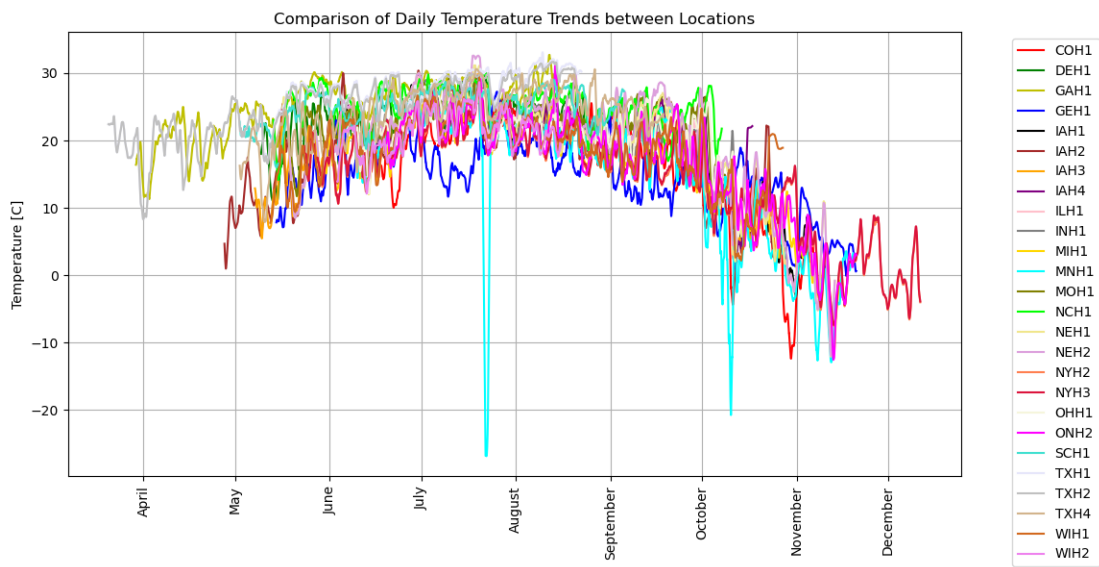


Figure 11: Plot of the trend component for all growing locations assuming daily seasonality

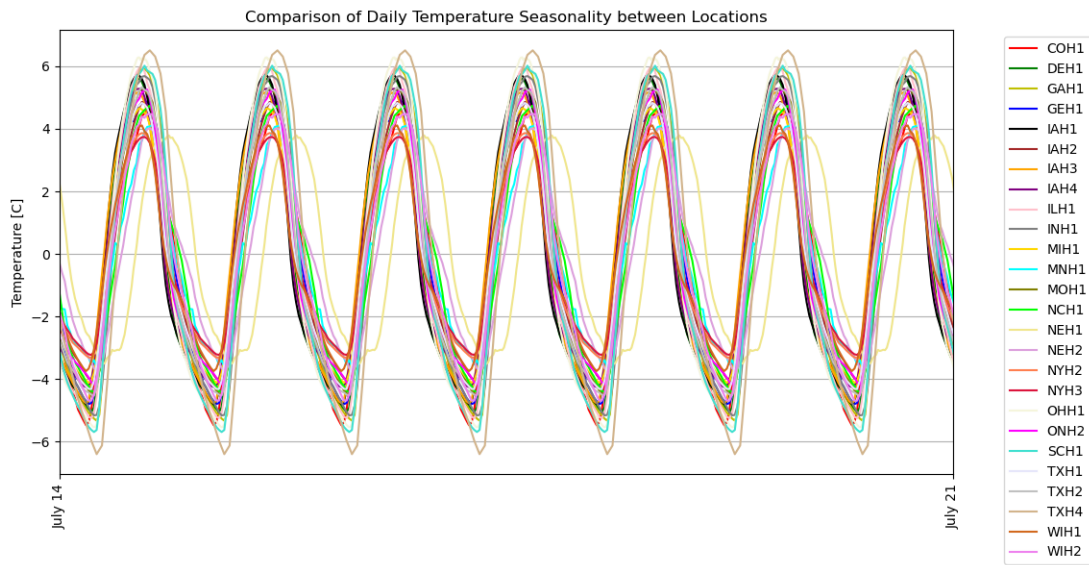


Figure 12: Seasonality component for all growing locations assuming daily seasonality

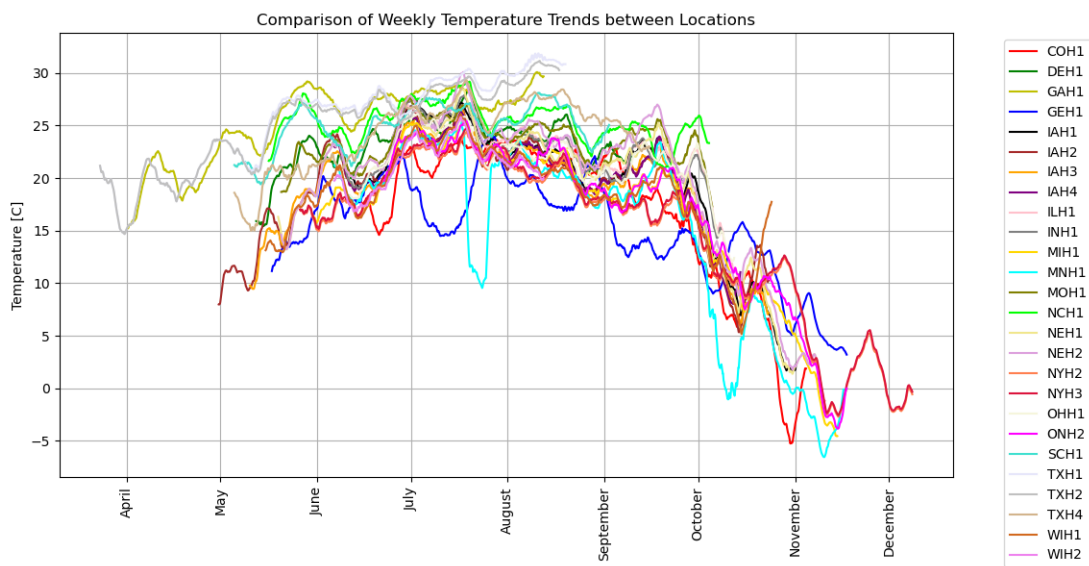


Figure 13: Trend component for all growing locations assuming weekly seasonality

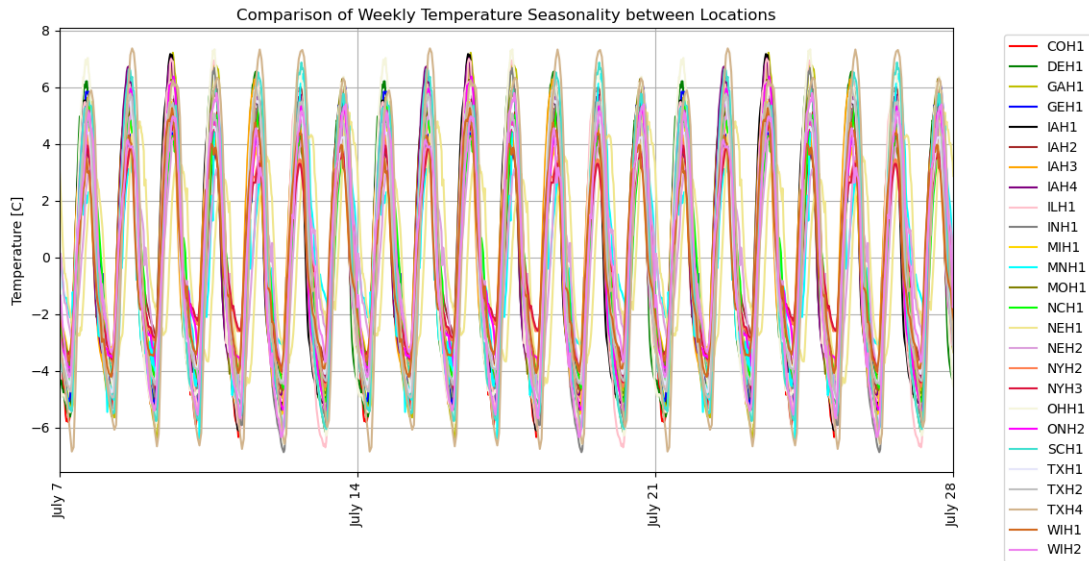


Figure 14: Seasonality component for all growing locations assuming weekly seasonality

On the trend plot assuming a weekly seasonality, (Figure 13), we see a somewhat smoother plot than that from Figure 11. When considering a longer periodicity, more of the fluctuation is shifted to the seasonality component as shown in Figure 14. We note that in future work we would like to consider times series decomposition in which we consider the presence of two different seasonal components, i.e.

$$Y = T + S_1 + S_2 + R$$

Nonetheless, with the smoother trend in Figure 13 we make some additional observations.

In Figure 13, one can more readily notice that locations like GAH1 and TXH2 have similar patterns but they appear to be offset from each other by some couple of days or weeks. GAH1 appears to have a phase shift to the right (i.e., offset to the

right of TXH2), but, if shifted to the left, they might be more in line with the other. But we need a way to numerically evaluate or measure how similar or dissimilar these growing seasons and locations are from each other and this brings us to the need of using dynamic time warping (DTW) to numerically evaluate the distance between two time series.

DTW is preferred over Euclidean distance in this research because of the following reasons. First, Euclidean distance is not used because the growing seasons are of different lengths. If one were to only consider the portion of the growing season in common between all growing locations, then we would only be able to consider the months between June and August, and, for one or two locations, we would need to disregard some daily measurements. Some have 48 measurements of temperature during the day while others had 72 or 96 total daily measurements. There would be equal lengths of growing seasons among locations with these restrictions but a lot of information would be lost. Hence the need for a different distance metric that will allow time series of different lengths. Another reason for not using Euclidean distance is because Euclidean distance considers differences at the same time points. Therefore, the value for the euclidean distance between GAH1 and TXH2 will be greater than when using a distance metric such as DTW which accounts for the fact that two time series might just be offset from each other. In the next section we describe the dynamic time warping distance metric in detail, showing how it accounts for both time series of different lengths as well as ones which might be more similar if aligning the series in a more ‘optimal’ way.

## 4 Dynamic Time Warping

As discussed, in order to cluster growing locations together, we need a way to systematically compare two temperature profiles such as that given in figure 11. Note that the start, end and length of a growing season will vary greatly across the locations; therefore we need a method that can compare time series data of varying lengths that might not sync up temporally. Dynamic Time Warping (DTW) is a way to compare two -usually temporal- sequences that do not sync up perfectly. It is a method to calculate the optimal matching between two sequences. DTW is useful in many domains such as speech recognition, data mining, financial markets, etc. [1]

We illustrate dynamic time warping by considering two series with varying lengths,  $A = [a_1, \dots, a_n]$  with length  $n$  and  $B = [b_1, \dots, b_m]$  with length  $m$ . Let  $\delta$  denote a base distance measurement between elements or coordinates of the sequences [25]. In this thesis we consider a difference between components of a series defined by  $\delta(b_i, a_j) = |b_i - a_j|^2$ . Other metrics can also be used such as  $\delta(b_i, a_j) = |b_i - a_j|$ . Note that if  $m = n$ , then the Euclidean distance between series  $A$  and  $B$ , which is commonly accepted as the simplest distance between sequences, is defined as

$$\|B - A\|_2 = \sqrt{\delta(b_1, a_1) + \dots + \delta(b_i, a_i) + \dots + \delta(b_n, a_n)}$$

where  $\delta(b_i, a_i) = |b_i - a_i|^2$  and one only considers corresponding components in a series. When two sequences don't match exactly or  $m \neq n$ , then an alternative method can be used to find an optimal alignment between series  $A$  and  $B$ .

This optimal alignment can be determined by first considering an alternative distance measurement (an accumulated distance) between *elements* of the sequences

$A$  and  $B$  computed by

$$D(b_i, a_j) = \delta(b_i, a_j) + \min\{D(b_{i-1}, a_{j-1}), D(b_{i-1}, a_j), D(b_i, a_{j-1})\} [25] \quad (1)$$

for  $2 < i < m$  and  $2 < j < n$ .

We then form an  $m$ -by- $n$  grid of values, called an accumulated cost matrix, where each point  $(i, j)$  in the grid is given by  $D(b_i, a_j)$  corresponding to elements  $b_i$ ,  $1 \leq i \leq m$  and  $a_j$ ,  $1 \leq j \leq n$ . Using the distance values in the grid, one forms a warping path  $W$  which maps the elements of  $A$  to  $B$  to minimize the distance between them.

The warping path is found using a dynamic programming approach to align two sequences. Going through all possible paths is “combinatorially explosive” [5]. Therefore, for efficiency purposes, it’s important to limit the number of possible warping paths, and hence the following constraints are outlined:

- **Boundary Condition:** This constraint ensures that the warping path begins with the starting points of both signals and terminates with their endpoints. In other words,  $W = \{(b_1, a_1), \dots, (b_m, a_n)\}$  where  $a_1$  is always paired with  $b_1$  (first components of each sequence) and  $a_n$  is always paired with  $b_m$  (last components of each sequence).
- **Monotonicity condition:** This constraint preserves the time-order of points (not going back in time).
- **Continuity (step size) condition:** This constraint limits the path transitions to adjacent points in time (not jumping in time). An acceptable warping path has combinations of the following acceptable moves (all forward in time)

- Horizontal moves:  $(i, j) \rightarrow (i, j + 1)$
- Vertical moves:  $(i, j) \rightarrow (i + 1, j)$
- Diagonal moves:  $(i, j) \rightarrow (i + 1, j + 1)$

In addition to the above three constraints, there are other less frequent conditions for an allowable warping path such as warping window conditions and slope condition. We refer the reader to references [28, 31] for full details on all the possible constraints.

#### 4.1 A toy example on DTW

We illustrate the concepts of dynamic time warping on a simple example, following the example given in [2]. Let's consider two sequences:  $A = [3, 1, 2, 2, 1]$  with length  $n = 5$  and  $B = [2, 0, 0, 3, 3, 1, 0]$  with length  $m = 7$ . Our aim is to find the best possible alignment and to calculate the DTW distance between the two sequences [24].

We start by meeting the demands of the boundary conditions, which state that the first and last points must line up, implying that the warping path must start with  $(a_1, b_1)$  and end with  $(a_n, b_m)$ . For this example, we start the warping path with the pairing  $(a_1, b_1) = (3, 2)$ . We then use an accumulated cost matrix (a 7x5 grid) to determine the next move. Recall that the allowed moves along a path are

- horizontal moves,  $(i, j) \rightarrow (i, j + 1)$ ,
- vertical moves,  $(j, i) \rightarrow (i + 1, j)$ , and
- diagonal moves,  $(i, j) \rightarrow (i + 1, j + 1)$ ;

therefore, for interior moves, the accumulated cost is given by Equation (1) from before, i.e.

$$D(b_i, a_j) = \delta(b_i, a_j) + \min\{D(b_{i-1}, a_{j-1}), D(b_{i-1}, a_j), D(b_i, a_{j-1})\}$$

where  $\delta(b_i, a_j) = |b_i - a_j|^2$ , and  $D(b_{i-1}, a_{j-1})$  is the cost from having used a diagonal movement to get to current point,  $D(b_i, a_{j-1})$  is the cost from using a vertical movement, and  $D(b_{i-1}, a_j)$  is the cost from using a horizontal movement.

We can best illustrate this accumulated cost matrix where the components of  $B$  are arranged in reversed order on the vertical axis and the components of  $A$  are arranged on the horizontal axis as shown in Table 2. In this format, the initial alignment, i.e.  $(a_1, b_1) = (3, 2)$  is in the bottom left-hand corner. Since this is the first alignment, the accumulated cost is simply calculated as

$$D(2, 3) = \delta(b_1 = 2, a_1 = 3) = |2 - 3|^2 = 1$$

Table 2: DTW distance matrix

0					
1					
3					
3					
0					
0					
2					
	3	1	2	2	1

Filling in this value to our accumulated cost matrix we have the grid in Table 3.



Table 3: DTW distance matrix

0					
1					
3					
3					
0					
0					
2	1				
	3	1	2	2	1

Now consider the other entries on the bottom row starting with  $a_2 = 1$  and  $b_1 = 2$ . Since there is only a left component filled in, we could only have horizontal movement. Therefore, the accumulated cost is given by

$$D(b_1 = 2, a_2 = 1) = \delta(b_1 = 2, a_2 = 1) + D(b_1, a_1) = |2 - 1|^2 + 1 = 2$$

Similarly, for all entries on the bottom row, the only move allowed was a horizontal movement thus

$$D(b_1, a_j) = \delta(b_1, a_j) + D(b_1, a_{j-1}) \quad \text{for } j = 3, 4, 5$$

Therefore, we have

$$D(b_1 = 2, a_3 = 2) = |2 - 2|^2 + D(b_1, a_2) = 0 + 2 = 2$$

$$D(b_1 = 2, a_4 = 2) = |2 - 2|^2 + D(b_1, a_3) = 0 + 2 = 2$$

$$D(b_1 = 2, a_5 = 1) = |2 - 1|^2 + D(b_1, a_4) = 1 + 2 = 3$$

obtaining the bottom row given in Table 4

Table 4: DTW distance matrix

$b_1 = 2$	1	2	2	2	3
a	3	1	2	2	1

We have a similar situation for the first column of the accumulated cost matrix.

The only movement is a vertical move, thus

$$D(b_i, a_1) = \delta(b_i, a_1) + D(b_{i-1}, a_1)$$

for  $i = 2, \dots, 7$ . For example, for  $D(b_2 = 0, a_1 = 3)$  we have:

$$D(0, 3) = \delta(b_2 = 0, a_1 = 3) + D(b_1, a_1) = |0 - 3|^2 + 1 = 10$$

Continuing this formulation, the first column entries are calculated as

$$D(b_2 = 0, a_1 = 3) = |0 - 3|^2 + D(b_1, a_1) = 9 + 10 = 19$$

$$D(b_4 = 3, a_1 = 3) = |3 - 3|^2 + D(b_3, a_1) = 0 + 19 = 19$$

$$D(b_5 = 3, a_1 = 3) = |3 - 3|^2 + D(b_4, a_1) = 0 + 19 = 19$$

$$D(b_6 = 1, a_1 = 3) = |1 - 3|^2 + D(b_5, a_1) = 4 + 9 = 23$$

$$D(b_7 = 0, a_1 = 3) = |0 - 3|^2 + D(b_6, a_1) = 9 + 23 = 32$$

giving the accumulated matrix in Table 5.

Once the boundaries are filled, the interior points can be calculated using the formula in equation (1). For example, for the interior point  $(b_2, a_2)$  we have all possible moves thus

$$D(b_2, a_2) = \delta(b_2, a_2) + \min\{D(b_1, a_1), D(b_1, a_2), D(b_2, a_1)\}$$

Table 5: DTW distance matrix

0	32				
1	23				
3	19				
3	19				
0	19				
0	10				
2	1	2	2	2	3
	3	1	2	2	1

or

$$\begin{aligned}
 D(0, 1) &= \delta(0, 1) + \min\{D(b_1, a_1), D(b_2, a_1), D(b_1, a_2)\} \\
 &= |0 - 1|^2 + \min\{10, 1, 2\} \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

Similarly for the  $a_2 = 1, b_3 = 0$  entry we have,

$$\begin{aligned}
 D(b_3 = 0, a_2 = 1) &= \delta(b_3 = 0, a_2 = 1) + \min\{D(b_2, a_1), D(b_2, a_1), D(b_2, a_2)\} \\
 &= |0 - 1|^2 + \min\{10, 19, 2\} \\
 &= 1 + 2 \\
 &= 3
 \end{aligned}$$

Repeating the same procedure we form the cost matrix given in Figure 15 which is plotted as a heat map for illustrative purposes using the `sbm.heatmap` Python command. We refer the reader to [24] for more details.

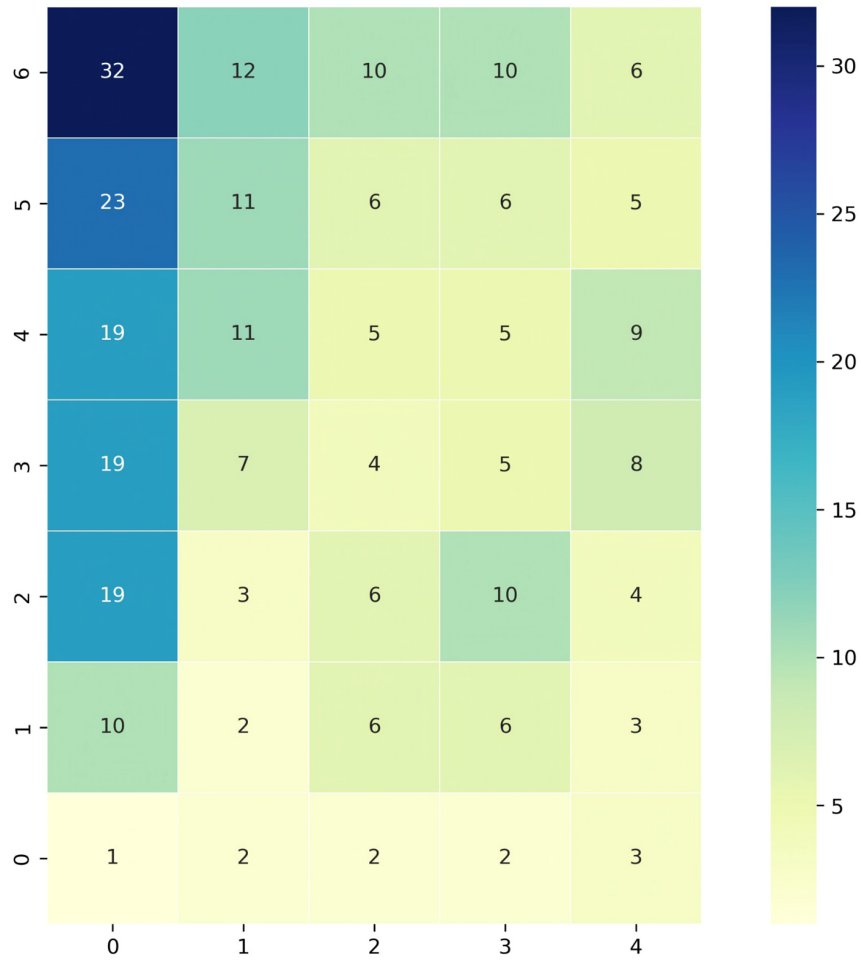


Figure 15: Heat map of accumulated cost matrix for toy example [24]

To obtain the optimal alignment for the warped path we start in the lower left corner and move according to lowest cost with the most effective movements as illustrated in Figure 16 which corresponds to an optimal alignment given as

$$X = [3, 1, 1, 2, 2, 1, 1]$$

$$Y = [2, 0, 0, 3, 3, 1, 0]$$

as plotted in Figure 17. Note that the path moves forward in time and maps high

points in one sequence with high points in the other sequence and similarly for low points. To calculate the final DTW distance, we use the Euclidean distance formula on the best alignment, given by

$$\sqrt{(3 - 2)^2 + (1 - 0)^2 + (1 - 0)^2 + (2 - 3)^2 + (2 - 3)^2 + (1 - 1)^2 + (1 - 0)^2} = 2.45$$

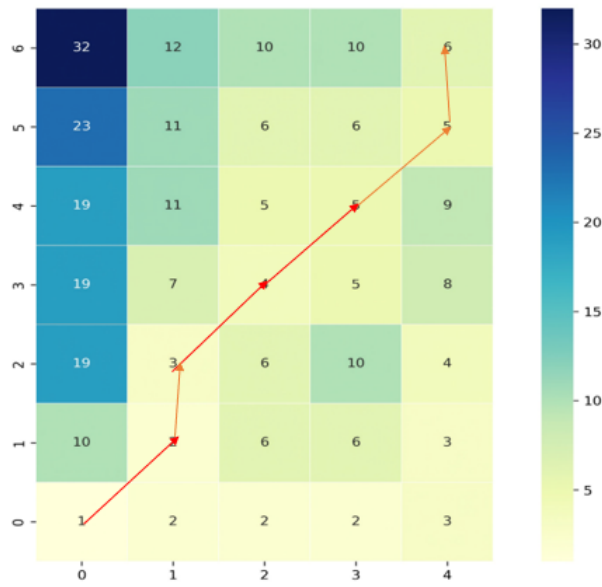


Figure 16: Heat map of accumulated cost matrix for toy example [24]

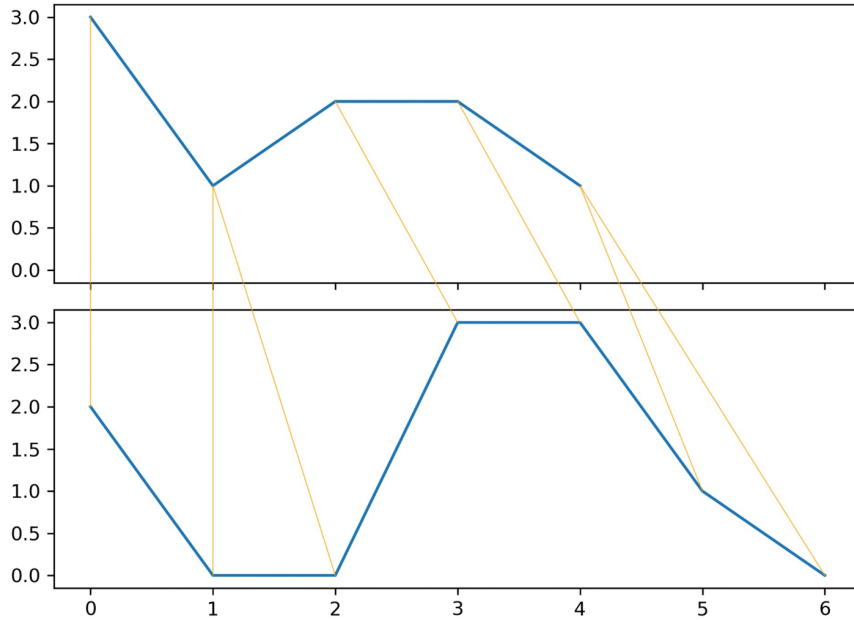


Figure 17: Alignment Graph for warped path on simple example

#### 4.2 The advantages of DTW over Euclidean distance for time series data

For time series data, we again emphasize that Dynamic Time Warping (DTW) has a number of important benefits over Euclidean distance [24]. First, by enabling non-linear alignments between two time series, DTW offers flexibility in matching. When comparing time series with various lengths or when patterns exhibit temporal shifts, its ability to accept differences in speed or phase shifts is especially helpful.

The insensitivity of DTW to scale is another benefit [19]. DTW is resistant to variations in the time series' amplitude or magnitude, unlike Euclidean distance. Instead of emphasizing absolute values, it concentrates on portraying how similar the form or pattern is. Since the cumulative distance matrix is computed via the dynamic

programming approach, the best alignment is chosen. Therefore it has the capacity to detect comparable patterns despite shifts or distortions. Applying dynamic time warping on the trend data assuming a daily seasonality period (as explained in section 3), we obtain the distance between successive growing points given in Table 6 and Table 7 for daily trend component data (as plotted in Figure 11) and daily seasonality component (as plotted in Figure 12) data respectively.

Table 6: Distance Matrix For Daily Trend With The Labels 1 Through 26 Representing The 26 Different Growing Locations Considered In This Study: 1:COH1, 2:DEH1, 3:GAH1, 4:GEH1, 5:IAH1, 6:IAH2, 7:IAH3, 8:IAH4, 9:ILH1, 10:INH1, 11:MIH1, 12:MNH1, 13:MOH1, 14:NCH1, 15:NEH1, 16:NEH2, 17:NYH2, 18:NYH3, 19:OHH1, 20:ONH2, 21:SCH1, 22:TXH1, 23:TXH2, 24:TXH4, 25:WIH1, 26:WIH2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	0																										
2	608	0																									
3	982	139	0																								
4	201	547	989	0																							
5	185	468	796	270	0																						
6	268	401	562	189	324	0																					
7	238	383	662	158	238	118	0																				
8	289	240	442	303	151	336	260	0																			
9	266	273	459	265	131	327	231	167	0																		
10	447	131	315	394	282	355	294	136	137	0																	
11	131	656	1038	198	172	246	218	308	285	464	0																
12	359	885	1269	417	400	444	437	473	480	711	367	0															
13	470	137	251	394	319	311	261	188	153	88	498	745	0														
14	604	109	178	550	445	403	372	249	209	117	639	906	106	0													
15	213	724	1138	305	199	313	268	337	321	511	178	382	539	721	0												
16	184	582	923	253	180	260	218	313	254	411	165	366	423	567	60	0											
17	149	803	1259	208	202	267	234	284	287	578	136	351	627	812	184	187	0										
18	153	791	1242	280	197	265	228	280	281	569	139	353	616	800	181	187	11	0									
19	292	258	525	272	143	294	204	135	95	114	298	507	152	232	318	264	324	317	0								
20	143	639	1027	214	136	298	248	251	243	445	108	370	486	628	188	179	151	152	271	0							
21	784	110	133	780	617	561	545	336	342	191	836	1085	192	109	934	748	1036	1021	376	824	0						
22	1119	197	77	1108	920	651	790	505	554	389	1174	1392	320	215	1274	1055	1399	1383	633	1163	180	0					
23	1071	168	79	1066	889	621	754	465	528	364	1126	1347	297	198	1228	1016	1342	1327	597	1119	158	20	0				
24	731	83	104	698	571	421	453	305	296	187	779	1022	149	115	883	693	985	971	320	773	104	161	154	0			
25	344	296	526	298	278	214	224	172	252	234	429	520	254	330	452	379	322	313	208	368	449	602	561	351	0		
26	598	101	243	531	452	370	344	246	207	132	635	895	128	120	730	574	813	802	221	631	129	344	312	101	275	0	

Table 7: Distance Matrix For Daily Seasonality With The Labels 1 Through 26 Representing The 26 Different Growing Locations Considered In This Study: 1:COH1, 2:DEH1, 3:GAH1, 4:GEH1, 5:IAH1, 6:IAH2, 7:IAH3, 8:IAH4, 9:ILH1, 10:INH1, 11:MIH1, 12:MNH1, 13:MOH1, 14:NCH1, 15:NEH1, 16:NEH2, 17:NYH2, 18:NYH3, 19:OHH1, 20:ONH2, 21:SCH1, 22:TXH1, 23:TXH2, 24:TXH4, 25:WIH1, 26:WIH2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	0																									
2	10	0																								
3	14	6	0																							
4	8	9	12	0																						
5	12	5	7	10	0																					
6	13	12	16	9	11	0																				
7	8	6	9	7	7	8	0																			
8	12	8	12	8	7	5	5	0																		
9	13	6	6	12	6	15	8	11	0																	
10	10	5	6	10	5	13	7	9	6	0																
11	12	13	17	8	12	4	7	7	16	14	0															
12	17	21	26	14	20	8	15	13	25	22	7	0														
13	10	7	10	7	5	6	5	5	9	7	7	14	0													
14	9	12	15	7	12	5	8	8	14	12	5	8	7	0												
15	22	30	33	21	30	17	25	23	33	30	17	10	22	14	0											
16	17	23	27	14	22	10	17	15	26	23	9	4	15	9	7	0										
17	21	26	31	18	24	11	19	17	249	27	10	4	17	13	13	6	0									
18	23	28	33	20	26	13	21	18	31	29	12	5	19	14	12	6	3	0								
19	24	11	8	22	11	29	17	21	7	9	30	44	19	27	54	46	52	55	0							
20	12	9	13	9	8	4	6	4	12	10	6	12	5	7	21	13	15	17	23	0						
21	15	8	6	15	10	20	12	15	6	7	20	30	13	18	37	31	35	37	7	16	0					
22	10	10	14	6	9	6	6	6	13	11	4	9	6	5	18	11	13	15	26	6	17	0				
23	10	7	10	7	5	7	5	5	9	7	7	15	3	7	23	16	19	21	18	6	12	5	0			
24	19	15	12	19	15	23	18	20	12	14	23	30	18	22	36	30	33	33	15	20	11	21	18	0		
25	16	20	25	13	19	8	14	12	23	21	6	4	13	8	13	5	4	5	43	11	29	8	14	29	0	
26	7	13	16	6	13	7	8	9	15	13	6	10	9	5	16	10	14	15	29	8	19	5	9	22	9	0

Notice that the distance matrix table for the daily trend data (Table 6) ranges from 20 to 1399 where the smallest value in the matrix indicates the most similarity between growing location TXH1 and growing location TXH2. The most dissimilar locations are NYH2 and TXH1 with a distance value of 1399. For the daily seasonality component (Table 7), the most similar locations are MOH1 and TXH2 and also NYH2 and NYH3 with a distance value of 3. While the most dissimilar growing locations are NYH3 and OHH1 with a distance value of 55. Assuming a daily seasonality



component, the most similar growing locations are still NYH2 and NYH3; whereas the most dissimilar (greatest distance) is no longer NYH2 and TXH1. Therefore, when performing clustering based on the dynamic time warping distance, we expect that clusters formed using the trend component might differ from those formed when considering the seasonal component.

## 5 Clustering

Clustering is a type of unsupervised machine learning [12]. It is referred to as “unsupervised”, because we are not guided by a prior idea of which features or samples belong in which clusters. It is a type of “learning”, because the machine algorithm “learns” how to cluster the data. Another name for this set of techniques is “pattern recognition” [6]. Clustering is used for many different purposes such as pattern recognition, image processing and information retrieval [18]. For the purpose of this thesis, we want to cluster growing locations according to their climate (as evidenced by temperature profiles across time), so one might be able to better predict which underlying genomic features most relate to crop yield given a similar climate profile. We used two types of clustering technique, hierarchical clustering and Kmeans clustering together with the dynamic time warping metric for comparing time series trend and seasonality profiles (as discussed in section 3) across various growing locations.

### 5.1 Hierarchical Clustering

Hierarchical clustering is a methods that recursively clusters two items at a time [22]. There are two different types of hierarchical clustering, agglomerative and partitioning. In partitioning algorithms, the entire set of items start in a single cluster which is partitioned into two more homogeneous clusters. The algorithm then restarts with each of the new clusters, partitioning each into more homogeneous clusters until each cluster contains only identical items (possibly only 1 item).

In agglomerative algorithms, each item starts in its own cluster and the two most similar items are then clustered. It continues accumulating the most similar items or

clusters together two at a time until there is one cluster. For both types of algorithms, the clusters at each step can be displayed in a dendrogram. In this thesis we consider an agglomerative hierarchical method. According to [12], the agglomerative process can be summarized by the following steps when clustering  $N$  items.

1. Choose a distance function
2. Start with  $N$  clusters, each containing one item. Then, at each iteration:
  - using the current matrix of cluster distances, find the two closest clusters.
  - update the list of clusters by merging the two closest.
  - update the matrix of cluster distances accordingly
3. Repeat until all items are joined in a single cluster.

We illustrate this process assuming a simple example [2] with five series labeled 1 through 5. We assume each series is a time series with a dynamic time warping (DTW) distance between series  $i$  and  $j$  given as the  $(i, j)$  entry of the distance matrix given in Table 8. We note this is a symmetric matrix with diagonal entries equal to 0 (the distance between a series and itself is 0); therefore only the lower triangle is displayed.

Table 8: Original DTW Distance Matrix for Example

$$D_1 = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & & & & \\ 2 & 9 & 0 & & & \\ 3 & 3 & 7 & 0 & & \\ 4 & 6 & 5 & 9 & 0 & \\ 5 & 11 & 10 & 2 & 8 & 0 \end{array}$$

Given the distance matrix, we locate the smallest entry which has the value 2 in the (5,3) entry, indicating the distance between series 3 and series 5. Therefore series 3 and series 5 are clustered together first. Then a new distance matrix  $D_2$  is created (see Table 9) where the separate series 3 and 5 are replaced with the 3-5 cluster. The distance between the 3-5 cluster and the remaining series is found by finding the maximum distance between series 3 or 5 and the other series. That means the distance between 1 and [3-5] is the maximum distance between series 1 and series 3 and series 1 and series 5 which can be expressed as

$$\begin{aligned}
 D_2(1, [3-5]) &= \max\{D_1(1, 3), D_1(1, 5)\} \\
 &= \max\{3, 11\} \\
 &= 11
 \end{aligned}$$

as shown in Table 9.

Table 9: Updated DTW Distance Matrix for Example

$$D_2 = \begin{array}{c|cccc} & [3-5] & 1 & 2 & 4 \\ \hline [3-5] & 0 & & & \\ 1 & 11 & 0 & & \\ 2 & 10 & 9 & 0 & \\ 4 & 9 & 6 & 5 & 0 \end{array}$$

Likewise,

$$\begin{aligned}
D_2(2, [3-5]) &= \max\{D_1(2, 3), D_1(2, 5)\} \\
&= \max\{7, 10\} \\
&= 10
\end{aligned}$$

and

$$\begin{aligned}
D_2(4, [3-5]) &= \max\{D_1(4, 3), D_1(4, 5)\} \\
&= \max\{9, 8\} \\
&= 9
\end{aligned}$$

Clustering continues with the new matrix  $D_2$  by determining the least entry in  $D_2$  which is 5, the distance between series 2 and series 4. Therefore series 2 and series 4 are now clustered together and an updated matrix  $D_3$  is given by

Table 10: Second Updated DTW Distance Matrix for Example

		[3-5]	[2-4]	1
$D_3$	[3-5]	0		
	[2-4]	10	0	
	1	11	9	0

Clustering continues and then can be visualized in a dendrogram as in Figure 18.

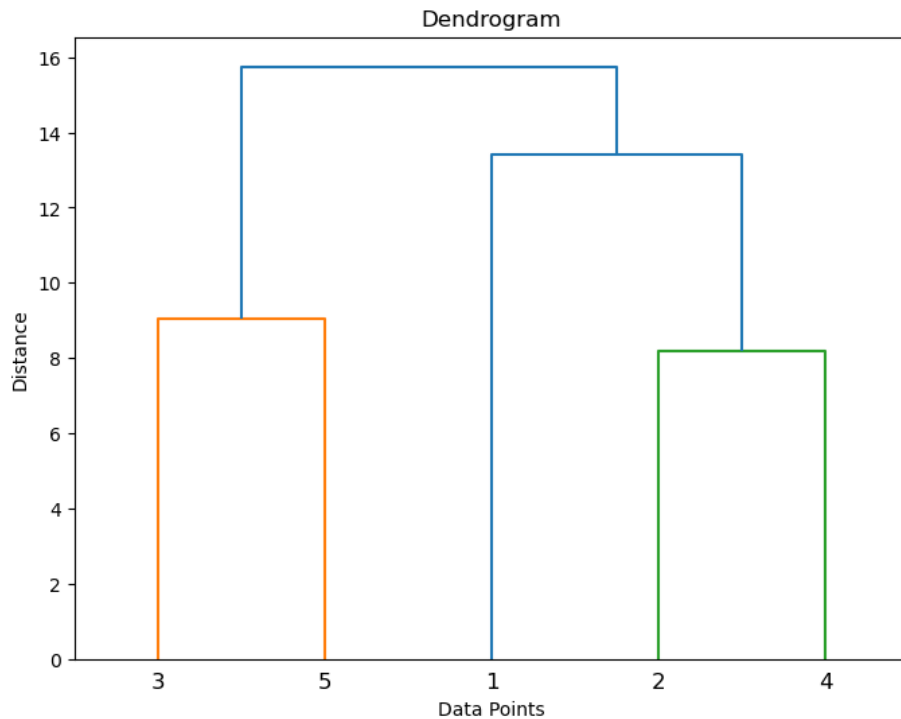


Figure 18: Dendrogram of hierarchical clustering of simple example

## 5.2 K-means Clustering

An alternative clustering technique is K-means clustering [21]. K-means clustering is an iterative algorithm that aims to partition data points into  $K$  clusters, where  $K$  is a pre-defined number. We describe K-means clustering assuming spatial points and then will explain the difference when considering time series. For spatial points, the algorithm starts by randomly selecting  $K$  initial cluster centroids and assigns each data point to the nearest centroid based on a distance metric, typically Euclidean distance when considering spatial points. It then updates the centroids by computing the mean of the data points assigned to each cluster. This process iterates until

convergence, where the centroids no longer change significantly or a maximum number of iterations is reached.

K-means clustering is a method that is frequently used to group geographic locations, but with minor modifications, it can also be used to group time series data. The fundamental idea behind K-means is still to categorize data points according to how far they are from centroids, but there are significant differences in how to apply K-means for time series data.

For time series, instead of centroids, we consider barycenters. A barycenter is similar to the centroid in spatial clustering; however calculating a barycenter also involves an iterative procedure, as explained in [25]. The K-means procedure applied to time series is given by the following process [25, 33]

1. Given a stated value of  $K$  (such as  $K = 3$ ), choose  $K$  time series as the beginning average or barycenter for each cluster.
2. Determine the DTW (dynamic time warping) distance between each time series to be clustered and each of the  $K = 3$  averages or barycenters.
3. Based on the DTW distance, assign each time series to the cluster with the closest barycenter.
4. Calculate the new barycenter of each of the  $K$  clusters again using a dynamic time-warping calculation of the new average. Referring to [25] we use the DTW Barycenter Averaging procedure which is an iterative process where;
  - It finds the DTW paths between series and the approximate center.

- It finds the approximate center by taking the weighted average of all the connected points.
  - It repeats the process again until there is no change in that center or until it gets below the tolerance level where it no longer have much movements in the barycenter average.
5. Re-organize the new time series and update the dynamic time warping distances between all of the time series and the revised centers.
  6. Keep iterating steps 2-5 until a certain tolerance is attained.

In conclusion, geographical point clustering and K-means clustering for time series data are both clustering techniques. Due to the lack of spatial points, centroids must be chosen and updated differently. Instead, dynamic time warping and barycenter concepts are used to calculate averages. This modification of the K-means clustering technique, designed specifically for time series data, enables efficient clustering and grouping of time series based on their similarities.

Both hierarchical clustering and K-means clustering have their strengths and weaknesses, and their suitability for time series data depends on various factors, such as data characteristics, clustering objectives, and interpretability requirements. In the following section, we will implement and compare these clustering approaches for time series data created by using time series decomposition techniques.



## 6 Results

In this section we consider several different scenarios for clustering growing locations based on temperature profiles across the growing season. Since both daily and weekly seasonality were present in the data (as shown in Table 1), we first cluster based on the trend of the temperature in the different growing locations across the entire growing season where the overall trend is determined as one component of the time series decomposition. Recall from Section 3 that this trend is determined when the appropriate seasonality (daily or weekly) is assumed and both the seasonal component and residual error are subtracted from the data. As shown in [3, 13, 14], the growth rate of crops is strongly dependent on temperature, as extreme temperatures can greatly affect plant productivity. For example, the range at which maize can grow is  $10^{\circ}C$  to  $38^{\circ}C$  with a maximum growth response around  $26^{\circ}C$  to  $30^{\circ}C$  [13]. Therefore, crops which have more growing days within the ideal range might result in a greater yield.

On the other hand, Sunoj et. al. [34] showed that diurnal temperature amplitude or variation, i.e. the temperature fluctuation in a single daily period, can also have an important effect on crop yield. For example, they showed that lower diurnal temperature amplitude negatively impacts maize growth. Therefore, when genetically engineering crops, certain genetic factors may be more or less important based on the daily fluctuation in temperature. As such, the daily seasonality component from the time series decomposition may also play an important role in the growth response of crops and might prove to be a better temperature profile on which to cluster growing locations when trying to predict which genetic factors might lead to higher crop yield.

Recall, the ultimate goal is to use the resulting clusters to control for climate and then within each cluster, determine which genetic factors in the maize might lead to the greatest crop yield. Therefore, clustering based on seasonality may provide a better control for climate in some circumstances. As such, we compare the clusters found when considering the trend across the entire growing season, i.e. the overall growth and decline in temperature across several months, versus the daily seasonality component in which we only consider the seasonal aspect of the temperature profile as the main commonality between growing locations. We do this using both hierarchical and K-means clustering algorithms as discussed in Section 5 and then compare whether the groups are similar or dissimilar when using the overall growing season temperature profile or the daily seasonality profile to cluster.

### 6.1 Clustering Based on Trend Assuming a Daily Period

Recall with hierarchical clustering, we systematically update the distance matrix in Table 6 to obtain the dendrogram in Figure 19. To determine the number of clusters from the dendrogram, we follow a step-by-step process. First, we need to select a threshold value on the vertical axis of the dendrogram. This threshold distance plays a crucial role in defining the number of clusters. This value should be carefully chosen to ensure that it cuts the dendrogram at an appropriate level, effectively separating the distinct clusters. Essentially, the threshold value represents the maximum vertical distance at which we are willing to merge clusters. Once the threshold distance is determined, we proceed to plot a horizontal line at this chosen threshold distance on the dendrogram plot (as shown in Figure 19). This line serves as a reference point

for identifying the clusters. The next step involves counting the number of vertical lines, or branches, that intersect this horizontal line. Each intersected line represents a cluster. By counting the intersected lines, we can determine the number of clusters. It is important to note that the choice of the threshold distance and the resulting number of clusters can be subjective and may require careful consideration of the specific problem domain or prior knowledge.

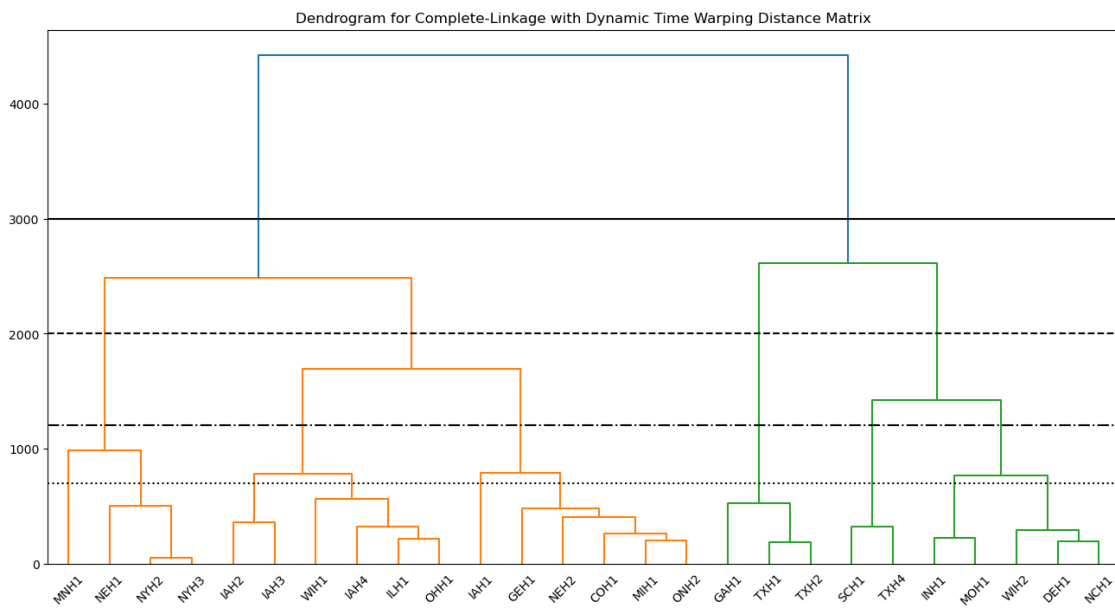


Figure 19: Denodogram using hierarchical clustering on the trend component of the time series decomposition when assuming a daily seasonality period.

In Figure 19, we have plotted four horizontal lines which correspond to threshold values of 3000, 2000, 1200, and 700 resulting in 2, 4, 6 and 10 clusters respectively. We refer the reader back to the distance matrix given in Table 6, where we noted that the most similar locations were TXH1 and TXH2 which you see grouped in the dendrogram. Assuming a threshold of 1200 with 6 clusters, for each cluster, we plot

the associated trends from the time series decomposition assuming a daily periodicity in Figure 20. We first notice that the clusters appear to group based on the length of the growing season; however, the total length does vary from location to location.

However, across all clusters, we can visually see why the various locations group together, with similar trends across the growing season. Another important note based on the dendrogram in Figure 19 is that different growing locations within each state do not necessarily cluster together. For example, TXH1 and TXH2, two different growing locations in Texas always cluster together; however, in Iowa, IAH1 only clusters with IAH2, IAH3 and IAH4 when considering a threshold level of about 1800 or higher. This is the same with the two growing locations in Nebraska, NEH1 and NEH2, which only cluster together at a threshold level of about 2500 or higher.

In general, we can make several other observations when analyzing the trend across a growing season assuming daily seasonality. First, the locations TXH1 and TXH2 consistently appear together in the same cluster as does NYH2 and NYH3 and IAH2 and IAH3 except for extremely small threshold values. When considering a smaller threshold of 700 or below, MNH1 and IAH1 form their own individual cluster without being grouped with any other locations, indicating these two growing locations are most dissimilar from other growing locations with MNH1 being the first growing season to form its own cluster at a threshold value of slightly less than 1000. Finally, growing seasons for maize are shortest in Texas, Georgia, and South Carolina, the southern-most states in Figure 1, and split across two clusters in Figure 20.

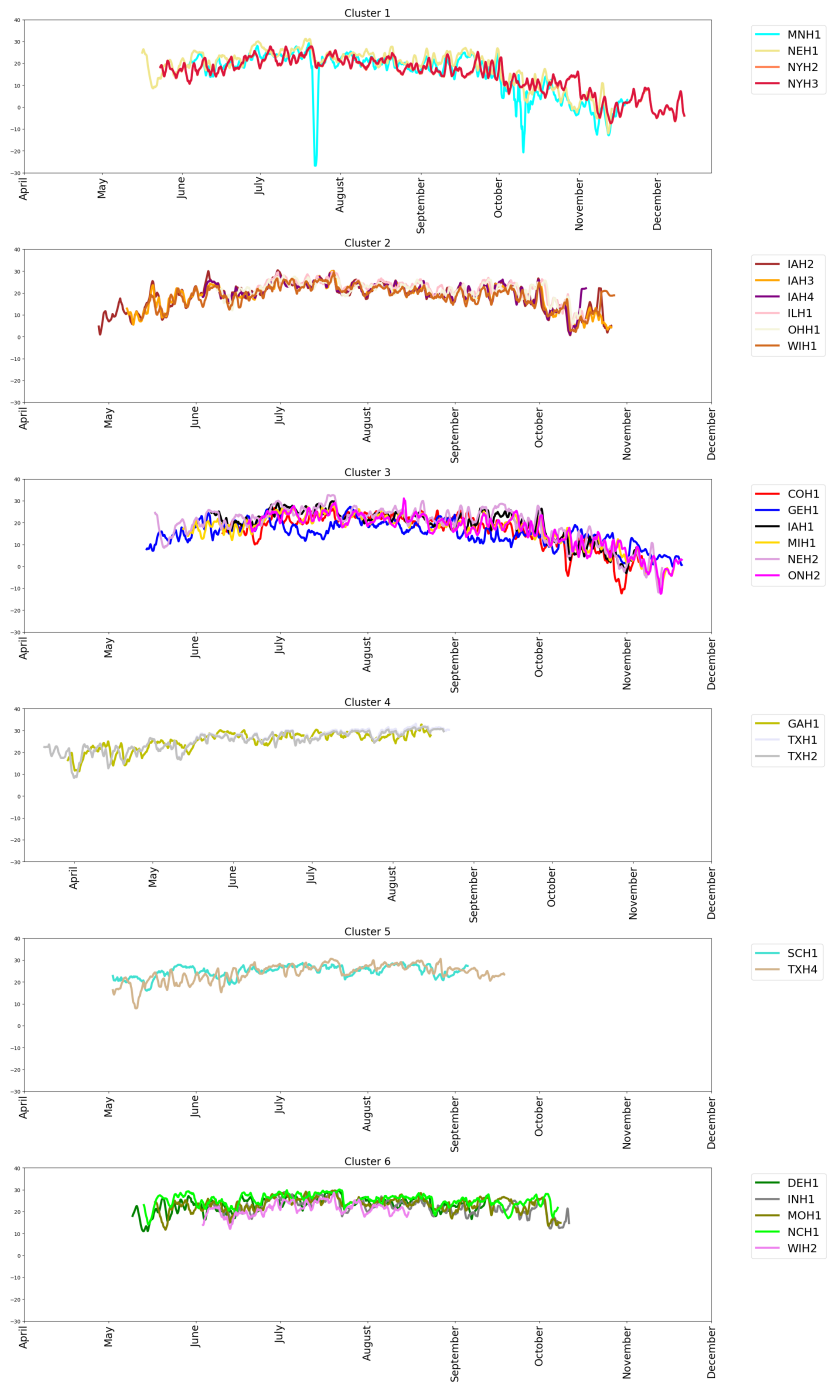


Figure 20: Clusters based on a threshold level of 1200 in the hierarchical clustering algorithm on the trend component of the time series decomposition assuming a daily period for seasonality

## 6.2 K-Means Clustering

Recall that in the K-means clustering algorithm, it is necessary to start with the total number of clusters ( $K$ ) first, and then the locations are grouped in a systematic manner by comparing the time series to the  $K$  barycenters. The time series are then grouped in the cluster where the distance (dynamic time warping distance) between the evaluated time series and the barycenter of the chosen cluster is smallest. A new barycenter is then determined for each of the  $K$  clusters and the procedure is repeated again. This continues until the maximum number of iterations are reached or the change in the clusters is below a given tolerance level. To implement, we used the python command `sklearn.cluster.K-means` in the `scikit.learn` library [29] with the default number of iterations of 50 and default tolerance of  $1e-6$ . In Figure 21, we illustrate the clusters when assuming  $K=6$  clusters.

In comparing the results to the hierarchical clustering when choosing a threshold level resulting in 6 groups (Figure 20), we notice some similarities in the clusters and some notable differences as expected based on the methodology. Similar to the results in hierarchical clustering, TXH1 is still grouped with TXH2 as is NYH2 with NYH3 and IAH2 with IAH3. We note that, unlike in the hierarchical clustering, IAH4 is no longer grouped with IAH2 and IAH3. Both IAH1 and IAH4 are in clusters which do not include other Iowa locations. Another difference can be seen when comparing the southeast growing locations. Even though the shortest growing seasons were in Georgia, South Carolina and Texas, in the hierarchical clustering algorithm, these locations were split across two clusters with  $K=6$  clusters. In the K-means clustering algorithm, they are now all grouped together. More differences can also be seen

with the pairings; however, one other notable difference is in the cluster containing only WIH1 in Figure 21. We note that the first single cluster from the hierarchical clustering would have been MNH1 at a threshold level around 1000; however, using  $K=6$  clusters with the maximum iterations of 50 and default tolerance of  $1e-6$ , WIH1 is clustered independently, indicating it is most dissimilar from the averages of the other growing locations.

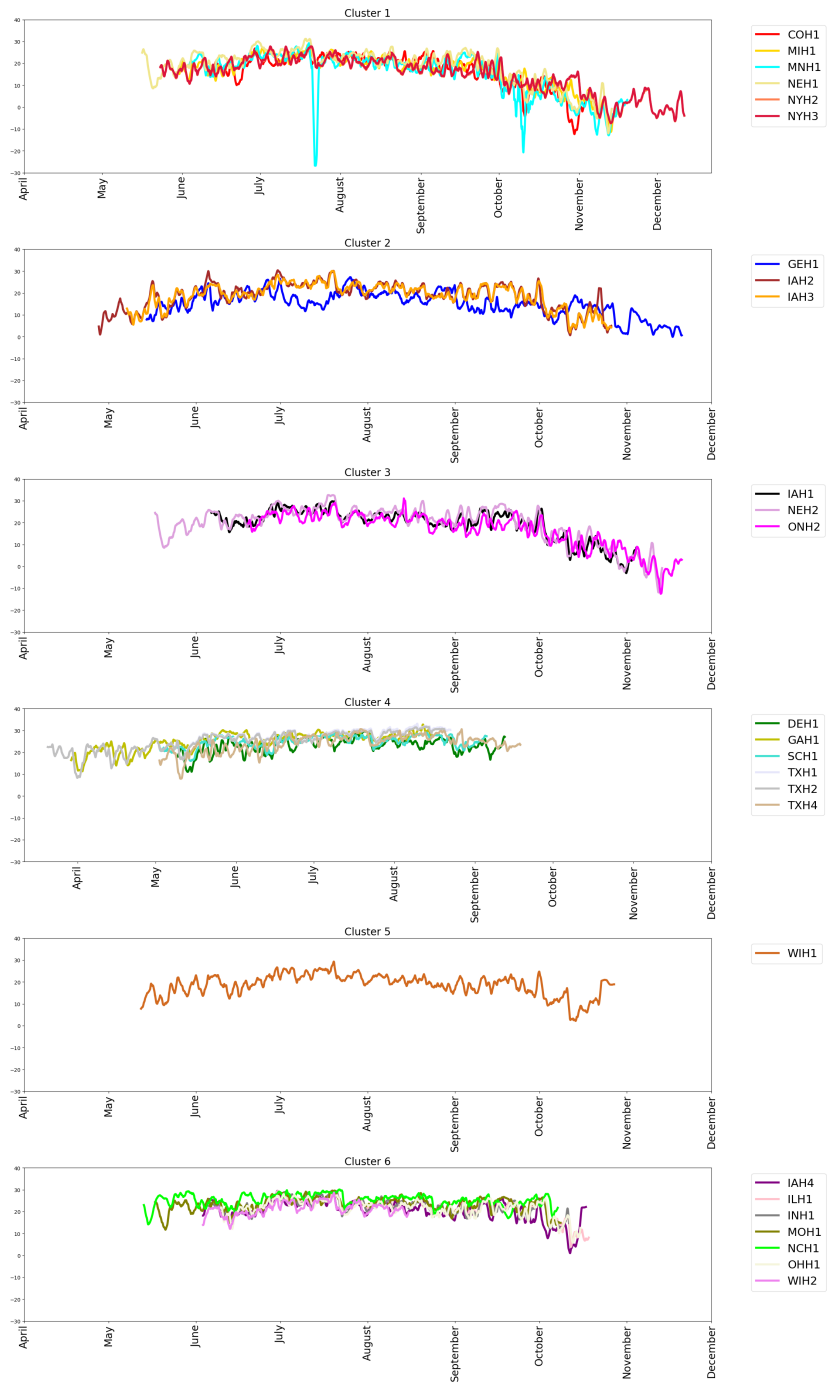


Figure 21: Clusters based using  $K=6$  clusters in the K-means clustering algorithm on the trend component of the time series decomposition assuming a daily period for seasonality



In addition to Figure 21 for  $K=6$  clusters, we include the results for  $K = 3, 5$  and  $9$  in tabular form in Tables 11-13 respectively. In general, we can make several other observations when analyzing the trend across a growing season assuming daily seasonality. First, the locations TXH1 and TXH2 consistently appear together in the same cluster as does NYH2 and NYH3 and IAH2 and IAH3 when using either hierarchical or kmeans clustering. When  $K=9$  with Kmeans clustering, MNH1 and WIH1 form their own individual clusters with WIH1 being the first growing location to form its own cluster. This differs from hierarchical clustering slightly in which MNH1 was the first to cluster individually at a threshold level of around 1000 followed by IAH1 and then WIH1 at threshold levels around 800 and 500 respectively. Other similarities can be found in the southern states in which GAH1 clusters with TXH1 and TXH2 while SCH1 clusters with TXH4 using either technique. We also notice a slight change when considering a small number of clusters, i.e. a threshold value of 3000 in hierarchical clustering (forming 2 clusters) and when using  $K=3$  in K-means clustering. We note that with the large group with hierarchical clustering (see Figure 19), INH1 and MOH1 are grouped with the locations found in cluster 2 of Table 11; whereas they are grouped differently when using Kmeans clustering. We also note that cluster 5 in Table 12 (Kmeans) is similar to the leftmost grouping of Figure 19 (hierarchical) but with the addition of NEH2. If one further examines these results we can continue to find other similarities with slight differences.

In this study, we distinguished between two popularly known clustering techniques which are K-means and Hierarchical clustering to gain more insight into the distinctions in cluster formation. It is well established fact that different clustering

algorithms can yield to distinct clustering patterns due to their unique approaches when executed as we saw in the analysis above[33]. This comparative analysis helped us gain a comprehensive understanding of the data’s inherent clustering tendencies and emphasize the influence of different algorithms on clustering outcomes. Though K-means clustering is a popular approach, its extensive iterative process requires a substantial amount of computational time and resources. Hence for the remaining research in this thesis, we shall focus exclusively on those results generated from Hierarchical clustering, while noting that the observations may be slightly different if using K-means clustering or a different clustering algorithm.

Table 11: Clusters using  $K=3$  in the K-means Clustering Algorithm

C1	C2	C3
GEH1		
IAH1	DEH1	COH1
IAH2	GAH1	MIH1
IAH3	NCH1	MNH1
IAH4	SCH1	NEH1
ILH1	TXH1	NEH2
INH1	TXH2	NYH2
MOH1	TXH4	NYH3
OHH1	WIH2	ONH2
WIH1		

Table 12: Clustering using  $K=5$  in the K-means Clustering Algorithm

C1	C2	C3	C4	C5
DEH1 IAH4 ILH1 INH1 MOH1 NCH1 OHH1 WIH2	COH1 IAH1 MIH1 ONH1	GAH1 SCH1 TXH1 TXH2 TXH4	GEH1 IAH2 IAH3 WIH1	MNH1 NEH1 NEH2 NYH2 NYH3

Table 13: Clustering using  $K=9$  in the K-means Clustering Algorithm

C1	C2	C3	C4	C5	C6	C7	C8	C9
GAH1 TXH1 TXH2	NYH2 NHY3	IAH4 ILH1 INH1 MOH1 NCH1 OHH1	GEH1 IAH2 IAH3	MNH1	DEH1 SCH1 TXH4 WIH2	NEH1 NEH2	COH1 IAH1 MIH1 ONH2	WIH1

### 6.3 Clustering Based on Trend Assuming a Weekly Period

In section 3 we noted that the trend component when using a weekly seasonality period (Figure 13) provided a greater degree of smoothing in the data when compared to using the daily seasonality period (Figure 11); therefore we examine the clusters formed when using this periodicity and compare it to the results in section 6.1.

In Figure 22, we have the resulting dendrogram with three horizontal lines which correspond to threshold values of 2550, 1900, and 1200 resulting in 3, 5, and 6 clusters respectively. In Figure 23, for each cluster, we plot the associated trends from the

time series decomposition assuming a weekly periodicity and a threshold of 1200 (i.e. 6 clusters). Note that with 6 clusters, the clusters assuming a weekly period have some similarities but also some noticeable differences from those when assuming a daily period (Figure 20). For example, in both cases GAH1, TXH1 and TXH2 form their own cluster; however, when considering daily periodicity and examining the trend SCH1 and TXH4 also formed their own cluster. On the contrary, when considering weekly periodicity, these locations are now grouped with DEH1, NCH1 and WIH2. The latter three locations were previously grouped with INH1 and MOH1 which now form their own cluster. Furthermore, clusters 2 and 3 from Figure 20 are now grouped as one cluster in Figure 23 with the exception of IAH1 and NEH2 which are now grouped with IAH2, IAH3 and WIH1. We also note that the overall shape of the dendrograms (Figures 19 and 22) are fundamentally different. In Figure 19, when we assumed a daily periodicity, the dendrogram branched (from top down) from a cluster of 2 to a cluster of 4; whereas in Figure 22, it branched from a cluster of 2 to a cluster of 3 instead. There is no one “right” way to group growing locations, so the clustering process might need to be considered in conjunction with the predictive modelling to determine what type of genetic properties best work in “similar” growing locations where “similar” needs to be determined concurrently. We now examine the seasonality component instead of the trend component.

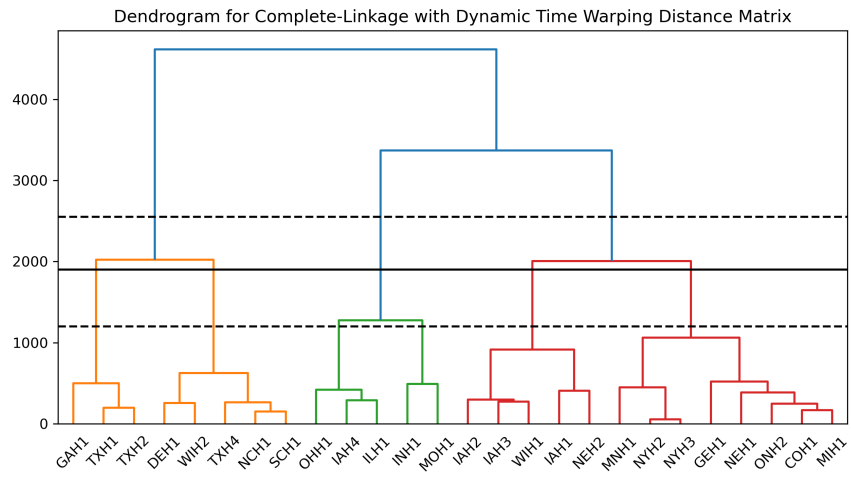


Figure 22: Dendrogram using hierarchical clustering on the trend component of the time series decomposition when assuming a weekly seasonality period

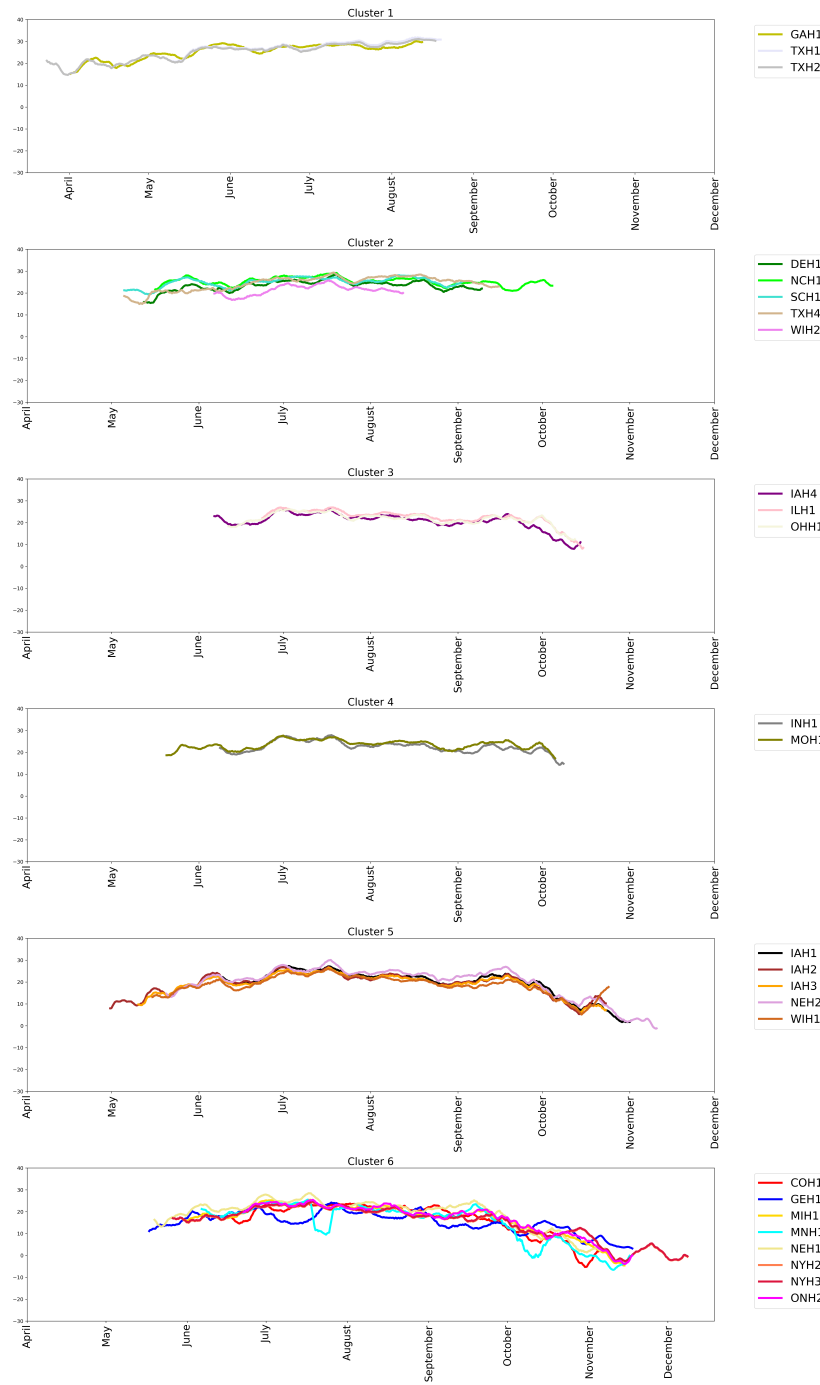


Figure 23: Clusters based on a threshold level of 1200 in the hierarchical clustering algorithm on the trend component of the time series decomposition assuming a weekly period for seasonality

## 6.4 Clustering Based on the Seasonality Component Assuming Daily Seasonality

As discussed in the introduction to this section, diurnal temperature can have a great impact on the productivity of plant growth; therefore, in this section we analyze the clusters formed when using the seasonality component as opposed to the trend component of the time series decomposition when assuming daily seasonality only. We then compare and contrast the resulting clusters with those found in section 6.1.1 where we used the trend component of the time series decomposition assuming a daily seasonality. For this section, we assumed only a two-week time period which overlapped between all growing seasons, consisting of 14 daily periods total as the daily seasonality component does not differ greatly across the season. Similar to the previous section, we first analyze the resulting dendrogram using hierarchical clustering. Figure 24 gives the dendrogram resulting from using dynamic time warping on the daily seasonality component with three threshold lines at values 90, 60 and 32 resulting in 3, 4 and 8 clusters respectively. In other words, this dendrogram considers grouping growing locations based more on the similarity of daily variation about the trend as opposed to the overall trend of the profile across the entire growing season. We note that the branching behaviour is more similar to Figure 22 (dendrogram for trend component assuming a weekly seasonality) in that it branches from 2 branches to 3 but the growing of locations vary from the analysis based upon trend when assuming either periodicity.

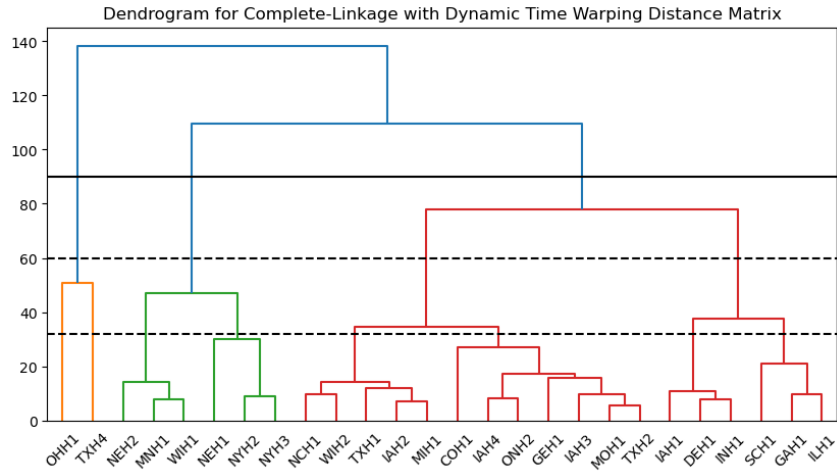


Figure 24: Dendrogram using hierarchical clustering on the seasonal component of the time series decomposition when assuming a daily seasonality period

In Figure 25, we plot the seasonal components for each of the growing locations in the four clusters formed when assuming a threshold level of 60. We noticed that all the growing locations display a consistent pattern in the range of variation of the daily temperature, indicating that they experience the same fall and rise in temperature throughout the growing season. The locations NEH1, MNH1 and WIH1 consistently appear together in the same cluster as does NYH2 and NYH3 except for extremely small threshold values, indicating that these locations share similar range of variation of the daily temperature seasonality. When considering a smaller threshold of around 55 or below, OHH1 and TXH4 form their own individual cluster without being grouped with any other locations, indicating that these two growing locations are most dissimilar from other growing locations. We also noticed that the range of variation is different across all the different clusters. The ones with large range



of variation away from the trend are grouped together versus the smaller ones. For example at the threshold value of 60 where  $K=4$  clusters, locations like OHH1 and TXH4 are clustered together with the largest range of variation in daily temperature seasonality (over  $12^{\circ}C$ ) as oppose locations like MNH1, NEH1, NEH2, NYH2, NYH3 and WIH1 which are clustered together with the smallest range in variation in daily temperature seasonality (approximately  $8^{\circ}C$ ). Finally based on Sunoj et al [34] maize productivity might be more improved in locations like OHH1 and TXH4, considering the large range of variation in daily temperature, but only if the overall average temperature is in the optimal range [13] which still requires analyzing the overall trend in Figure 20.

Comparing the trend component assuming daily seasonality to the seasonal component assuming daily seasonality, what stands out as a factor of similarity between them is the consistent clustering of the growing locations NYH2 and NYH3 which shows a strong association between the two locations. However, previously, when examining the trend component assuming daily or weekly seasonality, locations like TXH1 and TXH2 were always clustered together except at small threshold values. When examining the seasonal component, TXH1 and TXH2 are not immediately clustered together; they are only together if moving higher in the branching process. The same is true for IAH2 and IAH3.

Previously, we also noticed that the southern-most locations in Texas, Georgia and South Carolina always grouped together at smaller branches in the dendrogram (Figures 19 and 22) using hierarchical clustering or smaller  $K$  values in K-means clustering (Tables 11-13 and Figure 21) if grouping based on the trend across the growing season. However, TXH4 isn't grouped with these other locations except for when not splitting the locations (i.e 1 group). The other locations still don't group together until moving all the way up the dendrogram to a threshold above 80. Therefore, the seasonality component contains different information that might be important when clustering growing locations. It might be useful in future studies to incorporate both components, ideally with other climate measures as well, to cluster growing locations

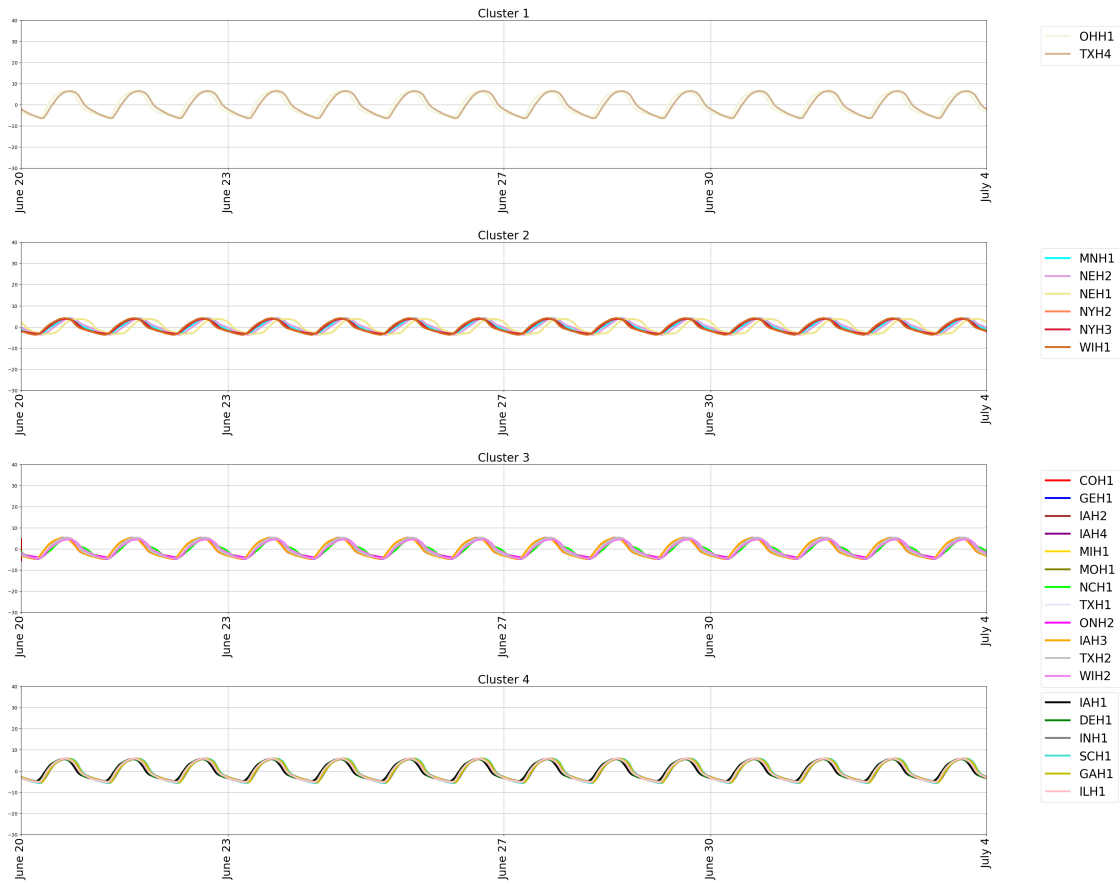


Figure 25: Clusters based on a threshold level of 60 in the hierarchical clustering algorithm on the seasonal component of the time series decomposition assuming a daily period for seasonality

## 7 Summary and Future Work

In this study, the overall goal was to develop a proof of concept for the use of clustering techniques previously implemented by Taylor et al [35], to better control for climate prior to trying to predict maize crop yield in a specific climate setting. It uses K-means clustering and hierarchical clustering, two well-known clustering techniques, a dynamic time warping distance metric, and time series decomposition. In time series data that describes climate variables like temperature, precipitation, humidity, and wind patterns, the goal is to find similarities. Even though the growing season for each time series data varies, the main goal is to compare the clustering results between hierarchical clustering and K-means clustering approaches, highlighting the differences of each method and how these differences resulted in different groupings considering the same set of time series. We then examined the similarities and differences in the resulting clusters when assuming different seasonality periods in the data while comparing the overall trend in the data. Finally, given the importance of the range in diurnal temperature, we also examined clusters when isolating just the seasonal component. Thus, in summary, we showed how time series decomposition can be leveraged in different ways to cluster time series data.

Having successfully implemented and compared hierarchical clustering and K-means clustering approaches to cluster time series data using the idea of time series decomposition couple with dynamic time warping distance metric, there are several recommendations one would consider for future work to further improve the predictive modeling of maize crop yield. First, it would be beneficial to examine clustering locations using *both* the trend *and* seasonality components simultaneously. In this

research, the idea of time series decomposition was used to extract separately the trend component and the seasonality component of a time series data for clustering analysis. So future work might consider clustering based on both the trend and seasonality at the same time in order to capture more comprehensive patterns and interactions within the data.

Furthermore, in our analysis, for the sake of identifying patterns, a single seasonality period was assumed. Therefore, future research could explore time series decomposition techniques that consider multiple seasonality periods at the same time, giving room for more flexibility and accuracy of modeling seasonal patterns in maize crop yield. For example, for most of our growing locations, there was a strong autocorrelation for both daily and weekly seasonality, so we could consider breaking our time series into a trend component, daily seasonality component, weekly seasonality component and the residual. Even though our primary focus was on temperature data, other climate factors, such as humidity, soil moisture, precipitation, or wind, might also play an important role in controlling for climate. So, in future work, one can incorporate these other climate factors into the clustering analysis process to identify clusters based on a comprehensive set of available variables, providing enough information for better understanding of the factors influencing maize crop yield.

Finally, once the clustering is implemented, one can perform predictive modeling within each clusters. This can help enhance targeted predictive models for maize crop yield after controlling for the effects of weathers conditions. By exploring these future recommendations, the accuracy and interpretability can be further improved for maize crop yield.

## BIBLIOGRAPHY

- [1] Alizadeh Hamidi, B., Khosravi, F., Hosseini, S. A., & Hassannejad, R. (2022). Closed form solution for dynamic analysis of rectangular nanorod based on non-local strain gradient. *Waves in Random and Complex Media*, 32(5), 2067-2083.
- [2] Alizadeh, E. (2020, October 11). An Illustrative Introduction to Dynamic Time Warping. Ealizadeh.com. <https://ealizadeh.com/blog/introduction-to-dynamic-time-warping/>
- [3] Backlund, P., Janetos, A. C., & Schimel, D. S. (2008). *The effects of climate change on agriculture, land resources, water resources, and biodiversity in the United States* (Vol. 4). US Climate Change Science Program.
- [4] Beddington, J. R., Asaduzzaman, M., Clark, M. E., Bremauntz, A. F., Guillou, M. D., Jahn, M. M., ... & Wakhungu, J. (2012). The role for scientists in tackling food insecurity and climate change. *Agriculture & Food Security*, 1(1), 1-9.
- [5] Berndt, D. J., & Clifford, J. (1994, July). Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining* (pp. 359-370).
- [6] Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.
- [7] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

- [8] Brockwell, P. J., Davis, R. A., Brockwell, P. J., & Davis, R. A. (2016). Nonstationary and seasonal time series models. *Introduction to time series and forecasting*, 157-193.
- [9] Brownlee, J. (2017). How to decompose time series data into trend and seasonality. *Machinelearningmastery.com*, Jan.
- [10] Cano, A. (2018). A survey on graphic processing unit computing for large-scale data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(1), e1232.
- [11] Chaurasia, V., & Pal, S. (2020). Application of machine learning time series analysis for prediction COVID-19 pandemic. *Research on Biomedical Engineering*, 1-13.
- [12] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. *The elements of statistical learning: Data mining, inference, and prediction*, 485-585.
- [13] Hatfield, J. L., & Prueger, J. H. (2015). Temperature extremes: Effect on plant growth and development. *Weather and climate extremes*, 10, 4-10.
- [14] Hatfield, J. L., Boote, K. J., Kimball, B. A., Ziska, L. H., Izaurralde, R. C., Ort, D., ... & Wolfe, D. (2011). Climate impacts on agriculture: implications for crop production. *Agronomy journal*, 103(2), 351-370.

- [15] Hertel, T. W., Baldos, U. L. C., Hertel, T. W., & Baldos, U. L. C. (2016). Climate change impacts in agriculture. *Global Change and the Challenges of Sustainably Feeding a Growing Planet*, 69-84.
- [16] Huang, G. (2021, February). Missing data filling method based on linear interpolation and lightgbm. In *Journal of Physics: Conference Series* (Vol. 1754, No. 1, p. 012187). IOP Publishing.
- [17] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- [18] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3), 264-323.
- [19] Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and information systems*, 7, 358-386.
- [20] Lesk, C. S. (2022). *New insights on how changing hydroclimate might affect crop yields—and a way to avoid the worst of it*. Columbia University.
- [21] MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281-297).
- [22] Milligan, G. W., & Cooper, M. C. (1988). A study of standardization of variables in cluster analysis. *Journal of classification*, 5, 181-204.
- [23] Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. ” O’Reilly Media, Inc.”.



- [24] Müller, M. (2007). Dynamic time warping. *Information retrieval for music and motion*, 69-84.
- [25] Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition*, 44(3), 678-693.
- [26] Pham, H. T. L. (2023). Examining the Local-Scale Relationship between *Human Mobility* and COVID-19: A Case Study of San Diego, CA (Doctoral dissertation, San Diego State University).
- [27] Pierre, S. (2022, October 12). A Guide to Time Series Analysis in Python. Builtin.com. <https://builtin.com/data-science/time-series-python>
- [28] Ratanamahatana, C. A., & Keogh, E. (2004, August). Everything you know about dynamic time warping is wrong. In *Third workshop on mining temporal and sequential data* (Vol. 32). Citeseer.
- [29] Raschka, S. (2015). *Python machine learning*. Packt publishing ltd.
- [30] RB, C. (1990). STL: A seasonal-trend decomposition procedure based on loess. *J Off Stat*, 6, 3-73.
- [31] Salvador, S., & Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5), 561-580.
- [32] Shumway, R. H., Stoffer, D. S., Shumway, R. H., & Stoffer, D. S. (2017). Additional time domain topics. *Time Series Analysis and Its Applications: With R Examples*, 241-287

- [33] Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques.
- [34] Sunoj, V. J., Shroyer, K. J., Jagadish, S. K., & Prasad, P. V. (2016). Diurnal temperature amplitude alters physiological and growth response of maize (*Zea mays* L.) during the vegetative stage. *Environmental and Experimental Botany*, *130*, 113-121.
- [35] Taylor Dennis, K. W. (2019). *Determining Over-Performing Phenotypic Features in Maize Using Machine Learning Techniques* [Technical report, Department of Mathematics and Statistics, East Tennessee State University, Johnson City, TN.].
- [36] Taylor Dunlevy, H. S. P. W. (2019). *Identifying Environmentally-Influenced Clusters to Better Predict Maize Crop Yield* [Technical report, Department of Mathematics and Statistics, East Tennessee State University, Johnson City, TN.].
- [37] Wahba, G. (1990). *Spline models for observational data*. Society for industrial and applied mathematics.

VITA

EMMANUEL AIGBOKHAVBO OGEDEGBE

Education: B.S. Mathematics, Bayero State University in Kano,  
Kano, Nigeria 2014  
M.S. Mathematical Sciences, East Tennessee State  
University, Johnson City, Tennessee, December 2023

Professional Experience: Mathematics Teacher, Mount Carmel Secondary School,  
Emaudo-Ekpoma, Edo State, Nigeria, 2016-2020  
Graduate Assistant, East Tennessee State University  
Johnson City, Tennessee, 2021-2023