APPLICATION OF REMOTE SENSING AND MACHINE LEARNING IN VEGETATION PHENOLOGY AND CLIMATE CHANGE STUDIES

MASOOMA SULEMAN

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF ARTS

GRADUATE PROGRAM IN INFORMATION SYSTEMS AND TECHNOLOGY
YORK UNIVERSITY
TORONTO, ONTARIO

APRIL 2025

© MASOOMA SULEMAN, 2025

Abstract

Remote sensing and machine learning (ML) have revolutionized phenology studies by offering scalable and automated methods for monitoring vegetation growth patterns. Traditional phenology detection methods, which rely on field observations, are often labor-intensive and geographically constrained. This thesis introduces a novel deep learning model, the Temporal Multivariate Attention Network (TMANet), which integrates remote sensing data, climate indices, and ground observations to enhance phenological stage detection in crops. Focusing on corn phenology, the study explores how remote sensing data preprocessing optimizes its utility for phenology applications, how ML techniques improve detection accuracy, and how TMANet outperforms traditional models in capturing temporal and environmental dependencies. The proposed framework provides a robust, data-driven approach to understanding vegetation responses to climate variability, supporting sustainable agricultural management. The findings contribute to advancing phenology research by offering a scalable and efficient methodology for monitoring crop development and assessing climate change impacts on vegetation phenology.

Acknowledgements

I would like to express my deepest gratitude to Professor Peter Khaiter, whose unwavering support, encouragement, and expertise have been invaluable throughout my master's studies. His insightful guidance introduced me to the application of machine learning and data science in climate change research—a field that was novel to me. Under his mentorship, I not only completed this thesis but also grew as a researcher, contributing to several conference and research publications. I am also sincerely grateful to the members of my examination committee, Professor Arik Senderovich and Professor Tasso Adamopoulos, for their time, thoughtful feedback, and expertise, which have greatly contributed to refining this research. Finally, I extend my heartfelt thanks to my family and friends for their unwavering support, patience, and encouragement. Their belief in me has been instrumental in reaching this milestone, and I am truly grateful for their presence on this journey.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The U.S. Geological Survey (USGS, 2011) defines phenology as the study of recurring life-cycle events in plants and animals, focusing on how they respond to seasonal and interannual climate variations. This field plays a crucial role in understanding ecosystem responses to climate variability, benefiting sectors such as agriculture, wildlife management, and ecological conservation by predicting the impacts of climate change on biodiversity and food-production systems. Crop-specific phenology research, for example, has focused on monitoring corn growth stages, providing insights into how climate affects key stages like emergence, silking, and maturation, which are essential for understanding agricultural productivity and resilience (Yang et al., 2021).

Traditional phenology detection methods, which rely heavily on field observations, are often labor-intensive, time-consuming, and geographically limited (Matongera et al., 2021). The advent of remote sensing and machine learning (ML) has significantly enhanced phenology research, offering scalable data collection and advanced analysis. Remote sensing enables broad spatial and temporal monitoring of vegetation, allowing researchers to study phenological patterns across seasons and large areas (Weiss et al., 2020; Yang et al., 2021). For instance, satellite-derived indices, such as the Normalized Difference Vegetation Index (NDVI), have become integral for monitoring vegetation health and phenology, providing continuous data that reflect environmental conditions (Liu et al., 2019; Seo et al., 2019). However, remote sensing alone can be limited by factors like cloud cover and atmospheric variability, highlighting the importance of integrating it with meteorological data to enhance the accuracy of phenology detection models (Nieto et al., 2021).

Machine learning has further revolutionized phenology studies, offering powerful tools to process extensive datasets and identify complex environmental patterns. ML algorithms now automate phenological stage detection in crops, with the ability to integrate vegetation indices, temperature, precipitation, and ground-truth data, offering a robust framework for phenological monitoring under varying climate conditions (Ramadhani et al., 2020). Despite these advancements, existing phenology models, such as threshold-based and curvature-based approaches, often face challenges in generalizing across diverse climatic conditions and accurately capturing temporal dependencies (Diao C., 2020; Yang et al., 2021). This limitation underscores

the need for models capable of capturing the intricate temporal interactions between climatic and vegetative data.

To address these challenges, this thesis introduces a novel deep learning model, the Temporal Multivariate Attention Network (TMANet), which utilizes multivariate temporal data and applies attention mechanisms to enhance detection of phenological growth stages in crops. TMANet integrates remote sensing data (RSD), climate indices, and ground observations, enhancing the precision of phenological stage classification and contributing to our understanding of vegetation responses to climate change. Additionally, we design and implement a machine learning framework that incorporates essential modules for precise, data-driven, and efficient monitoring of crop growth stages. This comprehensive approach offers a scalable solution for sustainable agricultural management and climate adaptation strategies.

The thesis focuses on corn crop phenology for TMANet's experimental validation, as corn is a major crop in North America, especially in the U.S. and Canada. Its well-defined growth stages make it ideal for phenological studies, allowing the model to be rigorously tested under varied climate conditions. Moreover, this focus aligns with the importance of accurate monitoring for high stakes crops that significantly impact the regional agricultural economy.

The corresponding research contributions are as follows:

RC 1. A comprehensive stepwise guideline for collecting, extracting and transforming RSD tailored for phenology detection in agricultural contexts.

RC 2. A novel framework which will establish a robust dataset for model training and provide empirical validation of the integrated approach against traditional phenology models, showcasing improvements in accuracy and reliability.

RC 3. The development of an innovative model, TMANet, leveraging attention mechanisms to prioritize critical agronomic features as well as model complex temporal patterns. It will combine multiple data streams, including remote sensing indices, meteorological variables, and historical phenological data, to provide a holistic understanding of factors influencing corn development.

RC 4. A comprehensive assessment of how agro-climatic data can identify climate vulnerabilities, enhance predictive capabilities and support sustainable practices.

The key research questions of this study are presented in Table 1.

Table 1. Research questions and motivation

N	Research Question	Motivation
1	What are the necessary steps to preprocess RSD to optimize its quality for phenology detection in agricultural applications?	To ensure high-quality, reliable RSD inputs, which are essential for accurately detecting and predicting agricultural phenological stages under varying environmental conditions.
2	How can RSD and climate indices be integrated with machine learning to improve phenology detection accuracy in corn crops?	To enhance phenology detection accuracy by leveraging the combined strengths of RSD and climate indices through ML.
3	How does TMANet improve upon existing phenology models, especially in capturing complex temporal and environmental dependencies in corn growth stages?	To address current phenology model limitations by leveraging TMANet's ability to capture complex temporal and environmental interactions, thereby advancing the precision of corn growth stage detection in diverse climatic conditions.
4	How can RSD, climatic variables and phenological data contribute to understanding and managing climate change impacts on vegetation phenology?	Understanding these relationships can help mitigate risks associated with climate variability and enhance food security.

The rest of the paper is organized as follows. Chapter 2 presents concepts crucial for the project and examines existing research on phenology models, remote sensing, machine learning applications, and climate influences on vegetation, while identifying the research gaps that this thesis seeks to address. It also presents a bibliometric review of prior phenology work utilizing RSD. In Chapter 3, the thesis details the data sources, preprocessing methods, and experimental setup. This chapter describes the remote sensing data, relevant climate indices, and the proposed model, the Temporal Multivariate Attention Network (TMANet), designed for phenology

detection. Chapter 4 documents the results in terms of performance metrics, model comparisons, and climate vulnerability assessment. Lastly, Chapter 5 concludes the thesis by summarizing the work and discuses its limitations as well as proposed future research directions.

Together, these chapters provide a structured framework of moving from theoretical foundations to practical implications, which allows for a comprehensive understanding of the study's contributions to the fields of data science, information systems and technology being applied in the domain of sustainable agriculture.

Chapter 2: Literature Review

2.1: Vegetation Phenology and its Importance

Vegetation phenology refers to the study of recurring biological events and patterns in the life cycles of plants, specifically related to their growth, development, and seasonal changes (Zeng et al., 2020). It focuses on the timing and duration of various stages in plant life described using phenological parameters such as budburst, leaf emergence, flowering, fruiting, and senescence. Numerous ecological processes, as well as several biosphere-atmosphere feedback, are regulated by vegetation phenology. This chapter aims to provide a comprehensive understanding of vegetation phenology, highlighting its importance in agricultural and ecological studies, the role of climatic factors in influencing these biological events, and its broader significance in understanding ecological responses to a changing climate.

2.1.1: Phenology in Agricultural Systems

One of phenology's main contributions to agriculture is its role in improving crop management and yield prediction. Review studies on vegetation phenology conducted by Adole et al. (2016) and Zeng et al. (2020) have demonstrated its use in precision agriculture as farmers can determine optimal planting times, anticipate harvest periods, and gauge water and nutrient requirements, making it possible to tailor agricultural practices to the growth cycle of specific crops. Monitoring phenology also helps identify and predict potential yield losses due to adverse climatic events, such as unexpected frosts or droughts, which can severely impact crop health if they occur during critical growth phases like flowering or grain filling (Čehulić et al., 2019; Rubio-Cuadrado et al., 2021). Results from Yang et al (2021) suggested increased corn yield was significantly correlated with the delayed harvesting dates (1.37 Bushels/acre per day) and extended growing season length (1.67 Bushels/acre per day).

In addition to crop productivity, phenology plays an essential role in pest and disease management within agricultural systems (Wu et al., 2023). Many pests and diseases follow a seasonal cycle that aligns with crop phenology, making it possible to predict pest emergence and infestation risks based on crop growth stages (Thomson et al., 2010). In their study of application of crop phenology, Potgieter et al. (2021) highlight the critical role of temporal information in Australian grain cropping environment for optimizing agricultural practices, particularly disease and weed management and the staged application of nitrogen fertilizer. They emphasize that certain

chemical treatments are most effective at specific crop growth stages, while nitrogen applications in cereals are strategically timed between mid-tillering and flowering to enhance crop productivity and resilience.

2.1.2: Climatic Factors Influencing Vegetation Phenology

Since the 1950s, ecological researchers have tried to identify the relationship between vegetation phenology and climatic factors. Nonetheless, it has been not before the early 1990s when it was officially confirmed that vegetation phenology is highly dependent on climatic variables; their interlinking acting as a responsive indicator of seasonal shifts in climatic variables and their evident impacts on the environment (Adole et al., 2016). In particular, temperature was found as one of the most influential climatic factors affecting vegetation phenology. Numerous studies demonstrate that rising temperatures due to climate change have advanced the onset of phenological events, such as leafing, flowering, and fruiting (Butler et al., 2018; Cui et al., 2017; Piao et al., 2019). Butler et al. (2018) analyzed the impact of climate change on corn yields in the United States, finding that warmer temperatures have caused earlier planting and accelerated corn development, leading to significant shifts in phenological stages such as silking and maturity. While earlier planting can mitigate late-season drought risks, it also increases the risk of early-season frost and reduces the grain-filling period, which can potentially lower yields. Yang et al. (2021) found that shifts in planting and harvesting dates for corn and soybean in Kentucky have occurred over time, reflecting the influence of changing climate conditions. Specifically, corn planting was delayed by 0.01 days per year, while soybean planting saw a larger delay of 0.07 days annually. Similarly, corn harvesting has shifted later at a rate of 0.67 days per year, whereas soybean harvesting advanced slightly by 0.05 days each year. These shifts have impacted the overall growing season length: it extended by 0.66 days per year for corn, while slightly shortening by 0.12 days per year for soybean. Their sensitivity analysis further indicated that planting dates are particularly responsive to early-season temperature changes, while harvesting dates are affected by temperature variations throughout the entire growing season.

Precipitation influences phenological events differently, depending on regional climate conditions. In arid and semi-arid ecosystems, water availability is a critical factor determining the timing and duration of phenological phases (Cui & Shi, 2021; Gong et al., 2015; Li et al., 2018; Wang et al., 2019). For instance, Gong et al. (2015) showed that increase of preseason precipitation

alleviated water stress and prolonged the growing season of autumn in inner Mongolia grassland. Studies on African savannas and other water-limited environments show that phenological responses to precipitation are complex, as variability in rainfall can create unpredictable growing conditions (Adole et al., 2016; Guan et al., 2014). Conversely, in temperate climates, precipitation primarily affects the phenological timing of species adapted to consistent water availability, with significant delays occurring during periods of prolonged drought (Richardson et al., 2013). In regions where rainfall patterns are significantly impacted by the El Niño-Southern Oscillation, it is anticipated that the frequency and severity of El Niño events will rise, and this shift has already started to influence the phenology of tropical forests in Brazil.

Photoperiod, or the length of daylight, acts as a stable environmental cue for regulating phenology, particularly for species in high-latitude regions. Although photoperiod remains relatively constant compared to temperature and precipitation, it plays a vital role in synchronizing seasonal events such as flowering and leaf senescence. Plants use photoperiod as a signal to begin or end growth phases, which can help them avoid frost damage in temperate and boreal climates (Adole et al., 2019). However, Ettinger et al. (2021) suggests that as climate change shifts temperature patterns, photoperiod-based phenology may increasingly misalign with optimal growing conditions, leading to potential ecological mismatches.

Understanding how these factors interact is crucial for accurately modeling phenology, as demonstrated in studies that incorporate multiple climatic variables into predictive models for diverse ecosystems (Butler and Huybers, 2015; Kumudini et al., 2014; Zai et al., 2019). Moreover, the combined effects of these factors can differ across species, complicating ecological forecasting and necessitating species-specific models.

2.1.2.1: Factors Influencing Corn Development

Corn is a highly temperature dependent crop and other factors such as photoperiod, precipitation and humidity have a second order impact on its growth (Stewart et al., 1999; Streck et al., 2008; Zai et al., 2019). Moreover, these secondary factors, strongly interact with the primary influence of temperature, making it challenging to differentiate between their individual contributing effects (Olesen et al., 2012). Optimal corn growth occurs within a specific temperature range following which numerous studies, including those by Lobell et al. (2013) and Zai et al.

(2019), have utilized heat accumulation measures like Growing Degree Days (GDD) and Crop Heat Units (CHU) to detect key corn phenological stages.

GDD heat sum index (HSI) was developed by Gilmore and Rogers (1958) and is one of the earliest and most commonly used HSI in studying corn growth stages (Anandhi, 2016; Angel et al., 2017; Zai et al., 2019). It is calculated based on daily minimum and maximum temperatures, with the linear growth between the base (10°C) and upper (30°C) thresholds and no marginal growth outside these thresholds. Anandhi (2016) used GDD as an ecological indicator to study changing diurnal temperatures and their effects on plant growth. She identified significant changes in corn phenology with most stages, except tasseling, advancing by about one day per decade. While GDD provides a cumulative measure of how much heat corn has experienced over the growing season, CHU better accounts for the crop's sensitivity to temperature during both day and night (Kumudini et al., 2014; Zai et al., 2019). CHU incorporates a base temperature of 10°C for daytime temperatures and 4.4°C (40°F) for nighttime temperatures and captures the nonlinear impact of temperature on plant growth. Butler and Huybers (2015) have further shown that the relationship between growth and temperature is not uniform across different corn growth intervals.

2.2: Remote Sensing in Phenology Studies

Remote sensing is the collection and interpretation of information about the Earth's surface or other objects without direct physical contact (Lechner at al., 2020; Weiss et al., 2020). Remote sensing can involve various methods and technologies, not limited to satellite-based systems. It includes the use of aircraft, drones, ground-based instruments, and other platforms to gather data about the Earth's surface. Satellite imaging is a specific application of remote sensing; it refers to the process of capturing images or collecting data on Earth's surface or other celestial bodies using satellites. It focuses on the use of satellites as the primary means of data acquisition. Satellites are equipped with sensors or cameras that capture images or collect data across different parts of the electromagnetic spectrum (Asokan et al., 2019).

The images and data obtained through satellite sensors can be used for various purposes as it provides a valuable tool for observing and understanding our planet. It can contribute to a plethora of industries with its ability to provide a global perspective and consistent data over time. The focus of section 2.2 is to obtain domain-knowledge expertise on satellite imaging to be applied for vegetation phenology monitoring.

2.2.1: Overview of Satellite Remote Sensing Technologies

Over the past decade, satellite remote sensing technology has advanced significantly, leading to improvements in data quality, spatial resolution, coverage, and accessibility (Cao, 2022). The number of Earth observation satellites has grown remarkably, driven by advancements in miniaturization and the deployment of small satellites (Vastaranta et al., 2020). This has resulted in enhanced global coverage by many nations and businesses and more frequent image acquisition; transmitting images with ground resolution as fine as 62cm.

Governments and commercial satellite operators have made significant strides in making satellite imagery and data more accessible. Initiatives like the Copernicus program by the European Space Agency, Landsat program by NASA, MODIS and Sentinel data distributed through various platforms such as the have made vast archives of satellite imagery freely available to the USGS Earth Explorer, NASA Earth Data and Google Earth Engine public, fostering innovation and benefiting a wide range of users (Schepaschenko et al., 2019; Szantoi & Strobl 2019). For instance, the Landsat series, being operational for nearly five decades, offers a valuable long-term dataset for examining extended phenological patterns and assessing climate change impacts on vegetation cycles. Meanwhile, MODIS and Sentinel, with their short revisit intervals, are particularly suited for capturing rapid phenological events and observing seasonal dynamics (Gong et al., 2024).

Advancements in sensor technologies, such as synthetic aperture radar (SAR), have expanded the capabilities of satellite imaging. The ability of SAR to penetrate clouds and provide high-resolution images regardless of weather conditions or daylight, helped Kobayashi & Ide (2022) to identify critical phenological developments in rice plant. It captured changes in water content determining the optimal harvest time by indicating the grain-filling phase, a period critical in rice to yield quality. The frequent revisit period of Sentinel-1 also allowed for continuous monitoring, making SAR an invaluable tool for real-time crop management and decision-making in agriculture.

Introduction of multi-spectral and hyperspectral imaging capabilities allowed satellites to capture data across multiple parts of the electromagnetic spectrum which include visible, near-infrared, shortwave infrared, and thermal bands. Studies conducted by Yang et al. (2021), Ramadhani et al. (2020) and many others (e.g., Kerner et al., 2022; Nieto et al., 2021; Ye et al., 2023) have demonstrated the use of vegetation indices, such as the Normalized Difference

Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), and the Land Surface Water Index (LSWI) derived across multiple spectral bands from satellites like MODIS, Landsat, Sentinel-2 in monitoring phenology of diverse crops such as corn, soybean, rice, and maize. Hyperspectral imaging provides even more detailed spectral information, enabling enhanced analysis of vegetation, mineralogy, and environmental conditions. The study by de Moura et al. (2017) utilized hyperspectral data from MODIS alongside data from a hyperspectral tower camera to examine the seasonal dynamics of Amazon rainforest canopy phenology during the dry season. Specifically, they analyzed the spectral characteristics of vegetation as the forest transitioned through various phenological stages, demonstrating that MODIS data could effectively capture canopy "greenness" and subtle leaf changes throughout the dry season, contributing to better understanding of how seasonal water limitations affect vegetation in Amazonian ecosystems.

These developments have democratized access to satellite imagery, increased the availability of data for various applications, and expanded our understanding of the phenology dynamics and changes over time.

2.2.2: Vegetation Indices for Corn Phenology Detection

In remote sensing studies of vegetation phenology, vegetation indices (VIs) derived from combinations of spectral bands are commonly used instead of raw spectral band data (Zeng et al., 2020). These indices, which represent mathematical transformations combining two or more spectral bands, are particularly effective in capturing key vegetation characteristics, such as chlorophyll content, water retention, and canopy structure. As a result, VIs often serve as stronger indicators of vegetation health and condition compared to analyses of individual spectral bands alone, making them essential tools in monitoring and assessing phenological changes in ecosystems.

NDVI is one of the most extensively used VIs in phenological studies. Its use is well established in literature as it is sensitive to changes in green biomass and has a strong relationship with photosynthetically active vegetation. It leverages the contrast between red and near-infrared (NIR) spectral bands, making it valuable for monitoring greenness and detecting the onset and senescence of growth stages across diverse crops. Diao (2020), Yang et al. (2023), Ye et al. (2023) and Zhao (2022) are some of the researchers amongst many others who have demonstrated the use of NDVI in corn phenology detection. However, NDVI has a potential to saturate when observing

dense biomass and thus become insensitive to plant canopy changes making it challenging to detect the change from vegetative to reproductive phases of corn (Venancio et al., 2020; Wang et al.,2016).

To address some limitations of NDVI, the EVI was developed to improve sensitivity to dense canopy vegetation and minimize atmospheric influences. EVI incorporates additional correction factors and the blue band, which helps reduce soil and atmospheric distortions, thereby offering a clearer signal in regions with high biomass (Zhong et al., 2014). Liu et al. (2018) and Zhong et al. (2016) demonstrated the effectiveness of EVI in monitoring phenology in agricultural systems where crops like corn and soybean experience rapid, high-density growth during peak phases.

The Land Surface Water Index (LSWI) is a VI for assessing water content in vegetation due to its sensitivity to fluctuations in plant moisture levels. Utilizing shortwave infrared (SWIR) and NIR spectral bands, the LSWI is particularly beneficial in agricultural regions where water stress, flooding, or drought conditions impact crop growth and phenology. Wu et al. (2024) demonstrated the utility of LSWI in phenological monitoring of corn, where it effectively detected critical growth stages, such as jointing and flowering, due to its high correlation with vegetation water content. This sensitivity resulted in an accurate root mean square error (RMSE) range of three to four days for these stages. Comparatively, the NDVI was more effective in identifying earlier growth stages like emergence and the seven-leaf stage, showing an RMSE of four days. By integrating LSWI with NDVI, researchers observed a significant increase in the accuracy of phenological stage identification, underscoring the benefits of using multiple VIs to capture comprehensive information on crop development.

2.2.3: Phenological Modeling of Vegetation Indices

Numerous models have been proposed to detect within-season phenological stages of corn (e.g., emergence or maturity dates) using remotely sensed VI (Gao & Zhang, 2021; Tang et al., 2016; Yang et al., 2020; Zeng et al. 2020). These methods can be classified into curve-based and trend-based approaches. Curve-based methods primarily focus on fitting mathematical functions to time-series data to model the seasonal dynamics of vegetation and require vegetation development data from previous years as ancillary information for the current year. These methods typically use functions like logistic models (Ban et al., 2016; Leale et al. 2023, Zhang et al., 2003), double-logistic functions (Beck et al., 2006; Guo et al., 2022; Myers et al., 2019), and linear regressions

(Diao, 2020). For example, the double-logistic model is widely used due to its ability to represent the asymmetry in the onset and offset of vegetation greenness. This model captures the rapid increase and gradual decrease of vegetation indices, providing a robust framework for identifying phenological markers such as green-up and senescence dates (Zhang et al., 2003).

Zeng et al (2016) proposed a hybrid corn and soybean phenology detection model incorporating "shape-model fitting" on MODIS wide dynamic range vegetation index (WDRVI), using scale and shift parameters to fit observed WRDVI on crop-specific shape. In another study, Shen et. al (2022) explored hybrid piecewise logistic model-based land surface phenology detection algorithm (Zhang et al., 2003) to identify phenometrics over the US corn belt on EVI2 time series derived from integrated harmonized Landsat and Sentinel-2 (HLS) and visible infrared imaging radiometer suite (VIIRS). The proposed algorithm produced an absolute average difference <5.3 days for six corn phenometrics observed at various crop growth stages. A curve-change based dynamic threshold approach was used by Yang et al. (2021) to detect the planting and harvesting dates of corn highlighting the significant role of climatic factors in growing season length by 0.66 days/year. Ye et al. (2023) introduced Derivative Dynamic Time Wrapping (DDTW) method as an innovative approach for detecting corn phenology using Sentinel-2 time-series data. The DDTW algorithm incorporated first-order derivative information to capture the rate of change in the NDVI values resulting in RMSE of detected corn phenology of less than 6 days.

Despite their advantages, curve-based phenology detection methods face challenges, particularly in data quality and availability. Cloud cover, sensor noise, and varying spatial resolution can introduce errors in time-series data, complicating the accurate detection of phenological events (Liu et al., 2013; Zhu & Woodcock, 2012). Additionally, they may struggle to adapt to irregular phenological patterns caused by environmental stressors or management practices, leading to potential inaccuracies in detecting the exact timing of growth stages (Diao, 2020).

Another widely used methodology to detect trends within the time series data involves computing moving averages (Gao et al., 2020, Reed et al., 1994). Unlike the curvature-based approach, trend-based methods rely solely on remote sensing data from the current growing season (Gao & Zhang, 2021). Recently, two methods, capable of running in near-real time, have been developed for mapping within-season crop emergence (WISE) and within-season crop termination

(WIST) (Gao et al. 2020). Moving average convergence divergence (MACD) was used to detect the trend in the VENμS, Sentinel-2, and HLS time series. They achieved an RMSE of < 6 days to detect emergence and harvest dates across all studied corn fields. A major drawback of this approach is the ineffectiveness of MACD to detect green-up or senescence when VI-time-series is linear with respect to date during these transition periods. Moreover, the optimal parameter values employed in experimentation may not be the best fit for other sensors and regions, requiring manual parameter tuning and field-scale validation across different agroecosystems.

2.3: Machine Learning for Phenology Detection

The integration of machine learning algorithms has further enhanced the predictive capabilities and analysis of phenological data (Czernecki et al., 2018; Murguia-Cozar et al., 2021; Nieto et al., 2021; Ramdhani et al., 2020; Yang et al., 2023; Zhao et al., 2022). Table 2 presents a comparison of existing studies on phenological mapping utilizing machine learning approaches.

Ramadhani et al. (2020) focused on combining ground-based observations with Landsat-8 Surface Reflectance data to map rice phenology in Indonesia. Using indices like EVI and LSWI, the study employed Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Networks (ANN), having the SVM achieved the best model accuracy. Despite delivering reliable results over different seasons, the model encountered limitations due to the coarse resolution of Landsat-8 and seasonal changes in planting times, which led to misclassification. Czernecki et al. (2018) explored plant phenology reconstruction by integrating MODIS vegetation indices and gridded meteorological data in Poland. Through various machine learning models like Generalized Boosted Models (GBM) and Lasso Regression, the study highlighted that meteorological variables were critical in predicting early spring phases, while MODIS-derived indices were more influential for autumn phases. This integration improved accuracy in identifying phenological stages by reducing the RMSE by 0.6 to 4.6 days, yet the models' accuracy varied across phenophases, especially during extreme temperatures. Murguia-Cozar et al. (2021) used Sentinel-2 imagery to classify corn phenological stages based on texture, colour, and vegetation features, applying classifiers like SVM and k-Nearest Neighbors (kNN). The study achieved high accuracy with the quadratic SVM, particularly in identifying the "end of senescence" stage. However, the model struggled to classify the "tassels and ears" stage due to limited sample representation. This work

illustrates the potential of incorporating texture and colour features with VIs for phenological mapping, though environmental and management variations remain challenges for model accuracy.

Recent advances in deep learning with attention mechanisms have been found conductive to within-season crop phenology mapping (Yang et al. 2023; Zhao et al. 2022). By dynamically weighting input data, attention mechanisms help the model to focus on the most relevant temporal or spatial features within phenological time series. Yang et al. (2020) introduced an attention-based 1D-CNN (At1DCNN), utilizing MODIS NDVI data to map corn and soybean phenology in Illinois, demonstrating the scalability of deep learning for crop mapping. Zhao et al. (2022) took this approach further by integrating PhenoCam ground observations, Sentinel-1 SAR, and Sentinel-2 optical imagery in their Deep-CroP model, which used CNN and Transformer architectures to bridge the gap between satellite-derived and ground-based phenology data. This approach reduced satellite-ground discrepancies significantly but was constrained by high computational costs and occasional misclassification due to complex SAR signals. These studies indicate that deep learning can align phenological observations across data sources, providing more accurate and scalable phenology tracking in various crops.

2.3.1: Identified Research Gap in Phenological Modelling

Despite their increased accuracy as compared to traditional models, existing ML models designed for crop phenology mapping (as summarized in Table 2) face several significant limitations. These include the reliance on less training and often noisy data, leading to inaccuracies, particularly in mixed cropping regions and during colder seasons. Models struggle with temporal dynamics, often failing to capture the complex interactions between environmental variables and vegetation responses over time. Additionally, they exhibit inconsistent accuracy across different phenophases. Issues like underrepresented phenological stages and multicollinearity among predictors further reduce model performance, highlighting the need for more sophisticated and adaptable model. The change in corn phenophases often depends on changes observed in mutually related climatic and vegetation features, and their quantitative influence across different phenophase may differ. Thus, we aim to develop a temporal multivariate attention mechanism to dynamically focus on most relevant features and time points, which is particularly beneficial for handling complex interactions between climatic and vegetation features to detect phenological stages accurately.

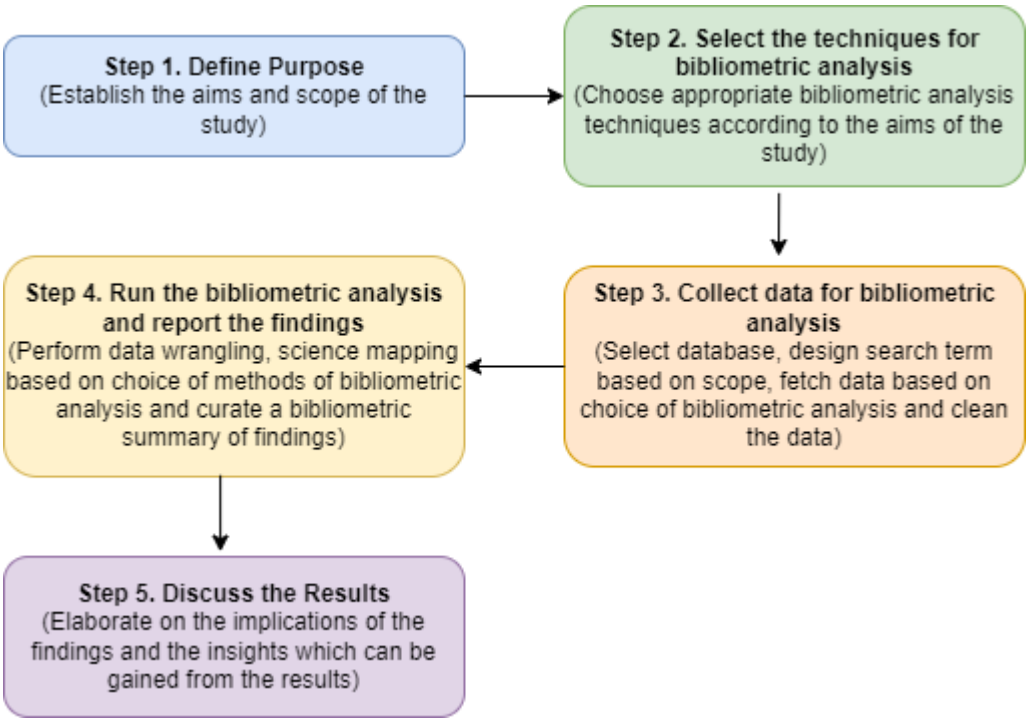

Figure 1. Workflow of bibliometric analysis

Table 2. Comparison of existing studies on crop phenological mapping using machine learning models

	Problem Covered	**Experimentation**					**Models Used**	**Results**	**Limitations**
		Data Source	**Studied Area**	**Time Period**	**Studied Vegetation**	**Studied Features**			
Ramadhani et al., 2020	To develop a mechanism to combine ground-based observation with remote sensing data for mapping rice phenology	Ground based observations from webcams and Landsat-8 Surface Reflectance data.	Indramayu (Indonesia)	2014-2016	Rice	Ground Validation Data, EVI, Land Surface Water Index (LSWI)	Random Forest (RF), Support Vector Machine (SVM) and Artificial Neural Networks (ANN)	The results show that the nonlinear support vector machine (SVM) algorithm gives the best model accuracy based on the test dataset and the lowest temporal changes (<11%) achieving an overall accuracy of 70.5%. Spatial-temporal assessment of rice phenology and bare land from Landsat-8 indicated that the models were reliable and robust over different seasons and years	Limited training data. Coarse resolution (30m) of Landsat-8 can lead to misclassification in mixed cropping regions. Some parts of the studied area were wrongly classified because of different agronomic practices, topographic variations and presence of cloud cover in data. The model's performance was also affected by the seasonality and shifting planting times in the region.
Czernecki et al., 2018	To reconstruct plant phenological phases using a combination of satellite-derived vegetation indices and gridded meteorological data by integrating machine learning	E-OBS gridded meteorological dataset by the European Climate Assessment & Dataset (ECA&D), Ground based observation by the	Poland	2007-2014	10 plant species	Temperature. Precipitation, NDVI, EVI, Leaf Area Index (LAI), Fraction of Photosynthetically Active Radiation (fPAR) and snow cover	Multiple Linear Regression (MLR), Multiple Linear Regression with Stepwise	The GBM and lasso models provided the best predictions, with RMSE values ranging from 6.4 to 12 days depending on the phenophase. MODIS-derived variables were more influential for autumn phenophases, while meteorological variables were more important for early spring	The models showed varying accuracy across different phenophases, with some phases (like early spring and late autumn) being harder to predict accurately. MODIS-derived indices are noisy, especially during the colder part of the year, which affected model performance. The study was limited by the

	techniques	Institute of Meteorology and Water Management - National Research Institute (IMGW-PIB), MODIS and National Snow and Ice Data Center					Selection (MLR AIC), Lasso Regression, Principal Component Regression (PCR), Generalized Boosted Models (GBM), And RF	phenophases. The integration of both data sources (meteorological and satellite) improved model accuracy by 0.6 to 4.6 days in RMSE.	relatively short time period and spatial coverage of the dataset, which could affect the generalizability of the models.
Murguia-Cozar et al., 2021	To develop and evaluate supervised classification models to accurately classify different phenological stages based on texture, vegetation, and color indicators	Sentinel-2 satellite images provided by the European Space Agency (ESA) through the Copernicus program	Hidalgo, Mexico.	2019	Maize crops	Texture Features: Moran's I (LISA), Local Binary Pattern (LBP). Vegetation Features: LAI, color ratios. Color Features: RGB color model, Lab* color space, YIQ color space, and NIR band.	Linear Discriminant (LD) classifier, SVM and k-Nearest Neighbors (kNN) Classifier	The quadratic SVM classifier achieved the highest overall accuracy of 82.3% after reducing the features to 45 key characteristics. This model was most accurate in classifying the "end of senescence" stage (92%) and least accurate in classifying the "beginning of senescence" (74%). The RGB color model, NIR band, color ratios, and LISA were identified as the most critical indicators for accurately classifying maize phenology.	The study was unable to accurately classify the "tassels and ears" phenological stage due to a limited number of samples and possibly the inadequacy of the selected indicators to capture this stage's characteristics. The variability in planting dates, management practices, and environmental conditions introduced noise into the model, affecting classification accuracy.
Nieto	To improve the	Landsat 8,	Kansas,	2013-2018	Maize crops	NDVI, EVI,	RF	The combination of Vegetation	The model showed lower accuracy

et al., 2021	precision of crop phenology classification by integrating ground truth field data, satellite imagery and weather data.	Gridded surface meteorological data from PRISM and NLDAS (North American Land Data Assimilation System) and Ground truth data on maize phenology provided by Crop Quest Inc.	USA			Green Chlorophyll Vegetation Index (GCVI), Global Vegetation Moisture Index (GVMI) and Normalized Difference Water Index (NDWI), Precipitation, maximum and minimum temperature, Vapor Pressure Deficit (VPD), Growing Degree Units (GDU)	classificatio n model	Indices (VIs) and Vapor Pressure Deficit (VPD) provided the highest accuracy, ranging from 86% to 98%. The combination of Vegetation Indices (VIs) and Vapor Pressure Deficit (VPD) provided the highest accuracy, ranging from 86% to 98%.	when applied to different spatial locations compared to the training data. This suggests that the model's performance is more sensitive to changes in spatial context than in temporal context. Certain phenological classes were underrepresented in the dataset, leading to poorer classification performance for those classes. he full model with all 21 predictors performed poorly, likely due to multicollinearity among the surface reflectance bands.
Yang et al., 2023	To proposes an innovative within-season emergence (WISE) phenology normalized deep learning model towards scalable within-season crop mapping	MODIS MCD43A4 Version 6 product, Cropland Data Layer (CDL) by USDA, Crop progress report (CPR) by USDA NASS	Illinois, USA	2017-2020	Corn and soybeans	NDVI	Deep learning classifier using temporal 1D-CNN and an attention module		
Zhao	Proposes a deep	PhanoCam,	Northern	2018-2019	corn, rice,	NDVI, GCC	deep	The model significantly improved	The performance was constrained by

et al. 2022	learning model Deep-CroP to bridge the gap between ground-based observations (PhenoCam) and satellite-derived data for crop phenology tracking	Sentinel-1 (SAR) and Sentinel-2 (optical) imagery	California, USA and northeast Arkansas, USA		winter wheat, tomatoes, and soybeans	(Green Chromatic Coordinate) and SAR backscatter signals	learning framework combining CNN and Transformer architectures	the alignment of satellite-derived phenology data with ground observations, reducing discrepancies from over 30 days to an average of 17 days.	the size and quality of PhenoCam samples. The computational cost of the model was high, especially when applied to large-scale and long time-series data. Occasional confusion between different crop phenology stages due to the complexity of SAR backscatter signals.

2.4: Bibliometric Analysis

"Bibliometrics" was first proposed by Pritchard in 1969 and is a powerful tool to assess and analyze progress in scientific research through statistical methods. It can provide valuable insights into impact of the research, the growth of a particular field of study, the productivity of researchers or institution and the interconnections between different areas of research through bibliometric indicators such as citation analysis, journal impact factor, collaboration analysis, bibliographic coupling, co-citation analysis and H-index (Ellegaard et al., 2015; Szomszor et al., 2021).

One of the key research contributions of this thesis is to build a phenological monitoring model which can contribute to understanding and managing climate change impacts on vegetation phenology. Bibliometric analysis is incorporated in this literature review to position this contribution as novel within the field, add rigor, objectivity and comprehensive perspective to the existing literature on phenological studies. The stepwise approaches described by Donthu et al. (2021) and Ellegaard et al. (2015) to conduct bibliometric analysis served as a framework for methodology, which was then modified to meet the research topic and available data.

2.4.1: Approach

This section discusses the workflow involved in conducting bibliometric analysis outlined in Figure 1. The first step was to define the purpose of the bibliometric analysis which involved establishing aims and scope of the study to warrant the use of bibliometric analysis, which must occur before selecting the techniques of bibliometric analysis (step 2), gathering of bibliometric data (step 3) followed by running the bibliometric analysis (step 4). Lastly, the results were analyzed to extract insights and satisfy the aims established for bibliometric analysis.

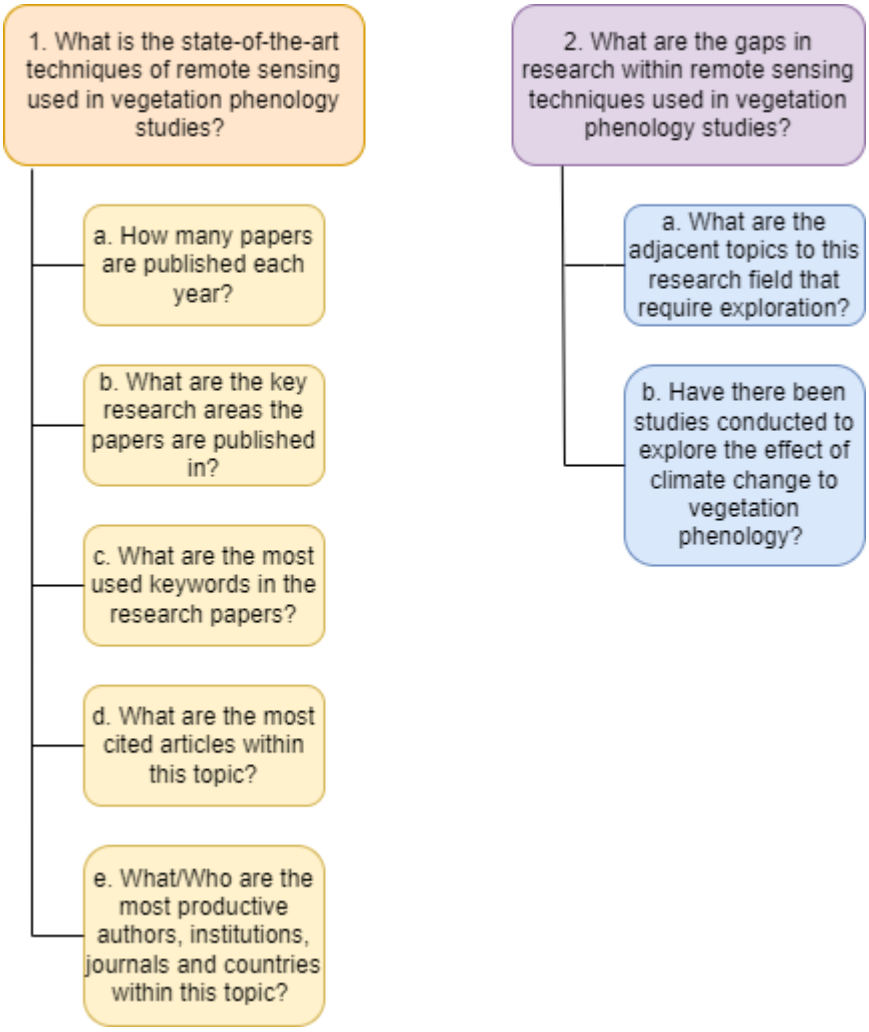

Figure 2. Bibliometric questions to meet the aims of bibliometric analysis

2.4.2: Methodology

Step 1. Define Purpose

The primary purpose of conducting this bibliometric analysis is to compile a comprehensive review of the research topic based on the existing literature on vegetation phenology studies. The scope of the study is restricted to the use of remote sensing methods in the vegetation phenology studies. The aims of the bibliometric analysis are two-fold: (1) to understand the state-of-the-art techniques of remote sensing used in vegetation phenological study and (2) to identify the gaps in research on remote sensing techniques used in vegetation phenological study. The questions of interest to meet the aims of bibliometric analysis are defined in Figure 2.

Step 2. Select the techniques for bibliometric analysis

The bibliometric techniques used to find answers to the bibliometric questions defined in step 1 are listed in table 3.

Table 3. Bibliometric Questions and Methods

N	Bibliometric Questions	Bibliometric Method
1a.	How many papers are published each year?	Year-wise evolution of publications (publication summary)
1b.	What are the key research areas papers are published in?	Most productive research areas (research area keyword summary)
1c.	What are the most used keywords in the research papers?	Most used keywords (word cloud summary)
1d.	What are the most cited articles within the topic?	Most influential publications (citation analysis)
1e.	What/who are the most productive authors, institutions, journals, and countries within this topic?	Most productive authors, institutions, journals and countries (publication analysis and authorship analysis)
2a.	What are the adjacent topics that require exploration?	Co-word analysis
2b.	Have there been studies conducted to explore the effect of climate change to vegetation phenology?	Manual analysis to determine the trend and novelty

Step 3. Collect data for bibliometric analysis

The main investigation is based on remote sensing in vegetation phenology studies and for this purpose, we chose *Scopus* database. Scopus indexes the largest number of journals in all fields as compared to *Web of Science (WOS)* (Li et al., 2010). We restricted the scope to one database to prevent data preprocessing and data integration steps required to merge results from two or more data sources. The next step was to design the search criteria based on the scope of the purpose defined in step 1. The search query created was:

TITLE-ABS-KEY(("vegetat pheno*" OR "crop* pheno*" OR "plant* pheno*" OR "forest* pheno*") AND ("remot* sens*" OR "satellite* imag*")) AND PUBYEAR > 1998 AND PUBYEAR < 2023 AND (LIMIT-TO (LANGUAGE,"English")) AND (LIMIT-TO (DOCTYPE,"ar") OR LIMIT-TO (DOCTYPE,"re"))*

The topic of interest was studies conducted in vegetation/crop/plant phenology using remote sensing or satellite imagery. The first part of the query reflects the topic and searches publications based on title, abstract or keyword containing the search terms and meeting the criterion of search query. The remainder of the query (i.e., the second, third and fourth components) reflects the inclusion and exclusion conditions. We have limited our search to include only articles or review articles published between 1999 to 2022 (inclusive) in English language. Based on the search query, there were 1,206 documents found. We exported the citation information, bibliographical information, abstract and keywords and references of these documents to a CSV file.

Since *Scopus* database is not exclusively designed for bibliometric analysis (Donthu et al., 2021), data cleaning becomes an essential step to prepare data for bibliometric analysis. It was conducted in MS Excel. Firstly, we replaced all the instances of non-abbreviated definitions to their respective abbreviations such as we replaced all instances of “Normalized Difference Vegetation Index” to “NDVI”, “Moderate Resolution Imaging Spectroradiometer” to “MODIS”, “Enhanced Vegetation Index” to “EVI” etc. Next, we replaced all the hyphen separated words to un-hyphenated words, e.g., we changed “remote-sensing” to “remote sensing”. Lastly, we converted the plural words to their singular forms, e.g., “systems engineering” to “system engineering”. We used the cleaned csv file to run bibliometric analysis and report the findings.

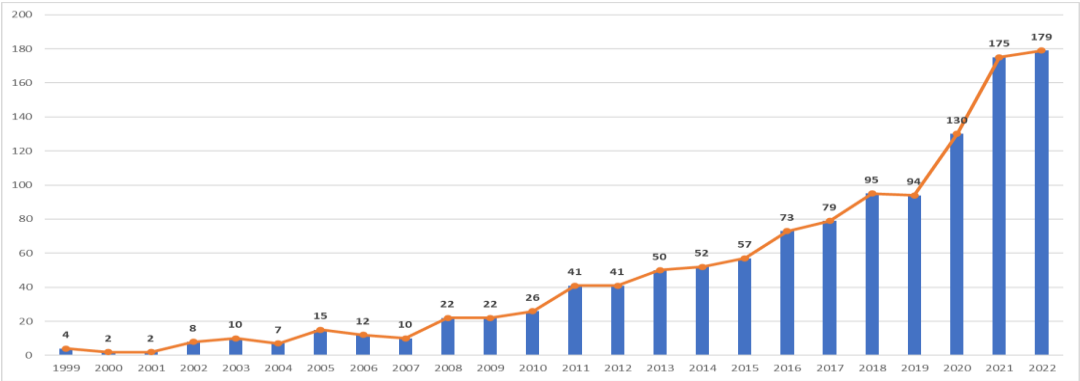

Figure 3. Year by year trend of number of publications in research domain

Step 4. Run the bibliometric analysis and report the findings

MS Excel and VOS Viewer software (VOSViewer, 2023) were used to perform data wrangling on the data obtained from step 3 to answer the bibliometric questions listed in Table 2. For each of those bibliometric questions, their respective bibliometric technique was applied as outlined below.

BQ1a: How many papers are published each year?

Figure 3 depicts the year-wise evolution of publications in the research domain and an increasing trend is observed from 2007 onwards depicting a rapid annual growth in publications. In the year 2021, the publications increased from 130 to 175 indicating a 35% growth compared to previous years.

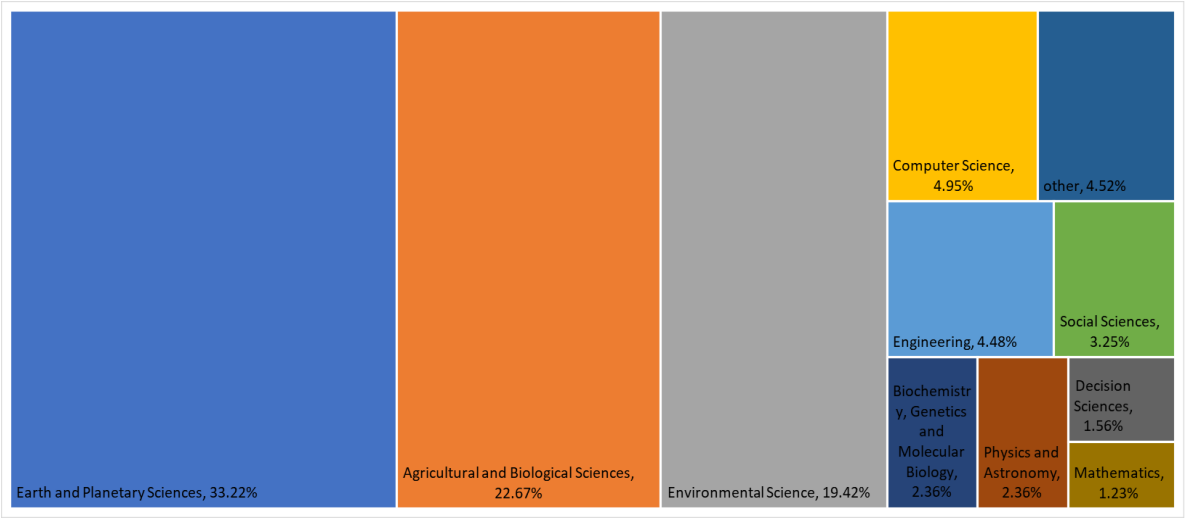

Figure 4. Most productive research areas for research involving RS in Vegetation Phenology

BQ1b: What are the key research areas papers are published in?

There are more than 22 research areas with publications related to remote sensing in vegetation phenology studies. Figure 4 highlights the 10 most prolific research fields, with Earth and Planetary Sciences, Agriculture and Biological Sciences, and Environmental Science collectively accounting for 75.31% of total publications. Given the strong connection between remote sensing data (RSD) and vegetation phenology, it is understandable that these disciplines dominate the field. However, remote sensing also plays a role, albeit to a lesser extent, in other areas such as Medicine, Economics, and Veterinary Science.

BQ1c: What are the most used keywords in the research papers?

We used word cloud to determine the most used keywords in the publications. Word cloud is a data visualization technique that displays the most commonly used terms in a given text, with the word in the center having the highest frequency of occurrence, and as we move out from the center, those words are used less frequently. We created the word cloud using wordclouds.com to visualize the occurrences of author keywords in the research papers. We applied several text mining techniques to ensure that the word cloud would consider the frequency of each phrase and not individual words from author keywords. A total of 1,482 phrases were included in the word cloud as shown in Figure 5. The top 10 keywords including their frequencies are depicted in Figure 6. The analysis indicates that most of the studies used MODIS satellite data and NDVI in their vegetation phenology studies. It also hints that a significant number of conducted studies related to phenology and climate change. However, we do observe that machine learning and deep learning techniques have been underexplored in phenology studies. There are only 120 phrases out of 1,482 with occurrences over 5 which explains the variability of research topics within the selected domain.

BQ1d: What are the most cited articles within the topic?

We have documented the details of 10 most cited publications in Table 4. The oldest article goes back to 1999 and the most recent was published in 2019. The topics covered by these articles range from phenology shift assessment to advances in RS technologies for phenology studies. Most of the journal focus on various aspects of remote sensing in Environmental Sciences.

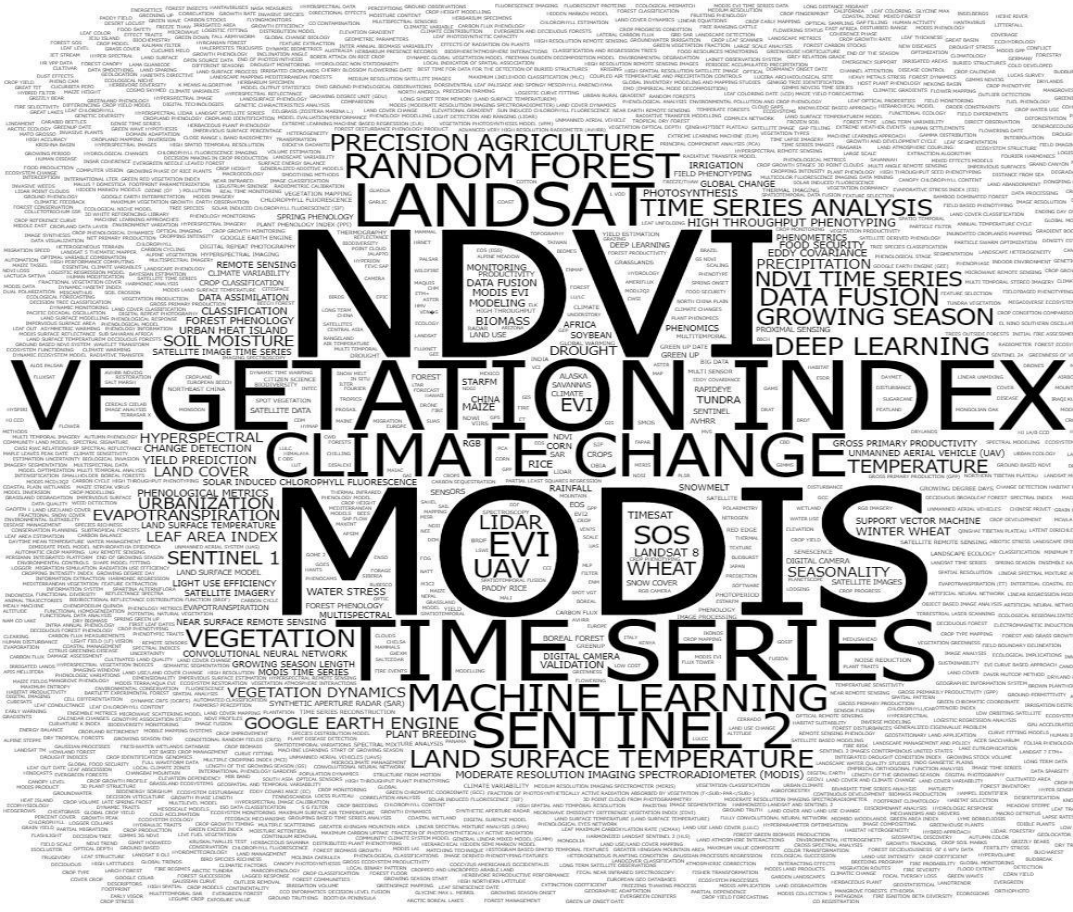

Figure 5. Word cloud of most used words/phrases

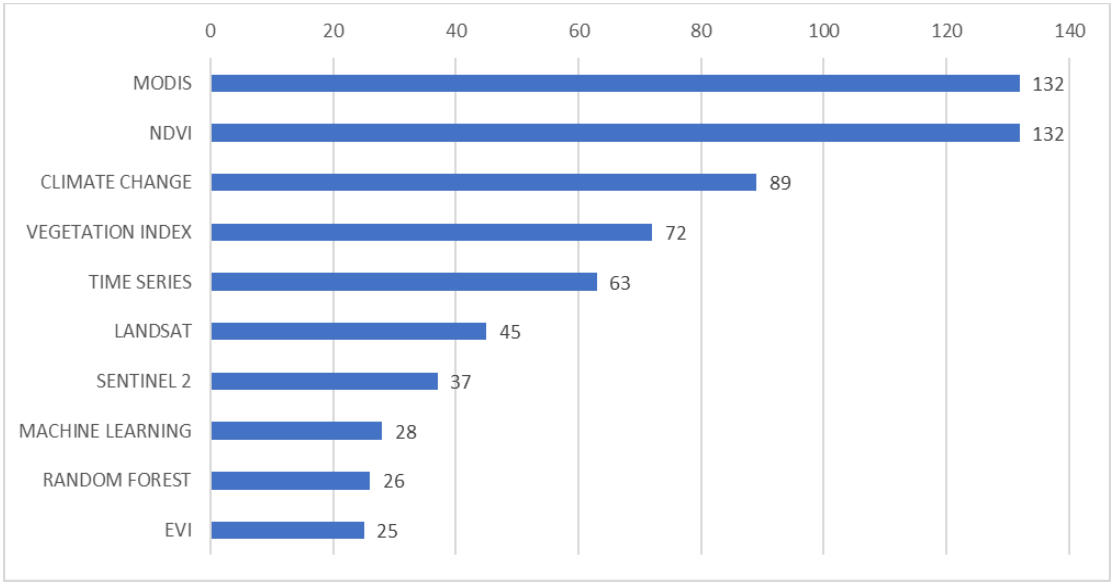

Figure 6. Frequency chart of top 10 phrases in author keywords

Table 4. Ten most cited publications

Title	Author(s)	Year	Journal	Citation Count
Monitoring vegetation phenology using MODIS	Zhang et al.	2003	Remote Sensing of Environment	1908
Shifting plant phenology in response to global change	Cleland et al.	2007	Trends in Ecology and Evolution	1607
Intercomparison, interpretation, and assessment of spring phenology in North America estimated from remote sensing for 1982-2006	White et al.	2009	Global Change Biology	825
Monitoring land-cover changes: A comparison of change detection techniques	Mas J.-F.	1999	International Journal of Remote Sensing	783
Improved monitoring of vegetation dynamics at very high latitudes: A new method using MODIS NDVI	Beck et al.	2006	Remote Sensing of Environment	734
A crop phenology detection method using time-series MODIS data	Sakamoto T et al.	2005	Remote Sensing of Environment	726
Plant phenology and global climate change: Current progresses and challenges	Piao S et al.	2019	Global Change Biology	689
A review of imaging techniques for plant phenotyping	Li L et al.	2014	Sensors (Switzerland)	657
Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs	Atzberger C.	2013	Remote Sensing	639
Amazon rainforests green-up with sunlight in dry season	Huete A.R et al.	2006	Geophysical Research Letters	607

BQ1e: What/who are the most productive authors, institutions, journals, and countries within this topic?

The 10 most productive authors with their respective number of publications are shown in Figure 8. Fredl, M.A and Zhang, X. have the highest number of publications which is 22. Both of them have co-authored 8 out of those 22 publications and have also worked together on the most cited article, "Monitoring vegetation phenology using Modis" (2003). The majority of articles are multi-authored. There are 74 authors who have published more than 5 articles and over 90% of authors have published only one article. The ten most productive institutions and journals are presented in Figure 8 and 9, respectively. The Chinese Academy of Sciences has produced the maximum number of publications (165). The most productive journal, Remote Sensing (MPDI), has published 195 out of 1206 articles or 16% of their total number. Remote Sensing of Environment (Elsevier) has the second most publications in the research domain with approx. 9% of total publications. It can be concluded that studying the evolution of our research topic should include a review of articles published by MPDI and Elsevier. Figure 10 depicts the most productive countries with their publication counts. The top 10 countries in order are: the United States, China, Canada, Germany, United Kingdom, France, Italy, Australia, Japan and Spain. USA (443) and China (379) have together published 822 out 1206 articles (approx. 68%). There are 90 countries that have contributed to the research topic out of which 26 countries have one publication.

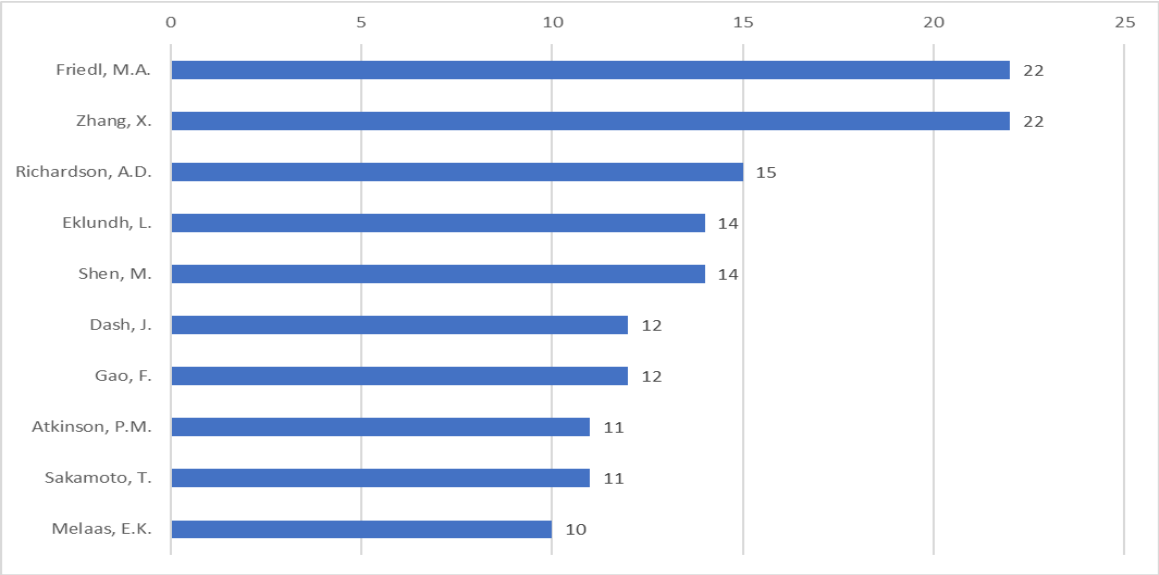

Figure 7. Ten most prolific authors

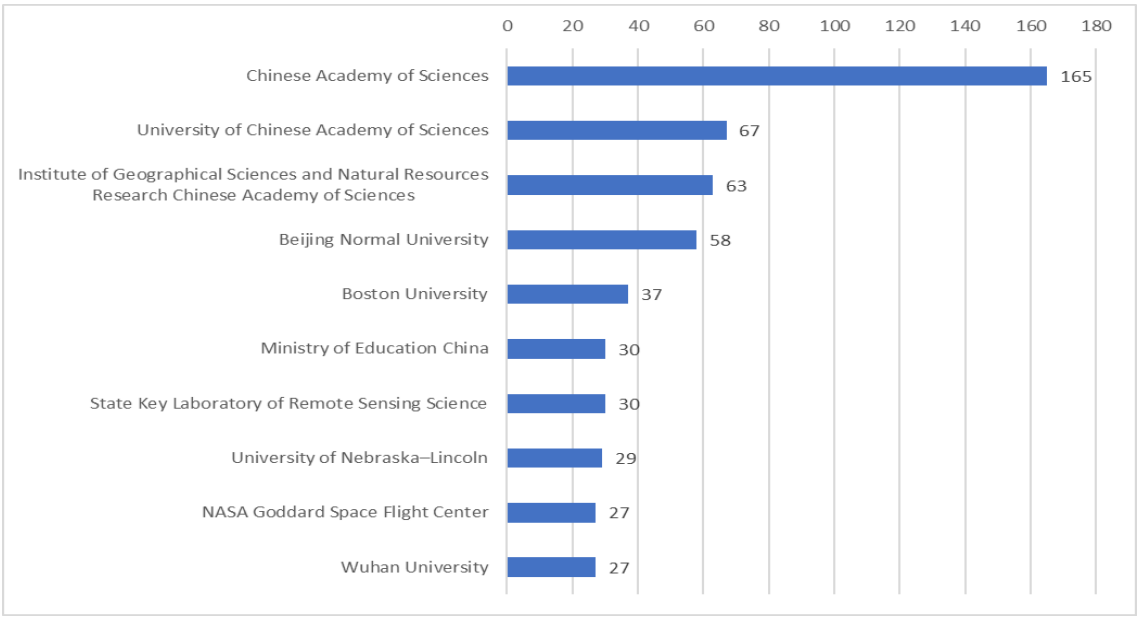

Figure 8. Ten most productive institutions

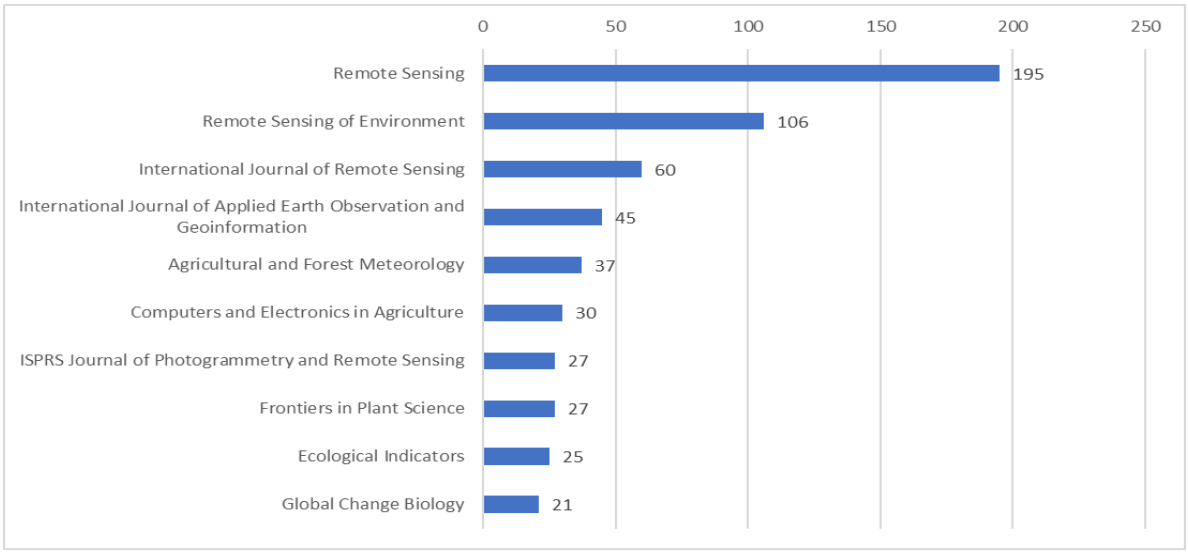

Figure 9. Ten most significant journals

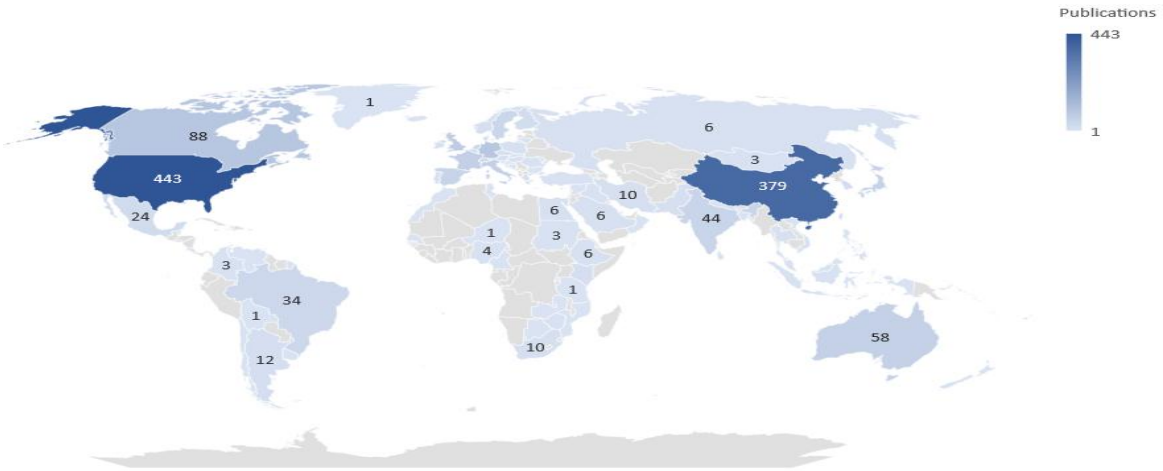

Figure 10. Most productive countries

BQ2a: What are the adjacent topics that require exploration?

Co-word analysis is a method used in bibliometric analysis to study the relationships between the terms in a collection of documents. It helps the researchers identify patterns of co-occurrence and relationships between terms to understand underlying themes, concepts, and trends within the research topic. We used VOSViewer software to visualize the co-occurrence network of keywords in our bibliographic data. It creates visualizations of co-occurrence networks, where terms (words or phrases) are represented as nodes and the strength of their co-occurrence relationships is represented by the proximity and thickness of connecting lines, i.e., edges. We uploaded all the keywords from our bibliographic data into VOSViewer and applied several configurations to generate the resultant co-occurrence network.

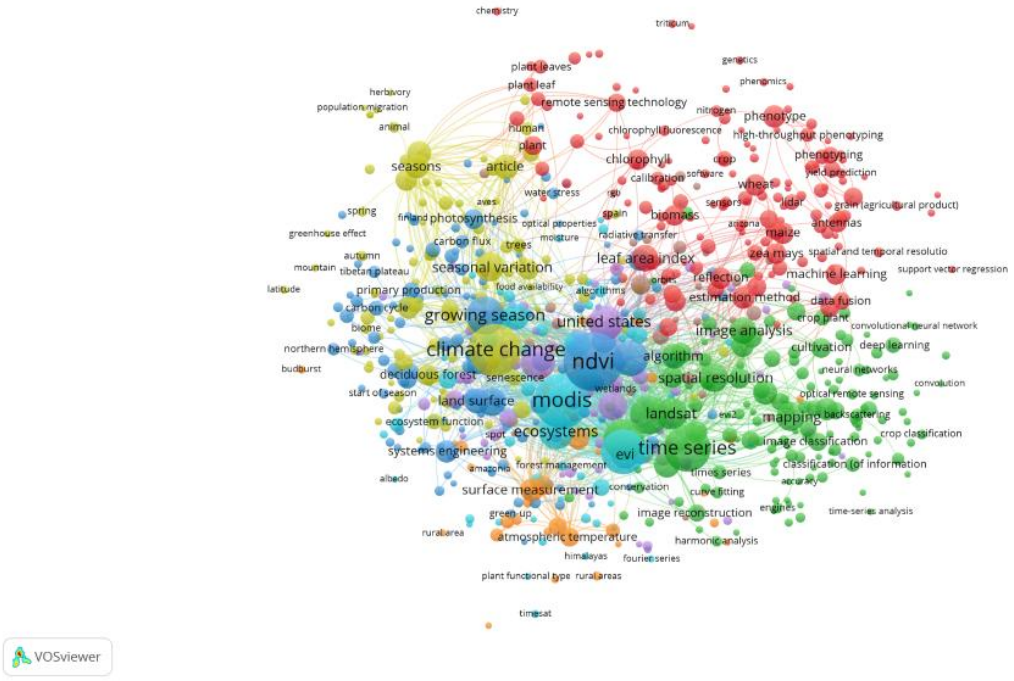

Figure 11. Keyword co-occurrence network

To create the network, each phrase had to occur a minimum of 5 times, and thus out of 7350 phrases, only 778 of them met the threshold. Furthermore, these 778 phrases were pruned to remove phrases from the search query (such as vegetation phenology, satellite imagery, remote sensing, phenology, vegetation, plants, etc.). 761 phrases in total were mapped to form the co-occurrence network in 3 types of visualizations, namely, based on clusters of keywords (Figure 11), developments over time (Figure 12) and density of co-occurrence relationships (Figure 13). VOSViewer can automatically identify clusters of related terms within the co-occurrence network which can provide a clear overview of the main themes and concepts present in bibliographic data. We set the layout attraction to 2 and repulsion to 1 to visualize the distinct clusters clearly as shown in Figure 11. VOSViewer partitioned 761 phrases into 8 distinct clusters in the co-occurrence network. The red cluster (Cluster 1) consists of 153 phrases such as leaf area index, phenotyping, biomass, chlorophyll, chemistry, corn, maize, etc. It mostly concentrates on vegetation, different types, and their chemical compositions which can be used in phenology studies. The green cluster (Cluster 2) consists of 147 phrases such as image analysis, machine learning algorithms, decision trees, image classification, image processing, image segmentation, time series analysis, etc. Based on the presence of these terms, Cluster 2 mostly focuses on different techniques applied in remote

sensing to conduct analysis and retrieve research results in phenology studies. The dark blue cluster (Cluster 3) and Light blue cluster (Cluster 4) contains a total of 193 phrases mostly focusing on the remote sensing part of the studies with keywords like MODIS, radiometers, GIS, NOAA satellite, also including vegetation indices like NDVI, EVI which can be retrieved from remote sensing data. The yellow cluster (Cluster 5) consists of 120 phrases with an emphasis on climate change and seasonal variation. The orange cluster (Cluster 6) has phrases such as atmospheric temperature, land surface temperature, surface measurement, etc. and therefore is a land surface model focused cluster. The pink cluster (Cluster 7) represents different types of forests, wetlands, agricultural locations and geographical areas which were studied in the publications. Cluster 8 which is the brown cluster is the smallest of 8 clusters and consists of synonyms from other clusters.

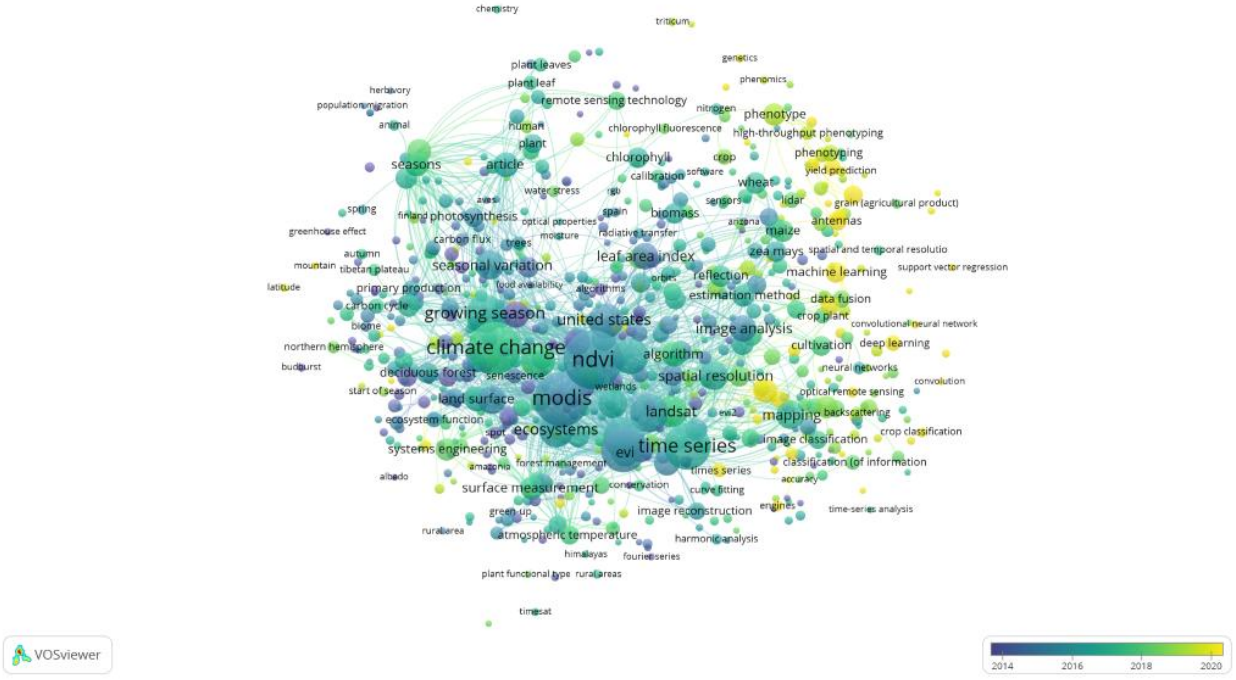

Figure 12. Keyword co-occurrence with overlay projection

Figure 11 and Figure 12 display the same network but in accordance with temporal trends in phrases and significant research areas respectively. The blue and purple clusters have MODIS and NDVI among the oldest trends since 2014 which indicates that MODIS satellite data and NDVI have been extensively used in phenology studies. This can also be confirmed by the density projection in Figure 12. We observe that climate change and seasonal variation have gained

prominence in phenology research in recent times. The most recent trend within this domain is focussed on the use of machine learning and deep learning techniques.

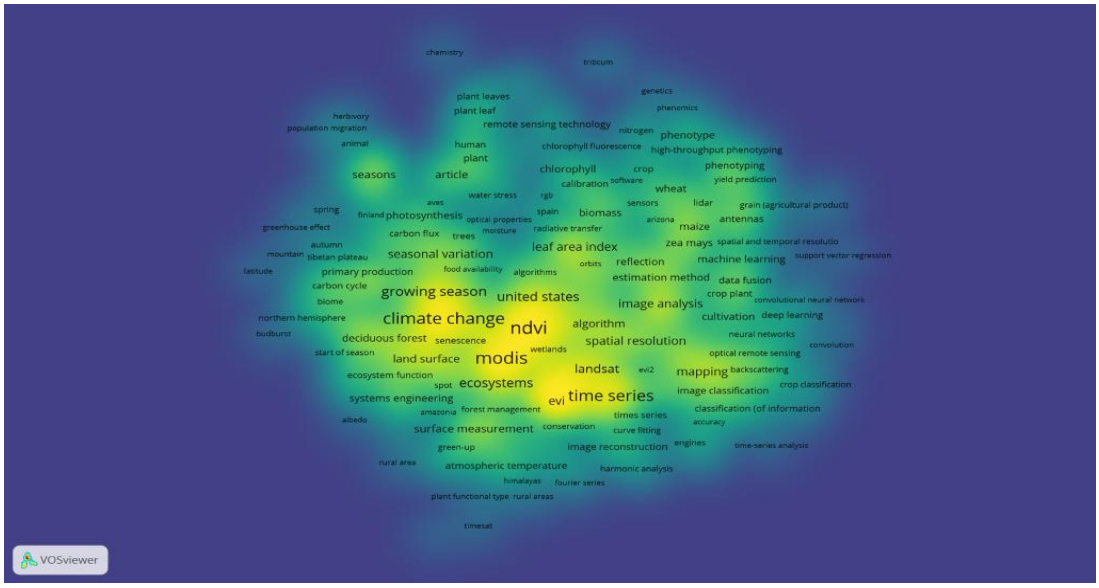

Figure 13. Keyword co-occurrence with density projection

BQ2b: Have there been studies conducted to explore the effect of climate change to vegetation phenology?

The bibliometric analysis implemented above provides performance measures that help outline the diverse contributions made to the research sector. We classified the corpus of 1206 documents based on their relevance to the effects of climate change to vegetation phenology and its contribution to the studies of seasonal variations. We skimmed through the title and abstracts of those 1206 documents to determine its relevance with the research topic. An article was determined relevant if its title or abstract contained "climate change" and "seasonal variation" phrases. Only 5 of those documents were found relevant and if we included keywords in our search, then a total of 42 documents was determined relevant. We sorted the list based on highest citations to review the top 5 cited articles in the research field. A literature review of those articles is presented in Table 5 summarizing the title, author, year of publication, source title, objective of the article pertaining to the thesis topic and gaps in the research to identify existing research gaps in the literature of the thesis topic.

Table 5. Review of articles from bibliometric analysis related to thesis topic

#	Article Title	Author(s) (Year)	Source Title	Objective of article as it pertains to thesis topic	Research Gaps
1	Widespread Climate Change in the Himalayas and Associated Changes in Local Ecosystems	Shrestha et al. (2012)	PLoS ONE	To study the significant changes in temperature, rainfall, and vegetation phenology across the Himalayas by analyzing temperature, rainfall and NDVI from remotely sensed imagery.	Does not conduct analysis on the seasonal variation of meteorological data used in the study.
2	Large seasonal swings in leaf area of Amazon rainforests	Myneni et al. (2007)	Proceedings of the National Academy of Sciences of the United States of America	To study seasonal swings in leaf area and its correlation with solar radiation and precipitation	Lack of robust validation strategy available for leaf area seasonality
3	Plant performance in a warmer world: General responses of plants from cold, northem biomes and the importance of winter and spring events	Aerts et al. (2006)	Plant Ecology	To review the response of plant phenology, changes in their range and distribution, soil nutrient availability and the effects on the structure and dynamics of plant communities by climate warming	Lacks research on the effects of changes in winter climate
4	Understanding spatio-temporal variation of vegetation phenology and rainfall seasonality in the monsoon Southeast Asia	Suepa et al. (2016)	Environmental Research	To quantify phenological patterns and trends in the monsoon southeast Asia and assess their relationship with climate change in the region	Lack of consideration of meteorological features other than rainfall in phenological change as a response to climate change
5	Alpine vegetation phenology dynamic over 16 years and its covariation with climate in a semi-arid region of China	Zhou et al. (2016)	The Science of the Total Environment	To analyze spatial and temporal patterns of SOS, EOS, growing season length (GSL) and maximum NDVI and its correlation with climatic factors	Lack of use of soil data such as soil temperature and moisture. Found differences in results due to differences in remote sensing data, climate data time series and spatial levels of studies.

Step 5. Discuss the results

From the 1206 articles reviewed, only a few articles were determined to have similar research topics and contributions. We found a lack of publications on mapping vegetation phenology changes to seasonal variation in response to climate change (Table 5). This can also be confirmed by the co-occurrence distance between the studies on vegetation phenology and seasonal variation.

Chapter 3: Research Methodology

This chapter presents the comprehensive methodological framework developed for utilizing Remote Sensing Data (RSD) in monitoring vegetation phenology. Following the guidance of Khaiter and Erechtchoukova (2022), the methodology integrates essential stages, including data acquisition, preprocessing, and analysis. Depending on the project's goals, it also extends to modeling, optimization, and decision-making phases, ensuring a robust and goal-oriented approach to RSD analysis.

The previous chapter highlighted critical research gaps that shape the methodological requirements for this study. These gaps inform the structure and design of the methods outlined in this chapter, ensuring that they directly address the research questions and contribute to advancing phenology detection methods.

Guided by the research contributions discussed in the introduction, this chapter is organized as follows: Section 3.1 details the study area, motivation behind the choice of research variables and data extraction; Section 3.2 describes the dataset construction and preprocessing for robust model training and validation; Section 3.3 elaborates on the design and training of the novel proposed Temporal Multivariate Attention Network (TMANet) for phenology detection along with its comparison with baseline models; and Section 3.4 discusses the model evaluation metrices and assessment framework for evaluating TMANet's effectiveness in identifying climate vulnerabilities, thus supporting sustainable practice. Finally, Section 3.5 provides a summary of the experimentation design offering a consolidated framework in line with research contributions of this study.

3.1: Data Collection

It is essential to note that the data collection process begins only after defining the type of vegetation to be monitored. In this thesis, we focus on *corn* ecosystems due to their economic importance as a staple grain and responsiveness to climatic and environmental conditions, which makes them a suitable subject for phenological studies. Corn’s growth stages are well-documented and have distinct phenological transitions, making it ideal for remote sensing-based phenology detection and validation.

3.1.1: Region of Study

In this research, we are considering the corn-growing regions in the state of Iowa, USA (40˚35´N to 43˚30´N and 90°8'W to 96°26'W), contributing 17% of total corn produced nationally, as per the 2022 reports published by U.S Department of Agriculture (USDA). In general, Iowa has a humid continental climate characterized by warm summers and cold winters. During the growing season, typically from May to October, the average temperature ranges between 20°C and 30°C. The precipitation is relatively well distributed during the growing season, averaging between 30 to 36 inches (760 to 915 mm) annually. Iowa's diverse topography, ranging from rolling hills to flat plains, and its temperate climate provide an ideal environment for corn cultivation, which accounts for 50-60% of total cropland in the state. This state is divided into nine agricultural statistics districts (ASDs) according to geography, farming practices and climatic conditions (Figure 14), each comprising a group of counties. This research is developed and tested on datasets derived from three experimental study sites located in Clay, Fayette and Story counties of northeast, northwest and central districts of Iowa, respectively.

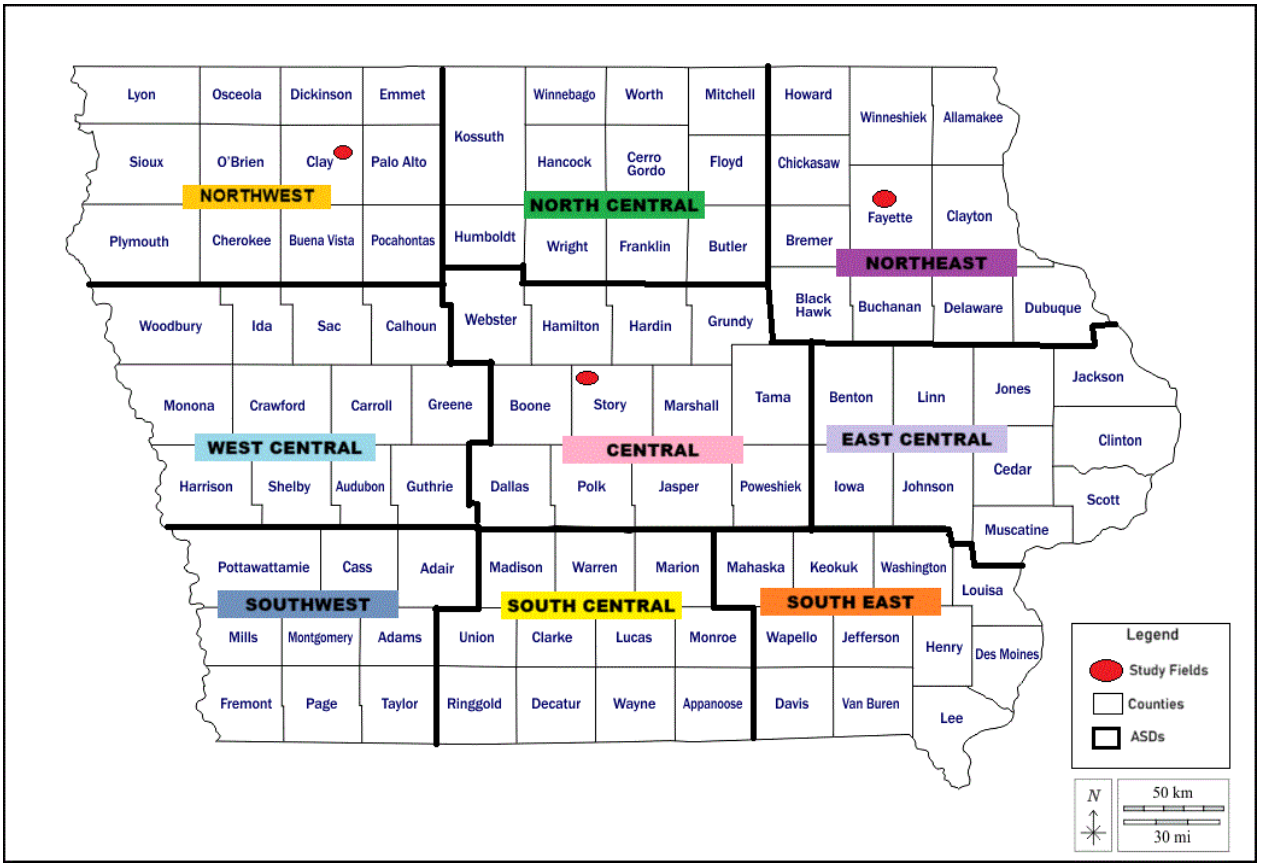

Figure 14. Agricultural statistics districts of Iowa, USA, highlighting study sites.

3.1.2: Study Period and Data Frequency

The data collection period spans from 2000 to 2022, inclusive. This range was selected based on the availability of research variables from reliable data sources, ensuring comprehensive coverage over more than two decades. The chosen timeframe is sufficient to capture interannual variability in corn phenology, allowing for an in-depth examination of how phenological stages respond to varying climatic conditions over time. Additionally, this period facilitates the observation of potential trends and patterns in phenology, which are essential for understanding the impacts of climate change on corn development. Daily data was retrieved to ensure high temporal resolution, enabling precise tracking of phenological transitions critical for accurate analysis.

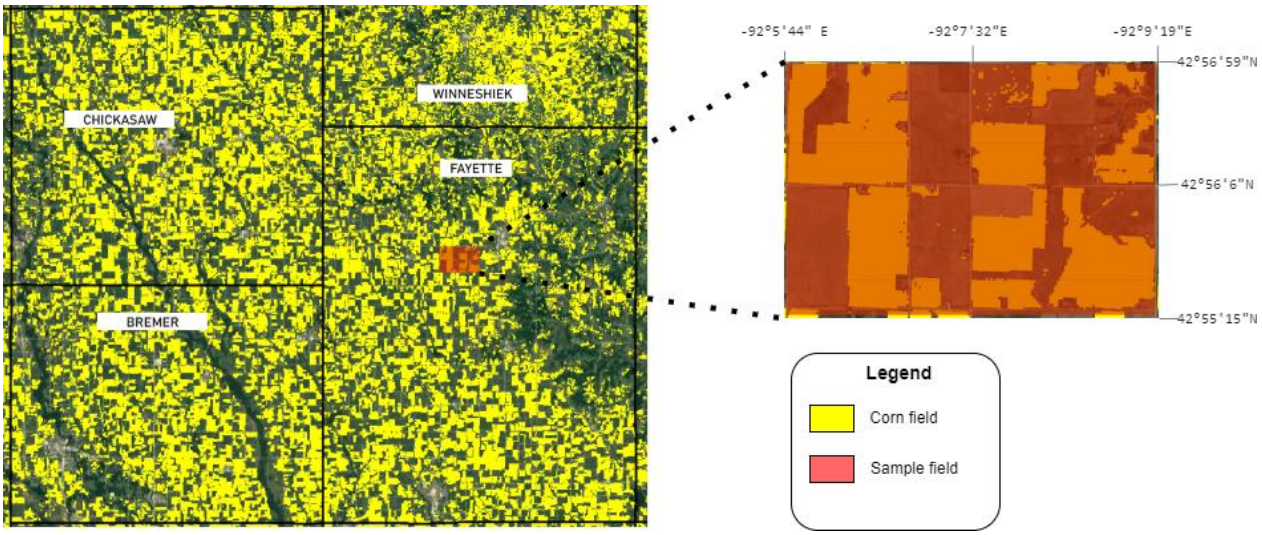

Figure 15. Location of the study site in Fayette County; the spatial distribution of corn and sample field based on NASS Cropland data layer (CDL) crop mask information for the year 2022

3.1.3: Corn Acreage Data

While retrieving corn related data, it is essential to identify fields dedicated to corn production each year, as corn is typically grown within crop rotation systems rather than on the same fields annually (Al-Kaisi & Kwaw-Mensah, 2020). Corn acreage data was obtained from the USDA NASS Cropland Data Layers (CDL) (NASS CDL, 2024). This data was used to delineate the spatial extent of corn cultivation zones across Iowa each year to get pure corn pixels, allowing for the accurate spatial mapping of phenological stages. The CDL provides annual classification of crop types at a 30m resolution, enabling precise identification of corn fields, and has been widely used as cropland reference data in numerous agricultural studies (Diao, 2020; Yang et al., 2021; Zeng et al., 2016).

For this study, sample field areas of approximately 25 square kilometers were chosen in each county to ensure representative coverage. A sample field in Fayette County investigated in this study is depicted in Figure 15.

Table 6. Description of input data in the study

Model Input	Definition	Parameters	Properties	Source
NDVI	$\frac{NIR - RED}{NIR + RED}$	NIR = near infrared band; RED = red band	Produced **daily** using 16 days of Terra and Aqua MODIS data at **500-meter** resolution	NASA's MODIS Nadir Bidirectional Reflectance Distribution Function Adjusted Reflectance (NBAR) data, product MCD43A4, version 6.1 (NASA MODIS, 2024)
LSWI	$\frac{NIR - SWIR}{NIR + SWIR}$	SWIR = short-wave infrared band		
GDD	$\frac{T_a + T_b}{2} - 10$	$T_a = T_{min}$, if $10 <= T_{min} <= 30$ $T_a = 30$, if $T_{min} >= 30$ $T_a = 10$, else $T_a <= 10$ $T_b = T_{max}$, if $10 <= T_{max} <= 30$ $T_b = 30$, if $T_{max} >= 30$ $T_b = 10$, else $T_b <= 10$ T_{min} = minimum temperature T_{max} = maximum temperature	Value at **2-meter**	NASA's POWER Project (NASA POWER, 2024)
CHU	$CHU_{min} = 1.8(T_c - 4.4)$ $CHU_{max} = 3.33(T_d - 10) - 0.084(T_d - 10)^2$ $\frac{CHU_{min} + CHU_{max}}{2}$	$T_c = T_{min}$, if $T_{min} > 4.4$ $T_c = 4.4$, if $T_{min} < 4.4$ $T_d = T_{max}$, if $T_{max} > 10$ $T_d = 10$, if $T_{max} < 10$ T_{min} = minimum temperature T_{max} = maximum temperature		

3.1.4: Study Variables and Data Sources – a Motivation

To effectively monitor corn phenology, we selected a combination of remote sensing and climatic variables as described in Table 6, the NDVI, LSWI, along with heat sum indices such as CHU and GDD. NDVI and LSWI were chosen as remote sensing indicators due to their sensitivity

to changes in vegetation health, biomass, and water content, which are crucial for detecting phenological stages. CHU and GDD provide accumulated thermal information, essential for understanding the temperature-driven growth progression of corn. The integration of these variables would offer an in-depth view of corn phenology by capturing both environmental and physiological factors that influence growth stages.

3.1.4.1: Remote Sensing Data

NDVI and LSWI are calculated using NIR, Red and SWIR bands as detailed in Table 6. The source data for daily NDVI and LSWI time-series were the MODIS Nadir Bidirectional Reflectance Distribution Function Adjusted Reflectance (NBAR) data (MCD43A4, version 6.1), produced daily at a spatial resolution of 500m based on a 16-day retrieval period (NASA MODIS, 2024).

3.1.4.2: Climate Data

Daily temperature values (minimum and maximum) for each U.S county included in this study were downloaded from the NASA POWER project dataset (NASA POWER, 2024). They are used to calculate the thermal indices; GDD and CHU, as per Table 6.

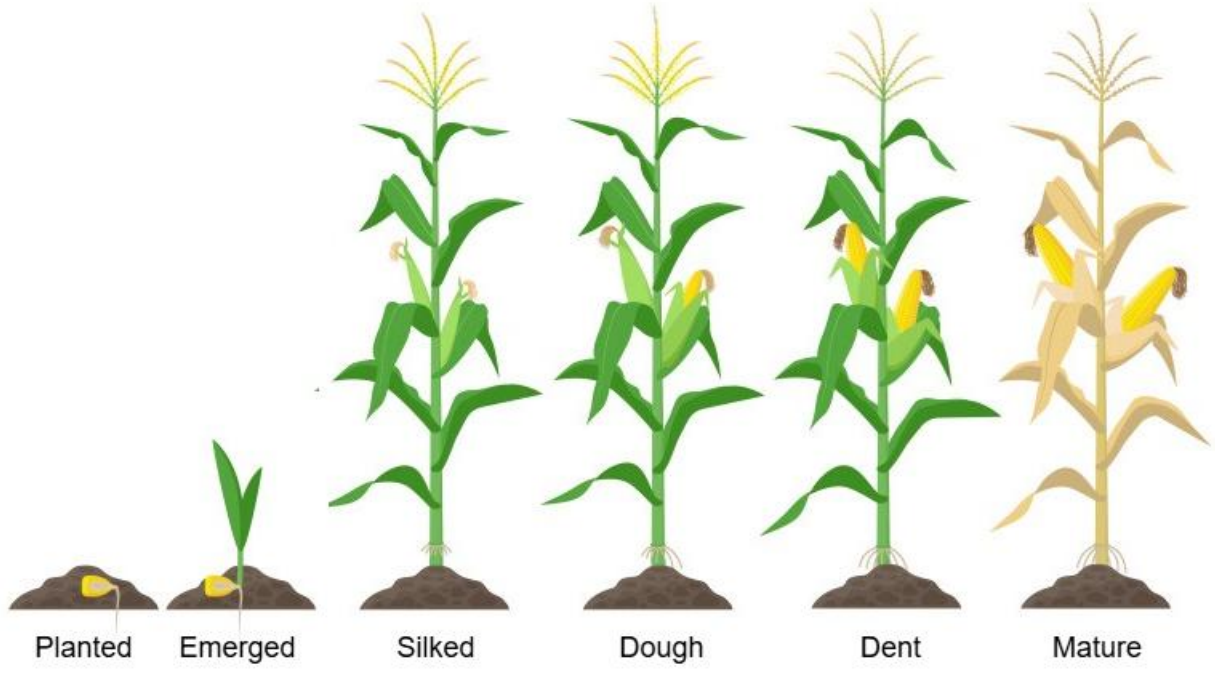

Figure 16. Key growth stages of corn recorded by USDA

3.1.4.3: Crop Progress Data – Ground Truth

The Iowa State University (ISU) classification of corn development divides the growth stages into two main phases: vegetative (V) and reproductive (R) stages (Ritchie et al., 1989). The description of each corn phenological stage is given in Table 7. This approach provides a detailed understanding of corn development, covering each key stage from planting to maturity.

Table 7. Growth and development stages in corn

Vegetative Stages		**Reproductive Stages**	
VE	Emergence – corn seedlings emerge from the soil.	R1	Silking - The emergence of the silks from the corn ear
V1	First leaf collar – one leaf with collar visible	R2	Blister - kernels appear as small blisters filled with a clear fluid.
V2	Second leaf collar – two leaves with collar visible	R3	Milk – kernels are yellow, filled with a milky white fluid
V3	Third leaf collar – three leaves with collar visible	R4	Dough - kernels start transitioning from a watery consistency to a dough-like texture as starch begins to accumulate
V(n)	Nth leaf collars visible	R5	Dent - kernels develop a dent on the crown as they lose moisture and harden
VT	Tasseling – tassel (male flower) fully emerges, and its last branch is completely visible	R6	Mature - Kernels reach their maximum dry weight, and the plant starts to senesce. The kernels are fully hardened and are about ready to harvest

Crop Progress Reports (CPRs) provided by USDA are the most extensive ground-based data source on crop phenological progress and conditions available to date (NASS CPR, 2024). Issued weekly during the growing season (April to November), CPRs include crop planting progress, crop development and harvesting progress. In Iowa, CPRs have been published at both state and district levels since 1974, providing data on the percentage of corn croplands that have reached specific phenological growth stages across these geographic scales. Ideally, county level

corn phenological data should be used, however district-level is the lowest granularity of corn growth data available to us from USDA-NASS. The CPRs record data specifically for the Planted, Emerged, Silked, Dough, Dent, Mature, and Harvest corn stages. For each district analyzed, the weekly time series of the area percentage of corn reaching the phenological growth stages shown in Figure 16 was obtained from CPR.

3.1.5 Data Extraction

3.1.5.1: Google Earth Engine

Google Earth Engine (GEE) is a cloud-based geospatial processing platform that offers access to an extensive catalog of satellite imagery and geospatial datasets (GEE, 2024). It enables large-scale analysis and visualization of Earth observation data, leveraging Google's cloud computing resources to process vast amounts of data efficiently. GEE comprehensive data catalog includes satellite imagery from sources such as MODIS, Landsat, and Sentinel, along with global geospatial datasets on climate, land cover, and vegetation indices, all accessible to the public (GEE Data Catalog, 2024).

For this study, GEE was essential in calculating and extracting NDVI and LSWI values specifically for corn-growing fields in Iowa. The detailed extraction process is illustrated in Figure 17. The JavaScript programming language was used to interact with the GEE web-based interface, facilitating the definition of the study area and enabling data manipulation.

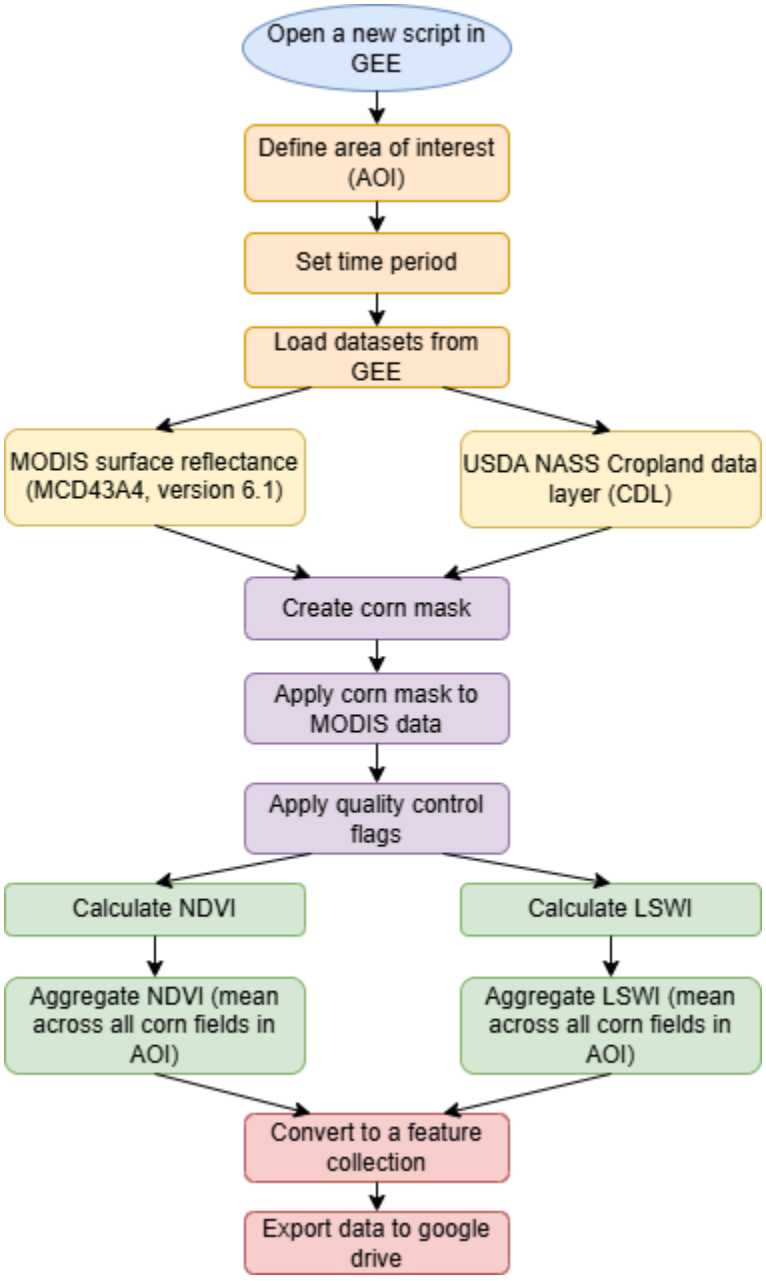

Figure 17. Flowchart of the extraction process of the NDVI and LSWI values on the GEE

At the beginning, the Geometry Drawing Tools in GEE's code editor were used to define the study area on the GEE map interface, focusing on selected districts within Iowa. Given the year-to-year variation in specific fields designated for corn cultivation, it was necessary to repeat this process for each growing season. Therefore, the code was run iteratively across 23 loops for each investigated district (one per each year from 2000 to 2023) to extract NDVI and LSWI values exclusively for corn fields cultivated each year in our study region.

Two key datasets were utilized in this process, both available in the GEE Data Catalog: the MODIS Surface Reflectance dataset and the USDA Cropland Data Layer (CDL). The MODIS dataset provides daily surface reflectance data, while the CDL offers annual land cover classifications. The CDL dataset was applied to generate a corn mask, isolating corn-growing areas within Iowa. This mask filtered out non-corn areas, focusing the analysis on the relevant land cover type. The CDLs were geographically downscaled to match the coarser MODIS resolution to retrieve surface reflectance values of pure corn field pixels.

After applying the corn mask, the filtered MODIS surface reflectance data was further refined by applying quality assurance bands to disguising cloud and snow-covered pixels, ensuring the use of only high-quality surface reflectance data for further analysis. Based on these refined surface reflectance values, NDVI and LSWI were calculated according to the formulas provided in Table 56, specifically for pixels representing pure corn areas. The computed values were then averaged across all corn fields within the defined region to provide a representative summary of vegetation health and moisture levels. The processed and aggregated daily NDVI and LSWI data were subsequently exported to Google Drive, making them readily accessible for further analysis and experimentation in the thesis.

3.1.5.2: NASA POWER Data Access Viewer (DAV)

NASA's POWER Data Access Viewer (DAV) web application (NASA POWER, 2024) was utilized to retrieve the maximum and minimum temperature data needed for calculating Crop Heat Units (CHU) and Growing Degree Days (GDD). The DAV interface enables precise data extraction over a specified period and area of interest, as depicted in Figure 18. Using the "Regional" tab in the DAV, we specified latitude and longitude coordinates that matched the study area boundaries set in Google Earth Engine (GEE) for each district in Iowa.

The temporal range was set from January 1, 2000 to December 31, 2022, with a daily resolution to capture consistent temperature data across the full study period. For each district, the daily maximum and minimum temperatures were selected under the parameters "Temperature at 2 Meters Maximum" and "Temperature at 2 Meters Minimum", respectively. The application

allowed downloading data in multiple formats; CSV was chosen for ease of integration with other tools.

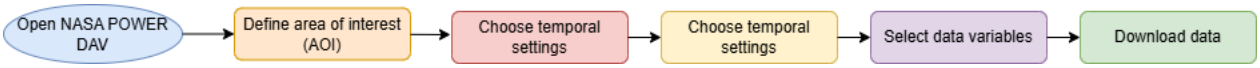

Figure 18. Flowchart of the process of extraction of minimum and maximum temperature values from NASA POWER DAV

The downloaded CSV files were then imported into Microsoft Access for data review and processing. Within this environment, formulas were applied to compute the GDD and CHU values as per the equations outlined in Table 6. This streamlined workflow facilitated the efficient calculation of daily CHU and GDD values across the 23-year study period.

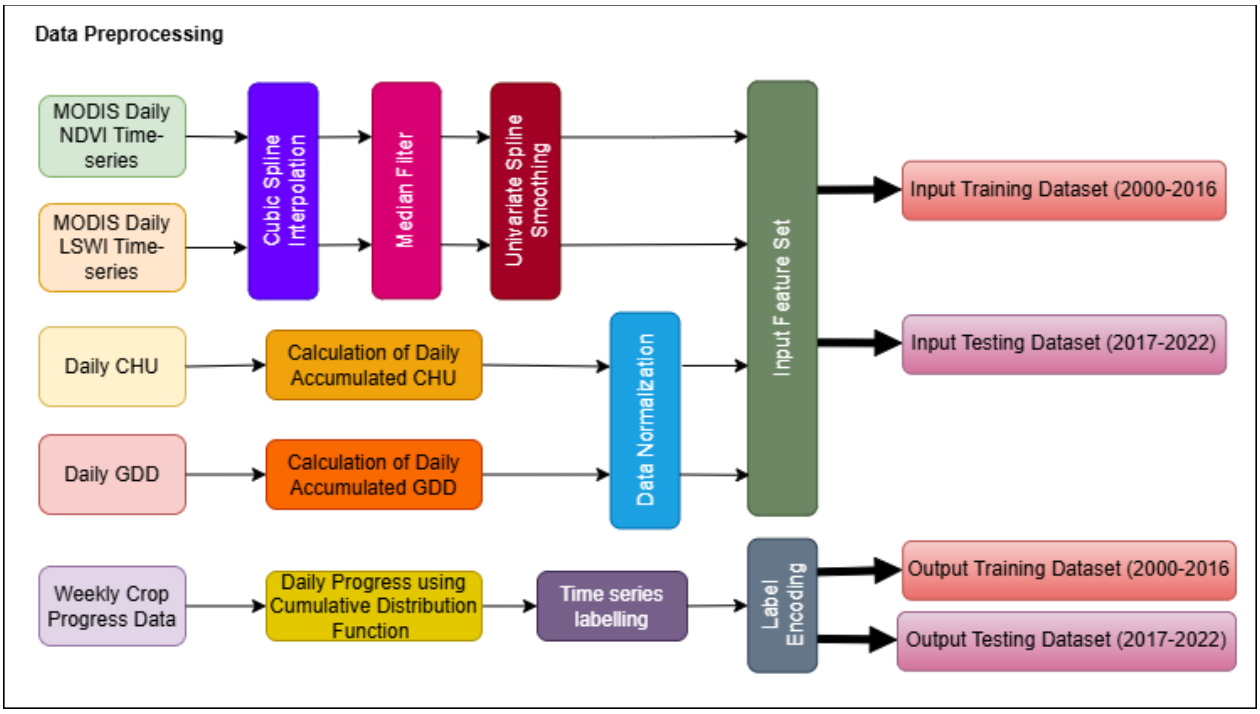

Figure 19. Block diagram for data preprocessing phase.

3.2: Dataset Construction and Preprocessing

This section outlines the methodology used to construct a robust dataset for training the phenology monitoring model. Python programming language was used for all data preprocessing and modeling tasks, with code written and executed in Jupyter Notebook and Google Colaboratory. Jupyter Notebook, a powerful open-source web application, enables interactive coding and visualization, while Google Colaboratory (Colab) provides an online environment with free access

to cloud-based computational resources, making it ideal for large-scale data processing and model development. Key preprocessing steps, illustrated in Figure 19, were applied to clean and standardize the study variables, ensuring consistency across the dataset. These steps involved transforming diverse input variables into a unified feature set compatible with the model's requirements. Finally, the dataset was divided into training and validation sets to facilitate reliable model evaluation and testing.

3.2.1: Data Cleaning and Transformation

3.2.1.1: Corn growth stage labelling

To obtain a daily record of corn growth progress, the weekly corn progress data was linearly interpolated using a cumulative distribution curve-fitting method. This approach provided a smooth, continuous curve that ensured the cumulative percentages followed a monotonically increasing trend, accurately modeling the gradual progression of corn through its growth stages. For labeling purposes, it was assumed that corn in a given sample field reaches a specific growth stage when 50% of the corn crops in the district are reported to have attained that stage.

Based on this interpolated daily time series, each calendar date was assigned a growth stage label corresponding to when approximately 50% of the crops had reached that stage. Each stage label continued until the date when 50% or more of the crops transitioned to the next phenological stage. An example of weekly and interpolated daily corn growth progress for the year 2000 is shown in Figure 20 while Figure 21 presents the assignment of Days of the Year (DOY) to their respective growth stages.

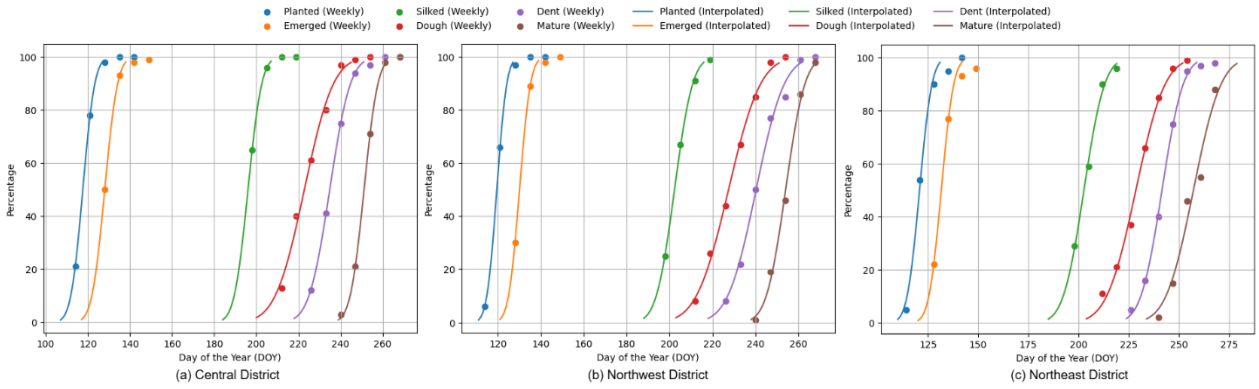

Figure 20. Corn growing progress (weekly and interpolated) in the year 2000 in (a) Central (b) Northwest and (c) Northeast districts of Iowa

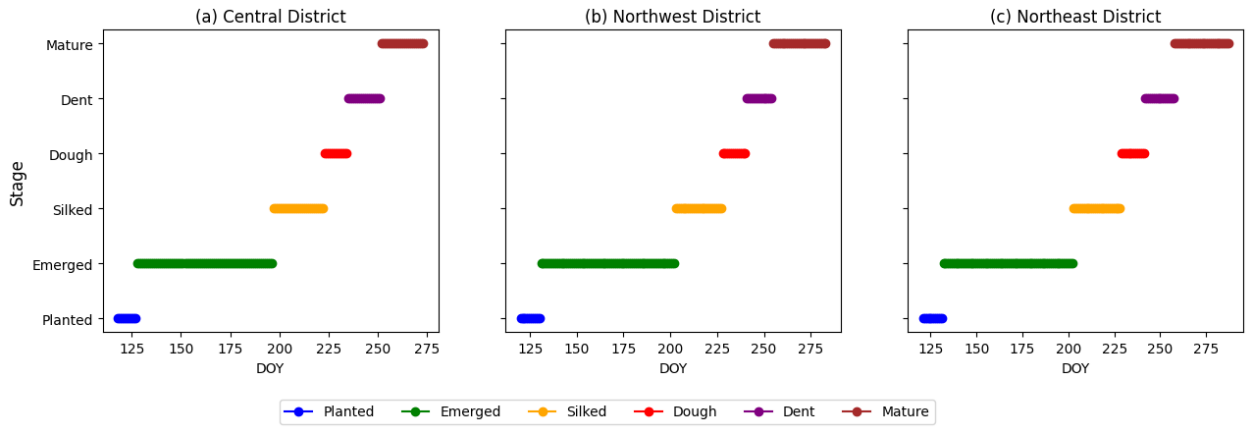

Figure 21. Example of corn growth stage labelling w.r.t DOY for the year 2000

For each year (2000–2022), the growing period was defined as the timeframe from planting to harvesting, ensuring that the analysis captured the crop's active growth phases. Figure 22 shows the total number of samples available for each corn growth stage across the districts studied during the analysis period.

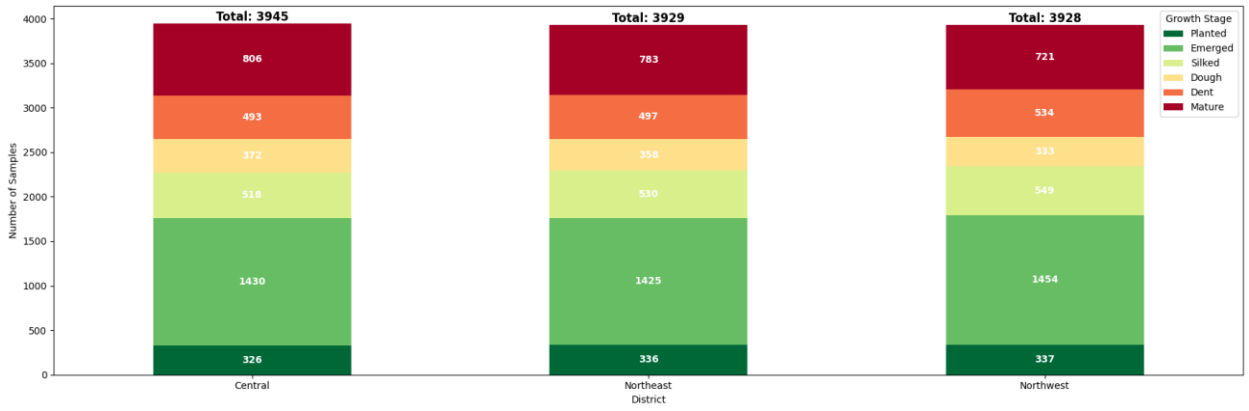

Figure 22. The stacked bar chart shows the total number of samples collected for each corn growth stage across the Northwest, Northeast, and Central districts of Iowa from 2000-2022.

3.2.1.2: Construction of Dataset

Crop growth, particularly for corn, progresses through developmental stages that are driven by cumulative exposure to warmth, as plants rely on accumulated thermal energy to regulate their growth processes (Nielsen, 2002; McMaster & Wilhelm, 1997). Since plant development is cumulative in nature, it integrates past environmental conditions to determine each stage of growth. To capture this progression, we calculated the daily accumulations of GDD and CHU by applying a cumulative sum of the daily values across each day within the growing season. The temperature

observations from NASA POWER were complete, with no missing readings, ensuring that we had uninterrupted daily GDD and CHU values. Figures 23 and 24 depict the cumulative GDD vs DOY and cumulative CHU vs DOY curves for the year 2000, respectively. To standardize the features for model training, accumulated GDD and CHU values were normalized using Z-score standardization, centering the data around a mean of 0 with a standard deviation of 1. This normalization mitigates the impact of features with larger magnitudes, ensuring balanced learning and reducing the likelihood of gradient issues during deep learning model training.

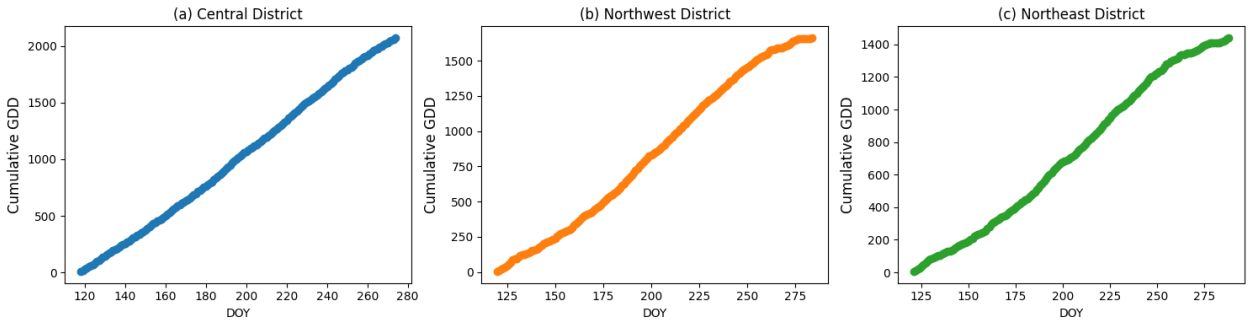

Figure 23. Cumulative GDD vs DOY curve for the year 2000

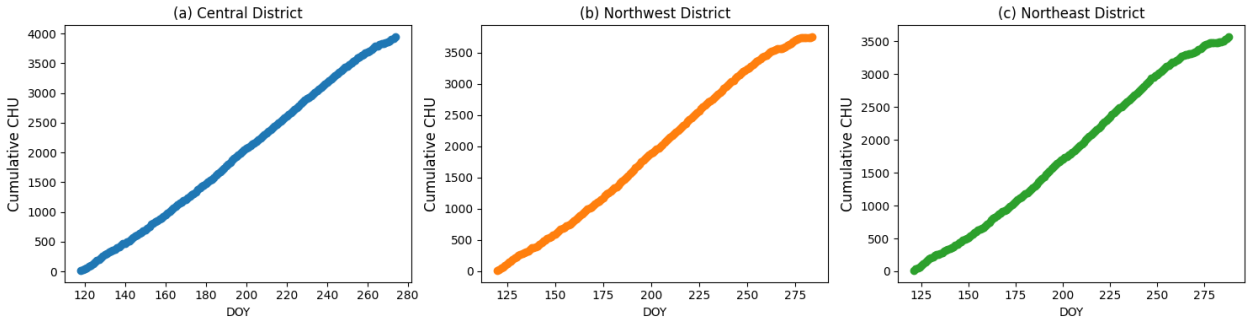

Figure 24. Cumulative CHU vs DOY curve for the year 2000

Satellite-derived NDVI and LSWI time series were extracted to align with the identified growing periods. These time series, however, contained missing values (Table 8) and spurious readings due to varying illumination, atmospheric interference, and sensor noise, which could obscure the actual phenological patterns.

Table 8. Number of missing observations in study variables

Feature	Central District	Northwest District	Northeast District
NDVI	21	33	34

LSWI	24	31	38
Cumulative GDD	0	0	0
Cumulative CHU	0	0	0

To address these issues, several preprocessing techniques were applied. Cubic spline interpolation was used to estimate missing NDVI and LSWI values by fitting smooth, piecewise third-degree polynomials between known data points, ensuring continuity in the first and second derivatives. This method preserved the gradual ecological processes without introducing abrupt changes or overfitting (Li et al., 2021; Vorobiova & Chernov, 2017). Median filtering was employed to remove anomalous values caused by adverse weather conditions, replacing outliers with the median value within a five-day temporal moving window (Diao, 2020). Finally, a smoothing spline (with a smoothing factor of 1) was applied to further refine the time series, enhancing smoothness while maintaining the integrity of the vegetation index trends (Figures 25 and 26). NDVI and LSWI are inherently normalized indices, calculated as ratios of different spectral bands. This inherent normalization means they did not require additional standardization, as their values are already scaled to capture relative vegetation characteristics without the influence of differing units or magnitudes.

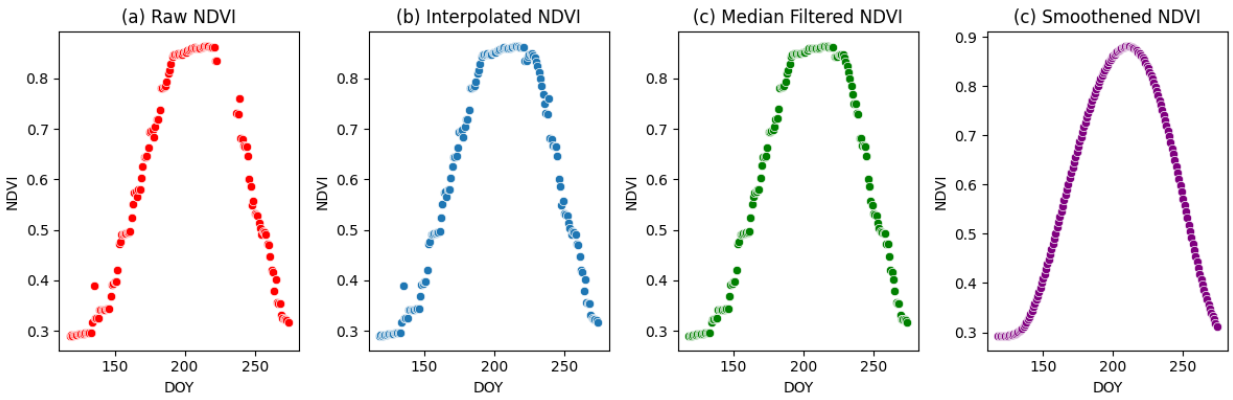

Figure 25. Preprocessing of raw NDVI time series for Central Iowa, 2000

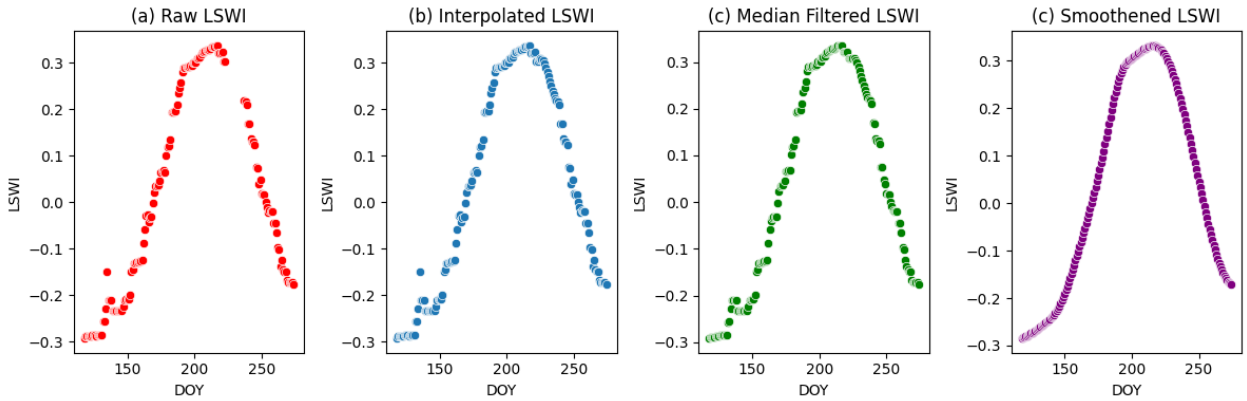

Figure 26. Preprocessing of raw LSWI time series for Central Iowa, 2000

Subsequently, Cumulative GDD, Cumulative CHU, NDVI, and LSWI after preprocessing steps were consolidated into a single table alongside a year, DOY and corresponding growth stage labels to create a comprehensive dataset for ML modeling aimed at classifying corn growth stages. The growth stages were label-encoded for the classification task, transforming categorical stages into numerical values that the model could interpret effectively. This encoding step is critical, as it converts each phenological stage into a distinct integer, enabling the model to identify patterns specific to each stage and improving classification accuracy. These preprocessing measures ensured spatial and temporal consistency across the dataset, providing a robust foundation for the model to learn complex relationships and enhance performance in predicting corn growth stages.

3.2.2: Dataset Structuring

The final preprocessed dataset (obtained in section 3.2.1) was divided into training (2000–2016) and testing (2017–2022) sets, with approximately 75% of the data used for model training and 25% reserved for testing. Figure 27 shows the distribution of training and testing samples across the various corn growth stages for each of the studied districts, highlighting the balance of data used for model development and evaluation. Within both the training and testing sets, the data was further organized into input and output subsets. The input training and testing sets contained the independent variables used to predict growth stages—namely, LSWI, NDVI, cumulative GDD,

and cumulative CHU. The output sets contained the target variable, which in this case was the corn growth stage to be predicted through ML modeling.

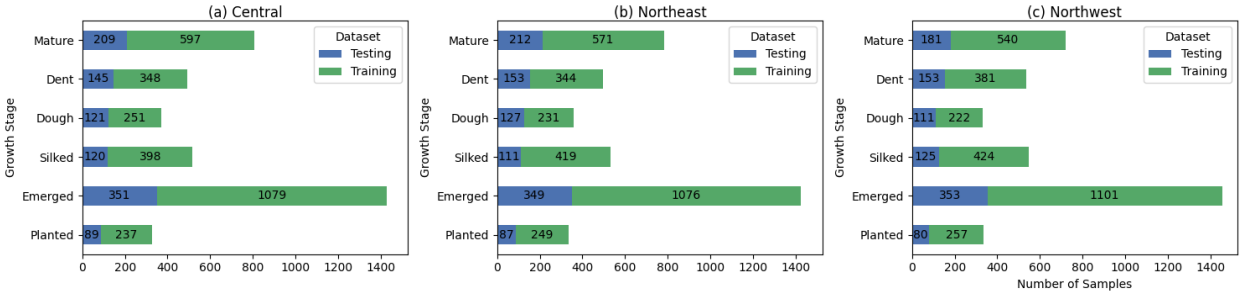

Figure 27. Distribution of training and testing samples across corn growth stages for (a) Central (b) Northeast (c) Northwest districts of Iowa

3.3: Proposed Model - Temporal Multivariate Attention Network (TMANet)

Originally introduced for natural language translation by Bahdanau et al. (2015), attention-based neural networks have rapidly evolved into essential tools across a broad range of domains, including time-series classification (Chen & Shi, 2021; Tripathi & Baruah, 2020; Karim et al., 2017). In this work, we propose a novel deep learning model— TMANet, designed to effectively capture both temporal dependencies and complex feature interactions from integrated agro-climatic and vegetation indices, enabling precise detection of crop growth stages. TMANet leverages both temporal and multivariate attention mechanisms to enhance the model's ability to detect crucial patterns in the data, ensuring more accurate predictions and interpretability.

The proposed model:

- Integrates multivariate attention to capture feature-level interactions across multiple variables
- Leverages temporal attention mechanism, enabling the model to concentrate on significant time steps, enhancing its ability to capture and learn long-range temporal dependencies across extensive historical datasets.
- Is computationally efficient and scalable designed to handle large time-series datasets from satellite observations and meteorological sources, making it applicable across different regions and crops.
- Offers interpretable results, with multivariate and temporal attention mechanisms providing insights into both feature importance and key time steps that influence growth stages.

3.3.1: Model Architecture

In this section, we introduce the proposed TMANet model and analyze its computational complexity. Unlike prior models (Yang et al., 2023; Zhao et al., 2022), TMANet integrates multivariate feature attention and temporal attention mechanisms within a unified architecture to simultaneously learn through significant features and relevant timesteps. TMANet has an attention-augmented bidirectional LSTM architecture, specifically designed for multivariate time series classification tasks.

Given a multivariate timeseries input with F features, denoted as $X = [x_1, x_2, \dots, x_F]^T \in R^{F \times T_{in}}$, where T_{in} is the total length of the input sequence, let the compact form of feature representation at time step $t \in [1, T_{in}]$ be denoted as $X_t = [x_{1t}, x_{2t}, \dots, x_{Ft}]^T \in R^F$. For each output time step $t \in [1, T_{out}]$, the model generates a vector $O_t \in R^C$, where C denotes the number of classes (distinct growth stages). Each element in the vector corresponds to the probability of a certain class (i.e., growth stage) at the given time step. We maintain a one-to-one correspondence between the input and output sequences, such that $T_{in} = T_{out} = T$, ensuring each input step directly predicts the output for the same time step. Specifically, for each time step $t \in [1, T_{in}]$, the model processes the multivariate time series data and predicts the output for the same time step $t \in [1, T_{out}]$.

To provide the model with positional awareness, we apply positional encoding in the input layer (Figure 28). The positional encoding $PE_t = PE(t)$ for each time step $t \in [1, T]$ is defined as:

$$PE(t, 2i) = \sin\left(\frac{t}{10000^{\frac{2i}{F}}}\right),$$

$$PE(t, 2i+1) = \cos\left(\frac{t}{10000^{\frac{2i}{F}}}\right),$$

where i is the index of feature dimension. This encoding provides each time step with a unique vector, embedding both short- and long-term dependencies through the varying frequencies of the *sin* and *cos* terms (Ashish, 2017). The positional encoding vector is added to the input feature representation at time step t forming, $X'_t = X_t + PE_t$, which enriches the input with temporal information.

Next, the temporally encoded input is passed to the Multivariate Feature Attention Layer, which allows the model to focus on the most relevant features at each time step. In this layer, Global Average Pooling (GAP) is applied to the multi-dimensional feature map to transform in one-dimensional feature vector representing the average activation for each feature across time.

Formula for GAP over timesteps T:

$$GAP(X'_f) = \frac{1}{T}\sum_{t=1}^{T} X'_{t,f},$$

where X'_f is the average feature f over all timesteps. The attention weights, which represent the significance of each feature relative to others, are computed using a dense (fully connected) with *softmax* activation applied to the pooled feature vector:

$$\alpha_f = \frac{\exp(h_f)}{\sum_{f=1}^{F} \exp(h_f)},$$

where h_f is the output of the Dense layer for feature f and α_f is its attention weights. The learned attention weights are applied to the input feature via element-wise multiplication scaling each feature of the input sequence by its corresponding attention weight, effectively focusing the model on the most important features. The output of multivariate attention layer is sequence G calculated as follows:

$$G_t = X'_{t,f} \times \alpha_f$$

Next, a Bidirectional Long Short-Term Memory (BiLSTM) layer is employed to capture both past and future dependencies in the time series. This layer enhances temporal feature extraction by considering information in both forward and backward directions, improving contextual understanding. LSTM uses a gated mechanism (input gate, output gate and forget gate) to control the flow, storage and dependency of information over time, addressing issues like vanishing and exploding gradients that affect traditional RNNs (Bandara et al., 2020; Suleman & Shridevi, 2022). The BiLSTM layer reads the input sequences and generates a sequence of hidden states after performing a series of mathematical calculations at each gate. The final hidden state for each time step t and S hidden states is obtained by concatenating the forward and backward hidden states:

$$h_t = \left[h_t^{fwd}, h_t^{bwd}\right],$$

where $h_t^{fwd} \in R^S$ and $h_t^{bwd} \in R^S$ represent the forward and backward hidden states, respectively, and $h_t \in R^{2S}$ is their concatenation.

The temporal layer is designed to focus on most relevant time steps in the sequence by applying attention over the time dimension. This layer helps the model assign attention weights to time steps, focusing on periods with stronger influence on final predictions (e.g., rapid NDVI changes during the growth stage transitions). The attention weights for time step t are computed as:

$$\alpha_t = Softmax(W_t h_t),$$

where W_t is a learnable weight matrix. The context vector $\hat{C}_t$, summarizing important sequence information, is obtained as:

$$\hat{C}_t = \sum_{t=1}^{T} \alpha_t h_t.$$

GAP layer is used at output to avoid overfitting issues (Malla et al., 2023). Finally, the Fully Connected Output Layer converts the context vector into a classification decision. For each time step t, the final output vector O_t is computed as:

$$Z_t = W_z \hat{C}_t + b_z,$$

where $W_z \in R^{2S \times C}$ and $b_z \in R^C$ are weight and bias matrices, respectively. A *Softmax* function is applied to $Z_t \in R^C$ to convert raw scores into a probability distribution over the C classes:

$$O_t = Softmax(Z_t).$$

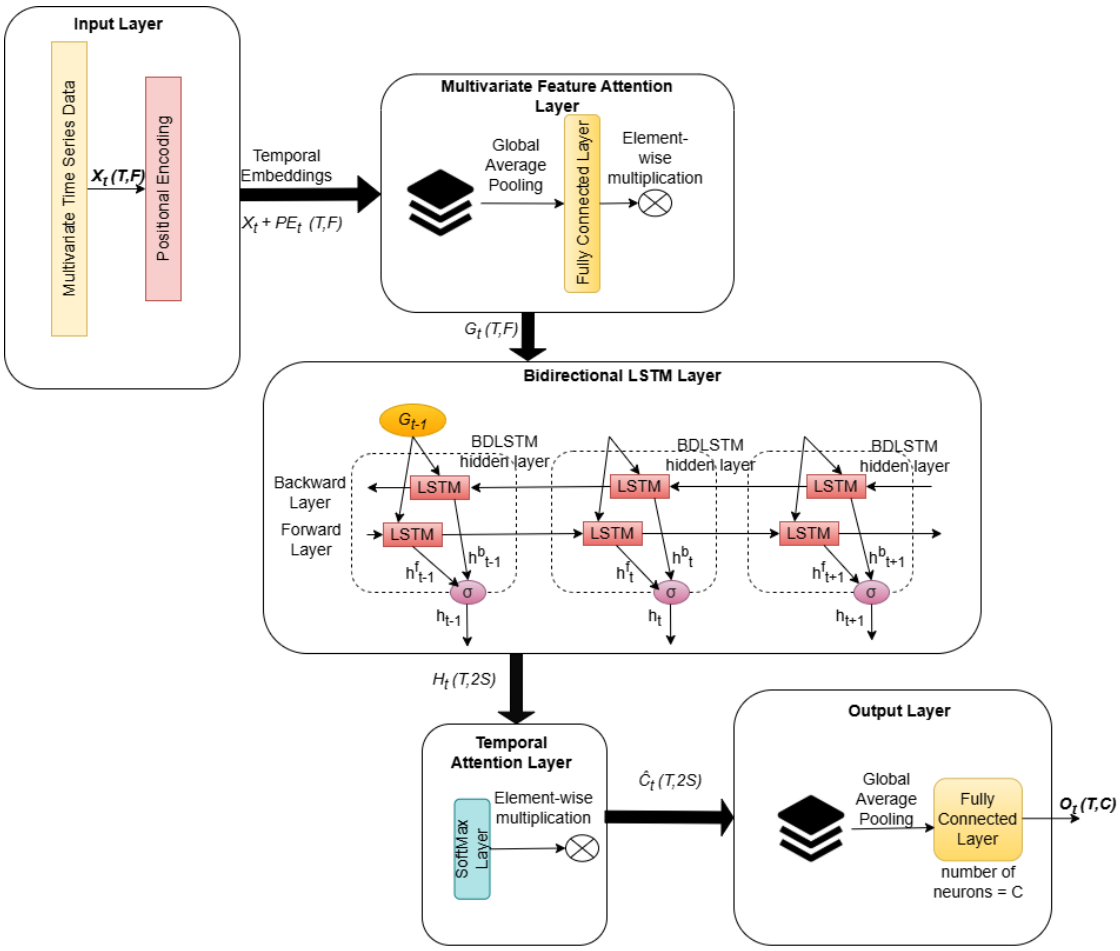

Figure 28. TMANet working architecture to compute output O at time step t.

3.3.2: Model Training and Comparison

TMANet was trained and evaluated on a desktop computer with an Intel Core i7 processor and 32 GB of RAM, using Python 3, TensorFlow (v2.12.0), and open-source Keras interface for ANN. Preprocessed data (as described in Section 3.2) was fed into the model using NumPy and Pandas, while Matplotlib, Seaborn, and Plotly were employed for visualizing model performance.

The model's detailed configuration and hyperparameter tuning are discussed in Section 3.3.2.1 and its comparison with baseline models is presented in Section 3.3.2.2.

3.3.2.1: TMANet Model Settings and Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the best set of hyperparameters for a machine learning or deep learning model to optimize its performance (Liao et al, 2022). Unlike model parameters (which are learned during training), hyperparameters are set before training

begins and control aspects of the model's architecture, training process, and learning behavior. Finding the optimal combination of hyperparameters is critical because they directly impact the model's accuracy, stability, and ability to generalize well to new data.

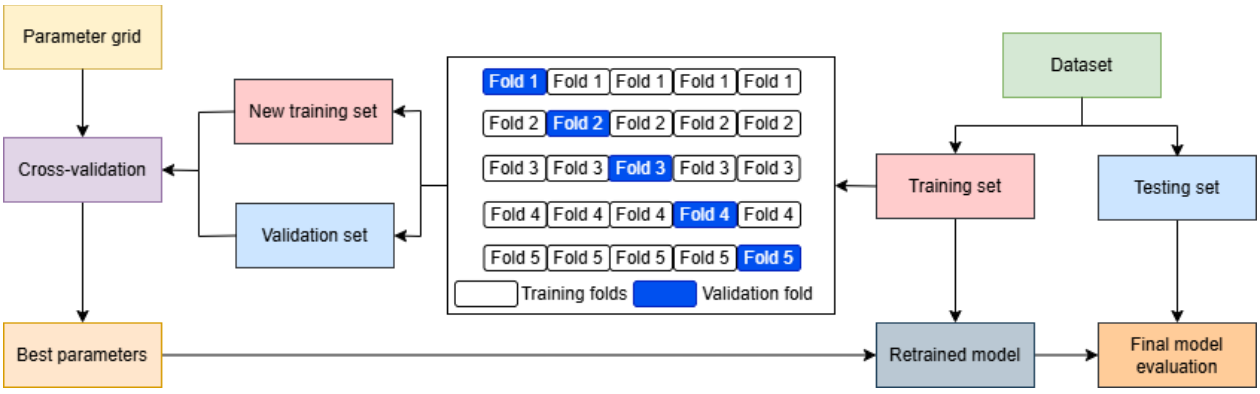

Figure 29. Flowchart of hyperparameter tuning process based on grid search and 5-fold cross-validation.

The TMANet model was optimized through hyperparameter tuning using a grid search with 5-fold cross-validation to ensure robustness. In this process, a grid search systematically explores predefined hyperparameter combinations to identify the optimal settings for model performance. The 5-fold cross-validation further divides the training dataset into five equal subsets, or 'folds,' where the model is trained on four folds and validated on the fifth as depicted in Figure 29. This process is repeated five times, with each fold serving as the validation set once. By averaging the performance across the folds, this approach reduces the risk of overfitting and ensures that the selected hyperparameters perform well across different data subsets, enhancing the robustness and generalizability of the model.

The model was trained for a maximum of 100 epochs, with early stopping based on validation loss. Training would cease if no improvement was observed over 10 consecutive epochs, thus preventing overfitting and saving computational resources. Key hyperparameters were systematically tuned to enhance performance. The batch size, explored in the range of 32 to 128, was set to 64, offering the best trade-off between memory efficiency and accuracy. The learning rate, a critical factor in the optimization process, was tested across values of {1e-1, 1e-2, 1e-3, 1e-4}, with 0.001 proving to be optimal for stable convergence when used with the Adam optimizer. This learning rate minimized loss and prevented overshooting or slow convergence. Additionally, the bidirectional LSTM layer, which handles the temporal dependencies, was optimized with

hidden units ranging from 64 to 256. A configuration of 128 hidden units provided the best results without overcomplicating the model or risking overfitting.

These carefully tuned hyperparameters resulted in a well-optimized TMANet model, which outperformed non-tuned configurations by achieving higher predictive accuracy and generalization capabilities.

3.3.2.1: Comparison with Baseline Models

The performance of the TMANet model was compared against four widely used baseline models: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Support Vector Regression (SVR), and Random Forest (RF). SVR and RF have been extensively used in supervised classification of crop phenological stages (Ramdhani et al., 2020; Murguia-Cozar et al., 2021; Nieto et al., 2021) while LSTM and GRU have shown to achieve state-of-the-art performance in timeseries classification (Karim et al., 2017; Mahjoub et al., 2022; Zainuddin et al., 2021). Each baseline model was tuned using grid search to its optimal configuration to ensure a fair comparison with the proposed TMANet model.

Random Forest (RF) is an ensemble learning method that builds multiple decision trees, combining their predictions to achieve a more robust outcome. It randomly samples data and features to grow each tree, reducing overfitting and improving generalization. The RF was configured with 100 estimators and a maximum tree depth of 10 to balance model complexity and computational efficiency, while the Gini impurity criterion was used to determine split quality. Gini impurity helps determine how well the data at a node is divided between the different classes.

Support Vector Regression (SVR) is a supervised learning model that uses a regression approach to map input features to continuous outputs. By maximizing the margin between data points and a decision boundary, SVR works effectively for continuous phenological stage prediction, employing a radial basis function (RBF) to model non-linear relationships. The regularization parameter (C) controls the trade-off between maximizing the margin (keeping predictions simple) and minimizing the regression error on the training data. It was tuned in the range $\{0.1, 1, 10\}$, with 1 yielding optimal result. A larger value of C places a higher emphasis on minimizing errors, making the model try harder to fit the training data closely.

LSTM was also used as one of the layers in the proposed TMANet model described in section 3.3.1. It was configured with a single layer of 128 hidden units, optimized for performance without overfitting. A dropout rate of 0.2 was applied, and the Adam optimizer with a learning rate of 0.001 was used. Training was carried out for a maximum of 100 epochs, with early stopping after 10 epochs of no improvement in validation loss.

Gated Recurrent Unit (GRU) is another RNN variant similar to LSTM but with a simplified structure, combining the forget and input gates into a single update gate. This reduced complexity allows GRU models to achieve similar performance to LSTM in time series applications while being computationally lighter, making them well-suited for datasets with temporal dependencies. GRU was similarly optimized, with 128 hidden units, a dropout rate of 0.2, and the Adam optimizer (learning rate = 0.001). The model was trained with early stopping based on validation loss, with a batch size of 64 selected for efficient training.

All models were evaluated on the same preprocessed dataset and assessed using consistent performance metrics to ensure a fair and direct comparison with TMANet in terms of predictive accuracy for phenological stage characterization.

3.4: Model Evaluation and Climate Vulnerability Assessment

3.4.1: Model Evaluation Metrics

The predicted corn phenological stages from each model were evaluated against ground-truth data based on two key evaluation criteria: classification performance and the accuracy in detection of phenological transition dates (Figure 30). First, classification performance was assessed using accuracy, precision, recall, and F1 score. Accuracy provides an overall measure of correctly classified growth stages from all the stages in the dataset. Precision and recall offer insights into the relevance and completeness of the classifications. Precision for a given stage is the fraction of instances correctly classified as belonging to the stage out of all the instances the model predicted to belong that stage. Recall is the fraction of instances in a stage that the model correctly classified out of all the instances in that stage. The F1 score, a harmonic mean of precision and recall, ensures a balanced evaluation of the model's effectiveness in identifying each phenological stage.

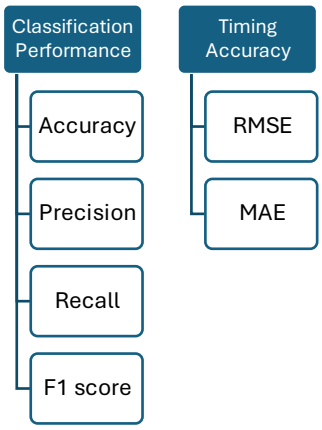

Figure 30. Metrics designed for model evaluation in detecting corn growth stages

The second evaluation focused on the timing accuracy of phenological transitions, which is critical for time-series prediction models. The predicted growth stages must be identified sequentially, adhering to the natural phenological order. For Random Forest and SVR models, post-processing was applied to enforce this sequence constraint. In contrast, LSTM, GRU, and TMANet naturally maintain sequential dependencies due to their design. Sequential constraint was enforced by checking the daily predictions and ensure that once a stage has commenced, any earlier stages cannot occur again. If an earlier stage appears after a later stage, replace it with the later stage to maintain the proper sequence. To assess the accuracy of the predicted phenology transition days, we compared the first occurrence of each predicted stage to the corresponding ground-truth data. This comparison was conducted using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). RMSE captures the overall magnitude of prediction errors, while MAE indicates the average error, providing a comprehensive evaluation of the models' ability to predict the timing of corn phenological stages.

In addition to assessing model accuracy, feature contribution analysis was conducted to uncover the features influencing corn phenology and to improve early-warning capabilities in agricultural risk management. Leveraging feature importance data from the TMANet model, we identified specific drivers critical to corn growth stages. The feature importance was calculated by first multiplying multivariate feature attention weights with temporal attention weights at each time step. These weighted values were then aggregated to determine the overall importance of each feature—NDVI, LSWI, cumulative GDD, and cumulative CHU—across various growth stages, highlighting the primary drivers of phenological transitions.

3.4.2: Climate Vulnerability Analysis

The study included the assessment of changing trends in crop phenology and analyzed its correlation with climate variability during the growing period, focusing on temperature and precipitation. To characterize trends in inter-annual and intra-annual crop phenology, methods such as linear regression analysis and interquartile range (IQR)-based outlier detection were employed. For inter-annual variations, key phenological events, including planting dates, harvesting dates, and growing season length, were analyzed. Planting dates were defined as the DOY marking the first recorded instance of the "Planted" stage, while harvesting dates corresponded to the DOY following the last recorded occurrence of the "Mature" stage. The growing season length was calculated as the difference between the harvesting and planting DOYs. Intra-annual variations were studied by examining linear trends in the durations of individual crop growth stages. Additionally, boxplots were utilized to identify IQR-based outliers in inter-annual phenological data, providing insights into the sensitivity of crop phenology to climatic extremes during the study period.

Regression analysis was used to examine temporal patterns of climatic factors and their relationship with crop phenology. The Pearson correlation coefficient was adopted to describe the sensitivity of crop phenology to climate change. Climatic factors include minimum, maximum, average temperatures, and accumulated precipitation during the crop growing period.

Moreover, stage specific thresholds of agronomic indicators (NDVI, LSWI, cumGDD and cumCHU) used in the study were extracted to identify precise climate conditions required for transitions between growth stages. Non-parametric tests like, Mann-Kendall trend test and Sen's slope estimator were used to analyze trends in these thresholds over time, revealing shifts in the climatic requirements for each phenological stage.

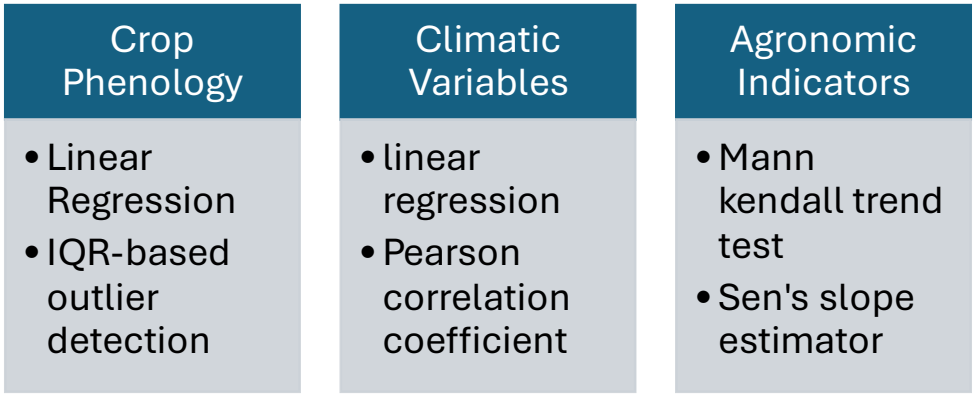

Figure 31. Methods used for assessing climate variabilities and its impact on corn phenology

3.5: Experimentation Design – A Summary

This section provides an overview of the comprehensive methodology used to classify corn growth stages, offering a consolidated framework (as depicted in Figure 32) that brings together data collection, preprocessing, model development, and evaluation as discussed in earlier sections of this chapter.

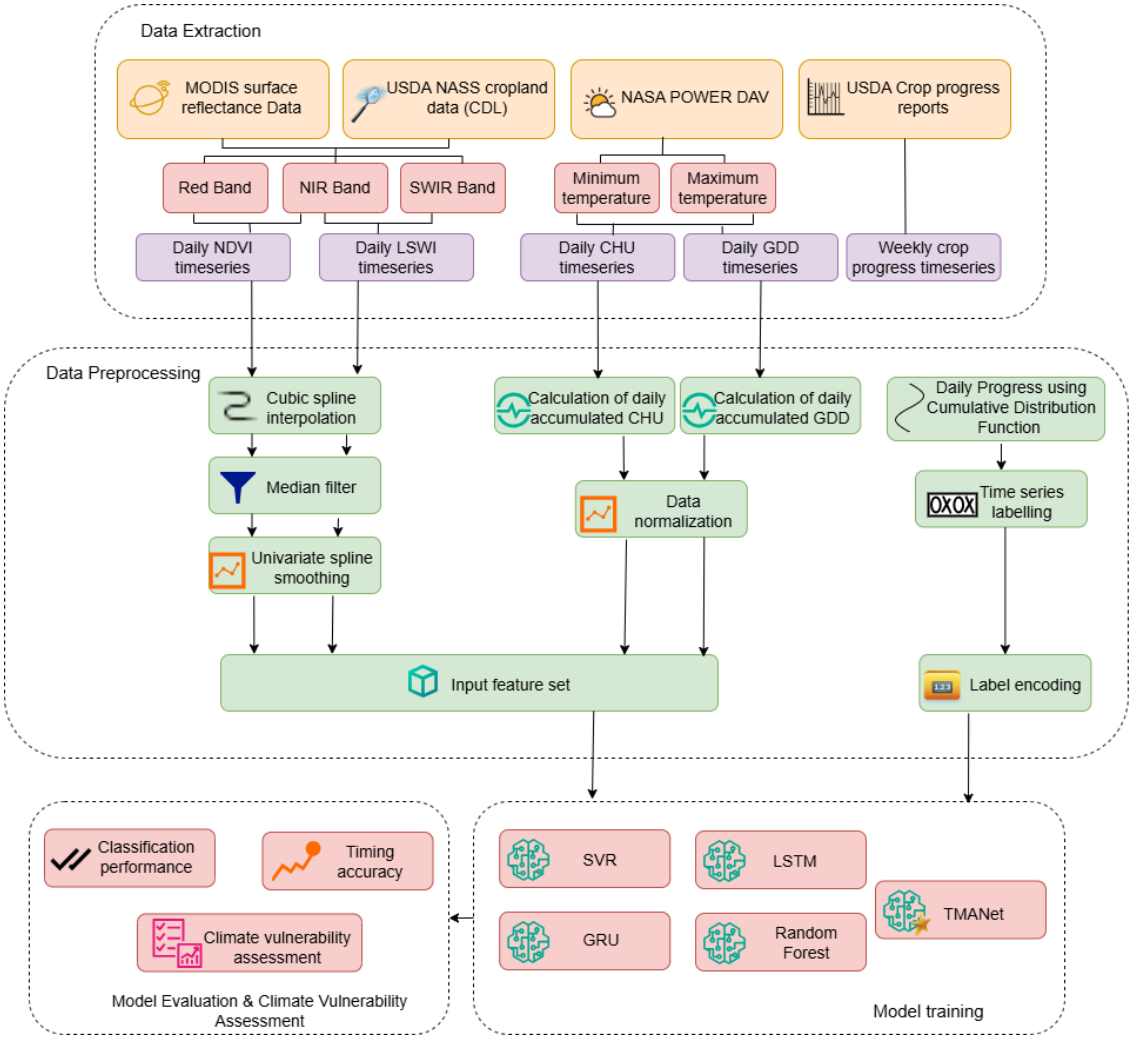

Figure 32. The overall proposed framework of methodology to characterize corn growth stages

Section 3.1 established the foundation by defining the study area and carefully selecting research variables critical to understanding corn phenology. This section detailed the rationale behind choosing specific RSD and meteorological variables, along with outlining the data extraction process to ensure high-quality inputs. It provided a comprehensive guideline detailing the process of collecting, extracting, and transforming RSD to NDVI and LSWI values tailored specifically for phenology detection in agricultural contexts. Section 3.2 followed by focusing on dataset construction and preprocessing, incorporating techniques to address data inconsistencies and enhance reliability for robust model training and validation. In Section 3.3, the focus shifted to the design and training of the proposed **Temporal Multivariate Attention Network (TMANet)**, a novel model developed to leverage temporal and multivariate attention mechanisms for detecting corn phenology. Here, TMANet's architecture was thoroughly explained against baseline models

to demonstrate its improvements in classification accuracy and phenological timing. Finally, Section 3.4 outlined the evaluation metrics to validate TMANet's effectiveness in identifying corn growth stages more accurately in comparison to baseline models and well as climate vulnerability assessment supporting sustainable agricultural practices.

Chapter 4: Results

In this section, we present the results of our study, for three regions of Iowa: Northwest, Northeast, and Central districts. The analysis is based on data collected from 2000 to 2022, described in Section 3.1, which includes vegetation indices (NDVI, LSWI), and cumulative GDD (cumGDD) and CHU (cumCHU), alongside ground-based phenology data (CPR).

The models outlined in Section 3.3 were trained independently on district-specific data to account for regional variations in climate, crop growth, and phenological patterns. This approach allowed us to capture the distinct growth stage dynamics driven by localized environmental and agro-climatic factors. The results of model performance (Section 4.1) are evaluated in terms of classification accuracy and the precision of phenological transition timing across the districts, providing a comparative analysis of the proposed TMANet model against existing state-of-the-art models. Additionally, Section 4.1.3 presents feature contribution analysis from TMANet, offering model's interpretability of feature contribution into monitoring crop's growth stages derived from its multivariate feature attention and temporal attention layers. Finally, section 4.2 documents the results of climate vulnerability assessment identifying climatic trends impacting corn phenology and highlighting vulnerabilities to climate variability.

4.1: Evaluation of Simulated Crop Phenology

4.1.1: Classification Performance of Models

Models were trained and developed based on the hyperparameters described in section 3.3 to classify each DOY (day of year) to one of the phenological stage (mentioned in Table 7). The classification results of crop growth stages for the test data using the TMANet model are illustrated in Figure 33. Black-bordered scatter dots denote instances of incorrect stage classification for specific DOY values, with adjacent numbers indicating the count of these misclassifications. Additionally, the red text represents the sample size for each growth stage, offering a detailed perspective on the model's classification accuracy across the evaluated regions.

Table 9 presents the overall classification performance of TMANet model as well as that of the existing state-of-the-art models.

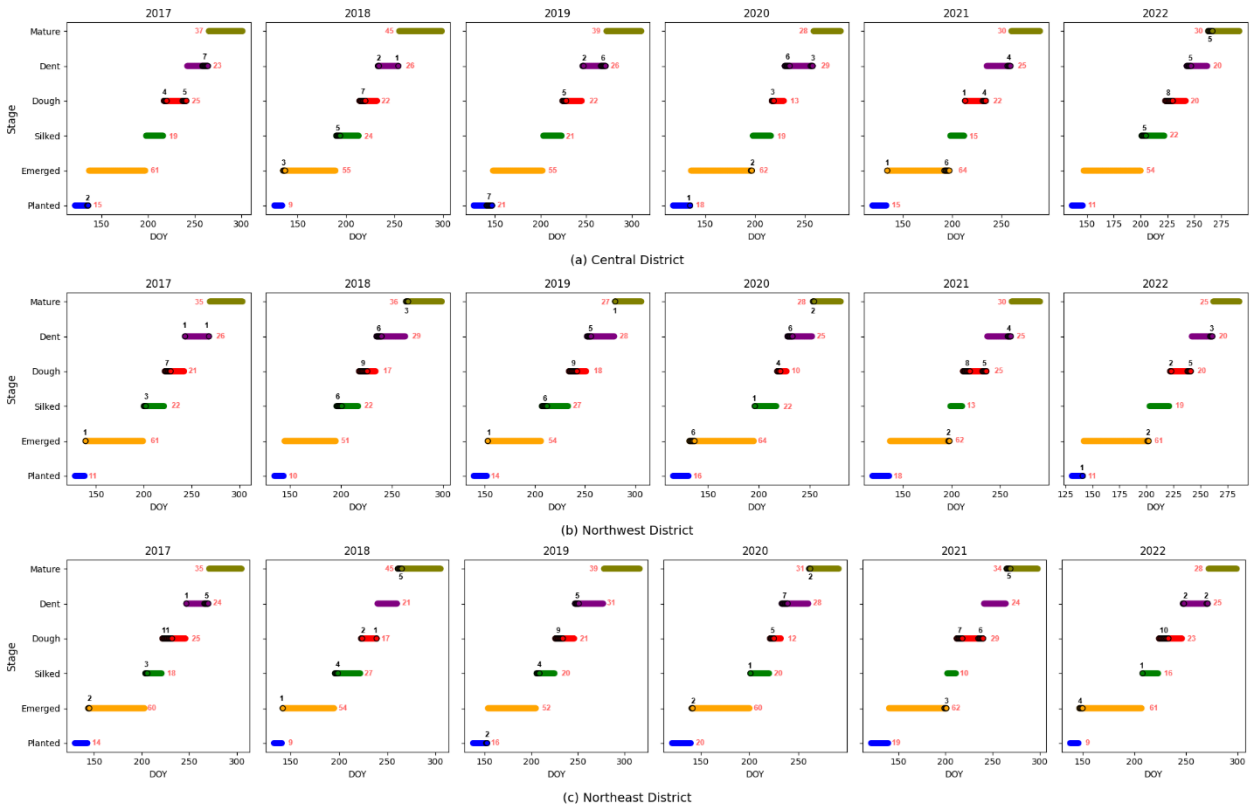

Figure 33. Accuracy assessment of corn growth stage classification (DOY vs. stage) using the TMANet model for (a) Central, (b) Northwest, and (c) Northeast districts of Iowa

Of the five approaches, our proposed TMANet model was found to achieve highest overall accuracy, precision, recall and F1-score in classification of corn growth stages across the three studied districts of Iowa. Notably, the TMANet model demonstrated superior generalization capabilities, as evidenced by its high recall values, indicating its ability to correctly detect the growth stages in comparison to other four models. Figurs 34-36 present the stage-wise precision, recall and F1-score, respectively, across the Iowa districts. Overall, TMANet consistently showed superior performance across regions and growth stages, especially in the later stages of corn growth, such as Dent and Mature, where it outperforms LSTM, GRU, RF, and SVR in almost all metrics. While models like RF and GRU occasionally outperform TMANet in certain early stages like Planted, TMANet's robustness is particularly evident in later stages and across diverse regions. Additionally, TMANet's recall values, which are higher on average, emphasize its ability to correctly classify the instances of each growth stage, making it more reliable for practical applications in phenology detection.

Table 9. Overall performance comparison of ML models for classifying corn growth stages across Iowa districts (best performances are highlighted)

	Central					*Northwest*					*Northeast*				
Metric	***TMANet***	***LSTM***	***GRU***	***RF***	***SVR***	***TMANet***	***LSTM***	***GRU***	***RF***	***SVR***	***TMANet***	***LSTM***	***GRU***	***RF***	***SVR***
Overall accuracy	**0.9**	0.87	0.88	0.87	0.88	**0.89**	0.88	0.88	0.88	0.88	**0.89**	0.88	0.87	0.88	0.87
Precision	**0.89**	0.87	0.87	0.85	0.88	**0.87**	0.86	0.86	0.86	0.86	**0.87**	0.87	0.87	0.86	0.87
Recall	**0.88**	0.84	0.83	0.84	0.86	**0.87**	0.85	0.85	0.84	0.85	**0.86**	0.82	0.82	0.85	0.83
F1 score	**0.88**	0.85	0.84	0.85	0.85	**0.87**	0.85	0.85	0.85	0.85	**0.86**	0.84	0.83	0.84	0.84

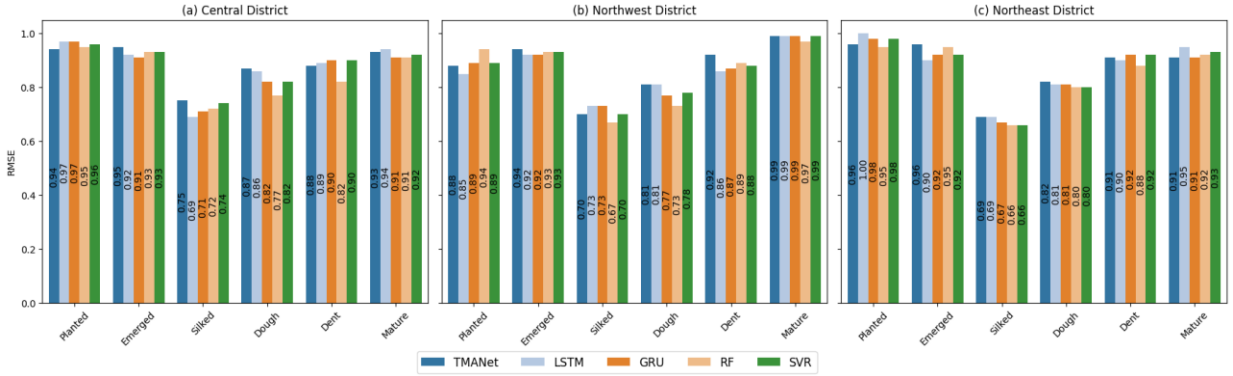

Figure 34. Comparison of precision achieved by ML models in classifying corn growth stages across Iowa districts

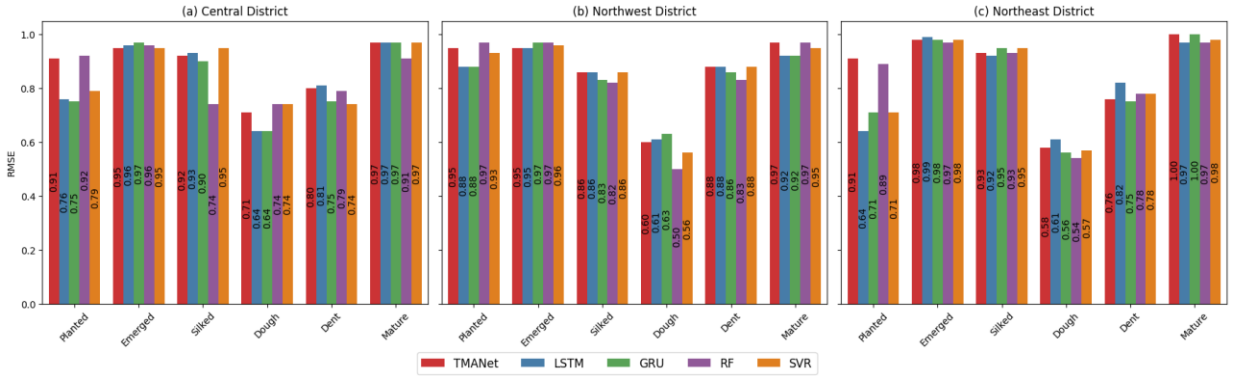

Figure 35. Recall of ML models in classifying corn growth stages across Iowa districts

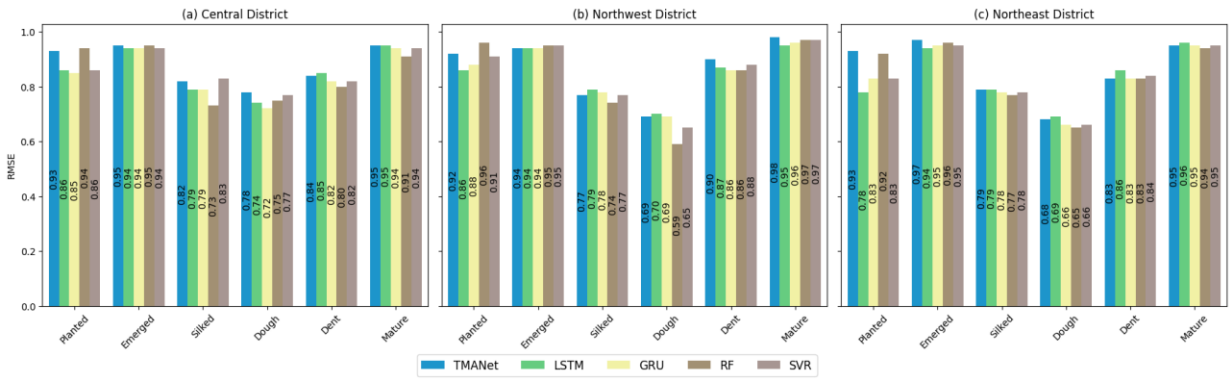

Figure 36. F1-score of ML models in classifying corn growth stages across Iowa districts

4.1.2: Timing Accuracy of Phenological Transitions

The analysis excluded the "Planted" stage, as data collection commenced from planting. Therefore, the evaluation of model performance focused on the prediction of phenological timings for subsequent growth stages. For each stage, the model's predicted DOY was compared against the ground truth DOY, assessing the model's accuracy in identifying beginning of key phenological events. Prediction accuracy was measured using RMSE and MAE, as presented in Table 10. TMANet demonstrated superior performance across all Iowa districts, achieving the lowest RMSE (under 4 days) and MAE (under 2 days), indicating its effectiveness in predicting the commencement of phenological stages with high precision (Figure 37). For stages with distinct input characteristics, such as emerged, silked, and mature, RMSE and MAE were consistently under 5 days (see Figures 38 and 39). For stages with less distinct characteristics, such as dough and dent, the RMSE remained within 8 days, further validating the robustness of the TMANet model. These results suggest that TMANet is highly effective in capturing the temporal patterns of corn growth stages and provides better predictive accuracy than traditional models such as LSTM, GRU, RF, and SVR.

4.1.3: Feature Importance from TMANet Model

This section presents the feature importance analysis derived from the Temporal Multivariate Attention Network (TMANet) in predicting corn growth stages. The analysis revealed that the importance of each feature shifted dynamically as the crop progressed through its six growth stages: Planted, Emerged, Silked, Dough, Dent, and Mature (Figure 40).

NDVI exhibited the highest importance during the early growth stages, particularly at the Planted stage, where it accounted for 62% of the model's attention weight. This indicates that

vegetation health, as captured by NDVI, is a critical indicator during the initial stages of growth when the crop is establishing its canopy. However, the influence of NDVI sharply declined in the later reproductive and maturation stages, dropping to as low as 1% by the Dent and Mature stages. This decline can be attributed to the saturation effect of NDVI in these stages, where the index plateaus and becomes less discriminative for identifying phenological transitions.

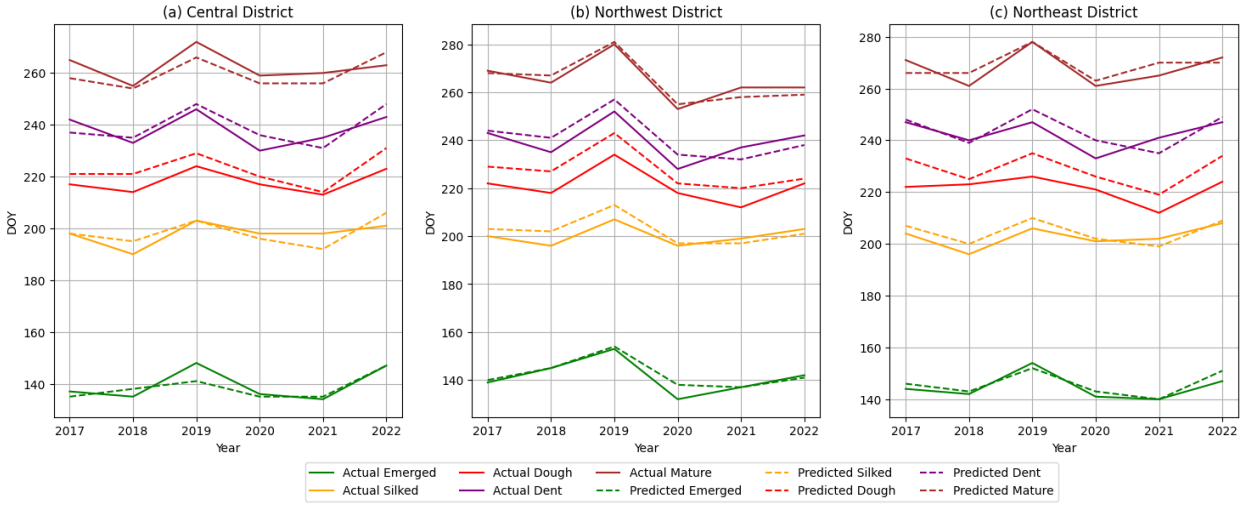

Figure 37. Accuracy assessment of estimated DOY for corn growth stages using the TMANet model compared to the actual DOY across three districts in Iowa: (a) Central, (b) Northwest, and (c) Northeast.

Table 10. Overall performance comparison of ML models in detecting phenological DOY across Iowa districts (best performances are highlighted)

	Central					Northwest					Northeast				
Metric	**TMANet**	**LSTM**	**GRU**	**RF**	**SVR**	**TMANet**	**LSTM**	**GRU**	**RF**	**SVR**	**TMANet**	**LSTM**	**GRU**	**RF**	**SVR**
RMSE	**3.79**	4.57	4.69	4.95	4.27	**4.09**	4.43	4.44	4.94	4.38	**4.28**	4.92	5.12	4.61	5.01
MAE	**1.7**	1.84	1.88	1.93	1.8	**1.71**	1.83	1.82	1.83	1.79	**1.76**	1.86	1.91	1.8	1.9

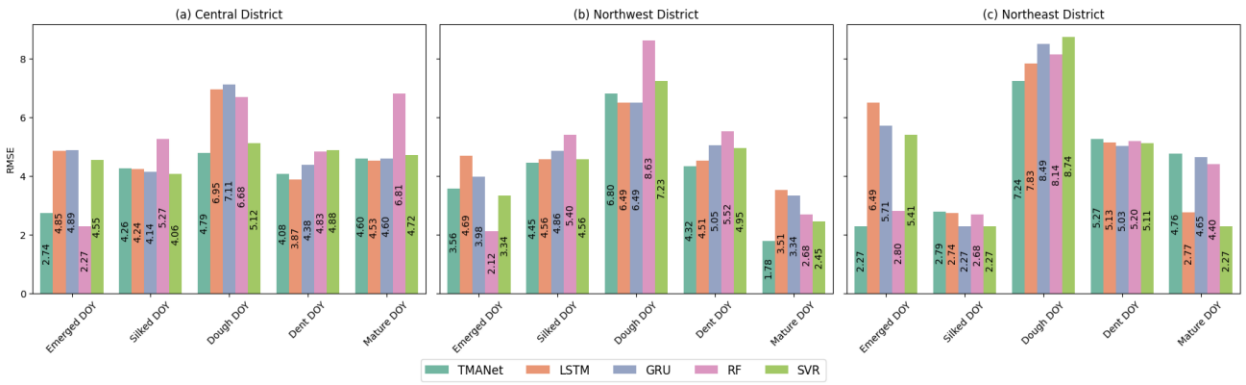

Figure 38. Comparison of RMSE of predicted DOY of corn phenological stage onset for (a) Central (b) Northwest (c) Northeast districts of Iowa

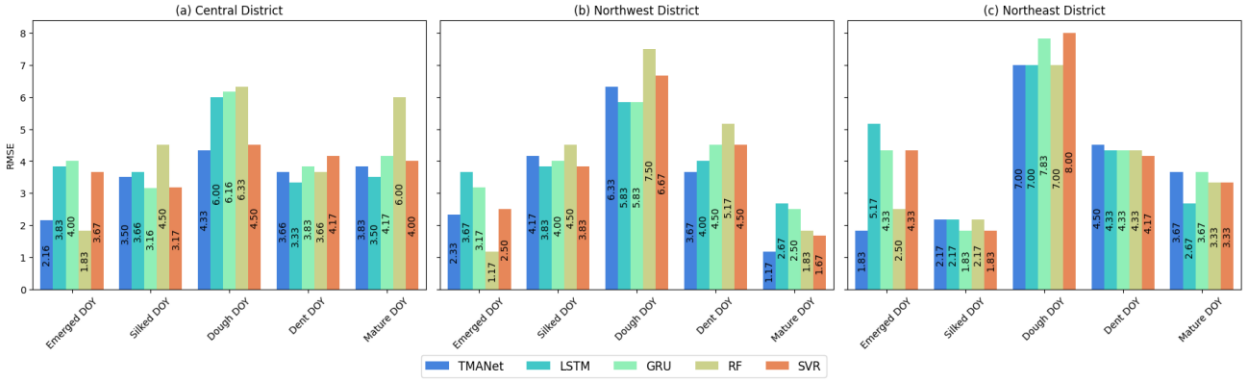

Figure 39. Comparison of MAE of predicted DOY of corn phenological stage onset for (a) Central (b) Northwest (c) Northeast districts of Iowa

In contrast, cumulative Growing Degree Days (cumGDD) emerged as the most consistent and influential predictor across all growth stages. While its importance was minimal in the Planted stage (1%), cumGDD progressively gained significance as the crop advanced through its development. It reached peak importance during the Mature stage, contributing up to 90% of the model's attention. This trend underscores the critical role of thermal accumulation in driving the crop's development from vegetative to reproductive phases. The consistent contribution of cumGDD highlights its robustness in capturing the thermal requirements necessary for phenological transitions.

Cumulative Crop Heat Units (cumCHU) followed a similar, though less profound pattern compared to cumGDD. During the Planted stage, cumCHU exhibited moderate importance (24%), maintaining relatively stable contributions throughout the remaining stages, with a slight increase during the Dent and Mature stages (23%). While both cumGDD and cumCHU are temperature-based indicators, the model's output suggests that cumGDD is a more dominant predictor, likely

due to its wider applicability in phenological modeling and its stronger correlation with thermal accumulation across stages.

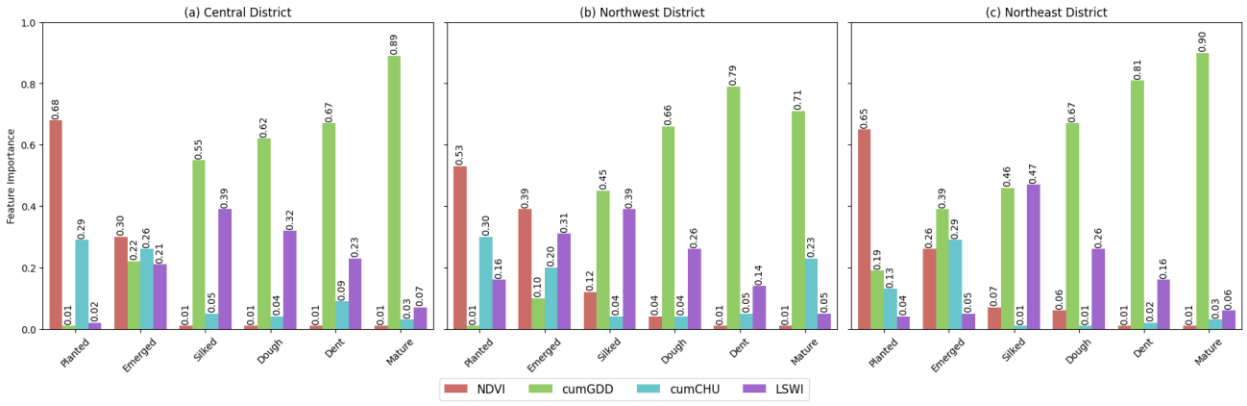

Figure 40. Overall feature importance aggregated from weights of multivariate feature and temporal attention layers in TMANet model

Lastly, Land Surface Water Index (LSWI) displayed stage-specific importance, particularly during the Emerged and Silked stages. Its contribution peaked at 39% during the Silked stage, reflecting the crucial role of moisture availability in the flowering period, which is highly sensitive to water stress. However, as the crop approached the Dent and Mature stages, the importance of LSWI diminished, corresponding to a reduced sensitivity to water content in the later growth stages when temperature-driven variables become more prominent.

Overall, the TMANet model's attention mechanism effectively identified and weighted the most relevant features at each growth stage, enabling accurate and stage-specific predictions. These results demonstrate that while NDVI plays a critical role in early-stage growth, temperature-related variables such as cumGDD and cumCHU dominate in the later stages, with LSWI contributing significantly during moisture-sensitive periods. The dynamic weighting of these features showcases the model's capacity to adapt to the evolving physiological requirements of the crop across its lifecycle.

4.2: Climate Vulnerability Assessment

4.2.1: Changing trends of Crop Phenology

The district-level analysis of inter-annual crop phenology shifts revealed notable trends in planting and harvesting dates. Planting dates across Iowa showed a delay at an average rate of 0.24 days per year (Figure 41). Similarly, corn harvesting dates were delayed by an average of 0.245

days per year in the Central and Northeast districts. In contrast, the Northwest district exhibited a slight advancement in harvesting dates, averaging 0.13 days per year. These shifts in planting and harvesting dates have influenced the length of the growing season. The Central district experienced a marginal reduction in growing season length, shortening by 0.46 days over the study period. In contrast, the Northeast district saw an increase of 1.38 days in growing season length. The most significant change was observed in the Northwest district, where the growing season length showed a profound shortening trend, averaging 0.41 days per year, resulting in a total reduction of 9.43 days over the study period.

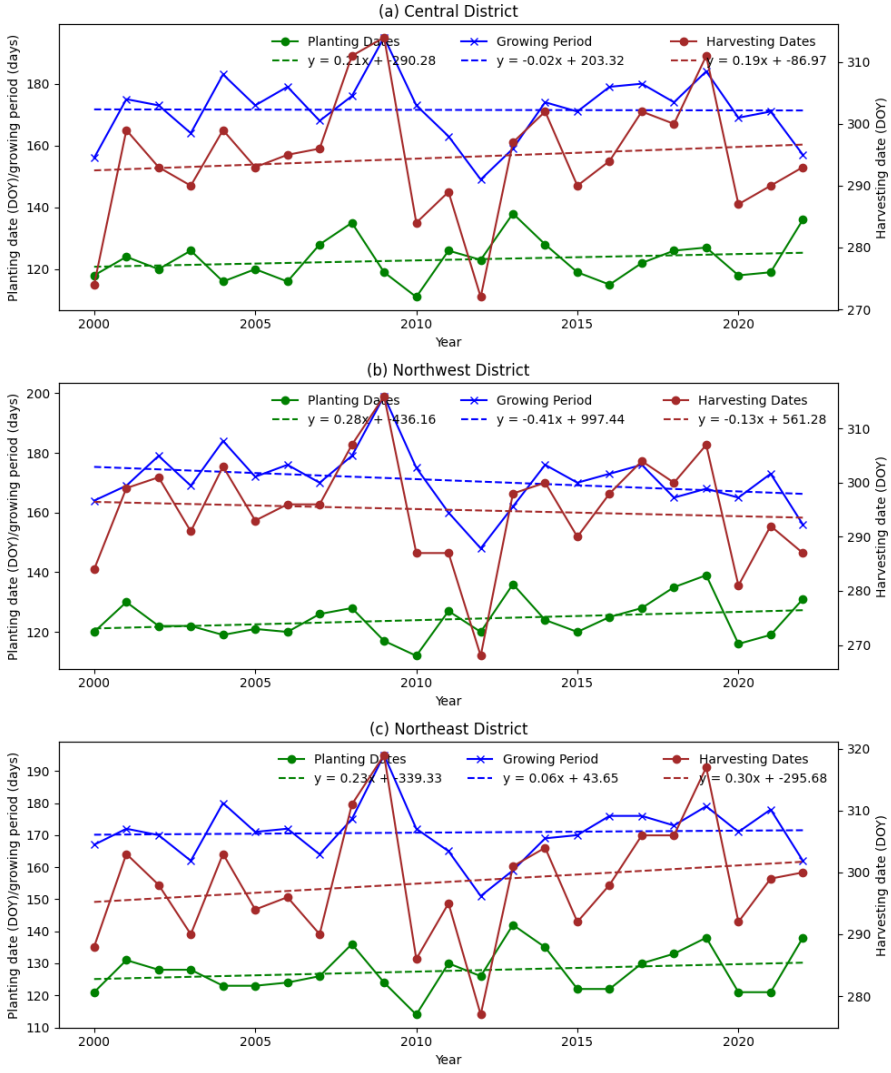

Figure 41. Linear regression analysis for trends of planting dates, harvesting dates and growing period in Iowa, 2000-2022: (a) Central district, (b) Northwest district and (c) Northeast district

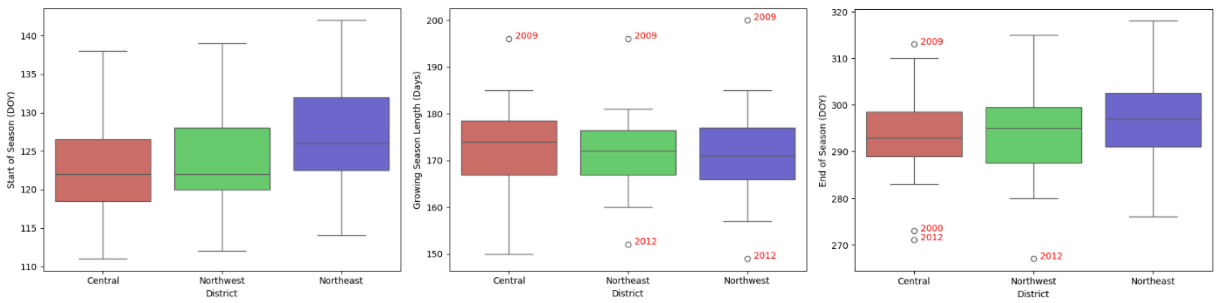

Figure 42. Boxplots representing characteristics of (a) Start of growing season, (b) Length of growing period and (c) End of growing season

The impact of climate anomalies on corn phenology is evident in the data. According to the NOAA National Centers for Environmental Information, State Climate Summaries 2022 (NOAA State Climate Summaries, 2022), the year 2012 was the hottest year on record for Iowa, with an average temperature of 52.1°F, which was 4.5°F above the long-term average (1895–2020). This trend in increasing warmth is reflected in the earlier end-of-season phenological transitions and a shortened growing season for that year (Figure 42). Contrastingly, the summer of 2009 featured cooler-than-average temperatures and heavy rainfall, particularly in August, which ranked among the coolest summers on record (NOAA National Climate Report, 2009). July temperatures were notably below normal, leading to delays in crop development. This cooler and wetter weather extended the growing season for that year, underscoring how climatic variability affects phenological events.

In addition, consistent intra-annual corn phenology trends in Iowa districts as represented in Figure 44, reveal clear and distinct variations over time across different phenological phases (emergence, silking, dough, dent, and maturity periods). The "Emergence" and "Silking" periods show significant shortening, averaging reductions of 9.4 days and 7.3 days, respectively, over the study period. In contrast, mid-season stages, such as the "Dough" and "Dent" phases, exhibit extended durations, increasing by an average of 10 days and 8.2 days, respectively, indicating shifts in crop developmental dynamics. The "Maturity" phase presents mixed trends, with a slight shortening tendency in the Central and Northwest districts, while remaining relatively stable in the Northeast. These findings underscore the complex interplay of climatic and agronomic factors influencing corn phenology across Iowa's regions.

Figure 43. Linear regression analysis for intra-annual corn growth stage periods

4.2.2 Trends in Climatic Variables

Between 2000 and 2022, Iowa's growing season exhibited significant climatic shifts, marked by consistent warming trends in minimum and average temperatures. Minimum temperatures increased at rates between 0.0323 and 0.0727 °C/year, while average temperatures rose by 0.001 to 0.0473 °C/year, as illustrated in Table 10. The Northwest district experienced the most profound warming in minimum temperatures, with a cumulative rise of 1.67 °C during the

study period. Similarly, maximum temperatures in the Northwest showed a gradual increase of 0.0219 °C/year, contrasting with cooling trends in the Central and Northeast districts, highlighting notable regional variations in temperature dynamics. Precipitation trends also displayed spatial heterogeneity, with accumulated precipitation increasing in the Northwest and Northeast regions while declining in the Central district.

Table 11. Trends of regional climatic factors in Iowa, 2000-2022.

District	Average Temperature (°C/year)	Minimum Temperature (°C/year)	Maximum Temperature (°C/year)	Precipitation (mm/year)
Central	0.0224	0.0474	-0.0027	-0.0050
Northwest	0.0473	0.0727*	0.0219	0.0004
Northeast	0.0010	0.0323	-0.0304	0.0241

Note: Trends are significant with * $p<0.05$.

These climatic changes had measurable impacts on the crop growing season, which demonstrated negative correlations with temperature variables but positive sensitivity to precipitation (Table 12). Minimum and average temperatures were significantly associated with reductions in the growing period across all regions, with a 1 °C increase in minimum and average temperatures shortening the growing period by an average of over 6 and 5 days, respectively. The evaluated minimum temperature responses likely accelerated phenological phases in Northwest district causing a pronounced reduction in growing period by over 9 days based on trend analysis in phenology mentioned in section 4.2.1 and steep rise in minimum temperatures in this region discussed in Section 4.1.2. In contrast to temperature, precipitation had a mitigating effect, particularly in the Central district, where a 1 mm increase in accumulated precipitation extended the growing period by over 4 days, reflecting the region's higher sensitivity to moisture availability compared to other parts of Iowa. These findings underscore the critical interplay between climatic factors and agricultural productivity in the state.

Table 12. Correlation between growing season period and climatic variables in Iowa, 2000-2022.

	Average Temperature		Minimum Temperature		Maximum Temperature		Precipitation	
District	**r**	**Response (days/°C)**	**r**	**Response (days/°C)**	**r**	**Response (days/°C)**	**r**	**Response (days/mm)**
Central	-0.81*	-5.92	-0.76*	-7.41	-0.79	-4.39	0.42*	4.79
Northwest	-0.80*	-5.99	-0.85*	-7.53	-0.72	-4.37	0.24	3.28
Northeast	-0.71*	-4.88	-0.65*	-5.68	-0.70	-3.70	0.33	3.42

Note: Trends are significant with * $p<0.05$.

4.2.3: Trends in Agronomic Indicators

Table 13 presents the trends in NDVI, LSWI, cumulative GDD, and cumulative CHU thresholds across various corn growth stages in Iowa. A significant increasing trend in both NDVI and LSWI thresholds was observed in the Northeastern region at the onset of the vegetation stage. This indicates that the region is influenced by wetter conditions and slightly moderated warming trends during the planted to emergence period, requiring crops to develop higher vegetation density and rely more on soil moisture to reach emergence. Statistically significant decreasing trends in the accumulated CHU required to mark the beginning of the silking stage were observed for the Central and Northwestern regions. The accumulated GDD also showed a decreasing trend; however, increasing trends in NDVI and LSWI in the Central district suggested that warming temperatures, combined with adequate moisture availability, were favorable for the reproductive stage to begin. In contrast, the Northwestern region experienced decreasing trends in both NDVI and LSWI, indicating warmer and drier conditions that delayed or altered the reproductive stage's onset.

For the dough stage, all three districts—Central (CT), Northwestern (NW), and Northeastern (NE)—show clear indications of the impacts of climate warming on phenological transitions. In the Central district, slight increases in NDVI and LSWI reflect stable or slightly improving vegetation and moisture conditions, which help mitigate some of the warming effects. However, decreases in cumulative GDD and CHU thresholds suggest accelerated crop development, driven by warming temperatures. The Northwestern district, while showing stable vegetation and moisture conditions (as evidenced by small increases in NDVI and LSWI), presents significant declines in GDD and CHU thresholds. This reflects a stronger warming effect, leading

to faster transitions to the dough stage. Similarly, the Northeastern district shows a comparable pattern, with minor declines in NDVI, stable LSWI, and sharper reductions in GDD and CHU thresholds, highlighting the intense impact of warming. Overall, while moisture conditions remain relatively stable across the regions, the pronounced reductions in heat thresholds (GDD and CHU) indicate that warming temperatures are the primary factor accelerating the phenological development of corn. This warming effect is more pronounced in the Northwestern and Northeastern districts.

In the Central district, the slight increases in NDVI and LSWI during the dent stage suggest stable or improving conditions, with moderate increases in both GDD and CHU, reflecting favorable growth patterns. However, the substantial rise in GDD and CHU during the mature stage suggests that warming temperatures are accelerating the maturation process. In contrast, the Northwestern district shows slower heat accumulation, with significant decreases in both GDD and CHU during the dent stage, indicating cooler conditions or slower crop development. During maturation, while NDVI and LSWI remain stable, moderate increases in GDD and CHU suggest that the crop is still maturing under relatively favorable conditions. The Northeastern district exhibits a more complex pattern, with a slight increase in moisture availability during both the dent and mature stages, accompanied by minimal changes in NDVI. While GDD remains stable during the dent stage, a significant increase in CHU during maturation accelerates the maturation process. Overall, these trends highlight the varying impacts of climatic conditions on crop growth. Heat accumulation, in particular, plays a dominant role in accelerating corn maturation, especially in the Central and Northeastern regions, where warming temperatures are most profound.

Table 13. Trends in thresholds of agronomic indicators to mark the beginning of corn phenological stages in Iowa, 2000-2022.

Features	Emerged			Silked			Dough			Dent			Mature		
	CT	NW	NE	CT	NW	NE	CT	NW	NE	CT	NW	NE	CT	NW	NE
NDVI	0.0016	-0.0005	0.0033*	0.0002	-0.0005	-0.0008	0.0003	0.0001	-0.0003	0.0018	0.0009	0.0017	-0.0015	0.0009	0.0007
LSWI	0.0022	0.0004	0.0034*	0.0009	-0.0011	0.0002	0.0014	0.0013	0.0014	0.0028	0.0017*	0.0038*	-0.0009	0.0039	0.0010
Cumulative GDD	-0.5314	-0.2750	-0.2485	-2.027	-2.6120	-0.5810	-2.9044	-6.1880*	-5.5466*	1.1156	-3.5662	0.0130	5.4236	3.1820	3.2085
Cumulative CHU	-0.9295	-0.2424	0.2125	-4.8053*	-6.0994*	-3.3822	-8.7134*	-13.4331*	-13.7061*	1.7028	-2.8479	0.6556	9.1436*	4.7491	8.3360*

Note: Trends are significant with * $p<0.05$. CT: Central district, NW: Northwest district and NE: Northeast district of Iowa.

Chapter 5: Conclusion and Future Work

This thesis proposed a comprehensive methodology and novel deep learning model—Temporal Multivariate Attention Network (TMANet)—to monitor crop phenology using Remote Sensing Data (RSD), with a focus on sustainable agricultural management. The research targeted corn-growing ecosystems, leveraging agro-climatic variables to address key Research Questions related to crop phenology detection and its sensitivity to climatic variability. Below, we summarize the major contributions and insights derived from the study:

- **Remote Sensing and Agro-Climatic Data Integration**: The study utilized Normalized Differentiated Vegetation Index (NDVI) and Land Surface Water Index (LSWI) as proxies for vegetation health and water content, respectively. These indices were complemented by Growing Degree Days (GDD) and Crop Heat Units (CHU) to quantify heat accumulation during the growing season. Rigorous preprocessing steps, including quality control, cubic spline interpolation, and smoothing techniques, ensured high-quality input data for phenology detection.

- **Framework for Phenology Detection**: The research proposed a consolidated dataset integrating RSD, climate data, and ground truth phenology labels. This framework addressed the limitations of traditional phenology models by combining climatic indices and vegetation indices to capture the interdependencies between growth stages and environmental factors.

- **Novel Deep Learning Model—TMANet**: TMANet introduced temporal and multivariate attention mechanisms, enabling dynamic prioritization of features and time steps most relevant to detecting phenological stages. This approach enhanced accuracy compared to state-of-the-art models while maintaining interpretability, a key advantage for practical applications in precision agriculture.

- **Climate Vulnerability Analysis**: Employing techniques such as linear regression, Pearson correlation, Mann-Kendall trend test, and Sen's slope estimator, the study identified evolving climatic patterns in Iowa. These findings underscored the sensitivity of corn phenology to climate variability, providing actionable insights for developing climate-resilient agricultural practices.

Future research can focus on enhancing the scalability and generalization of TMANet to accommodate larger datasets and higher-resolution remote sensing data. As the availability of satellite and environmental data increases, improving the model's ability to handle these inputs efficiently will be crucial for its broader application across diverse agricultural landscapes. Additionally, extending TMANet to other crops such as wheat, soybeans, and rice, as well as applying it to various geographic regions, can rigorously test its robustness and generalizability under different environmental and agricultural conditions.

Advancing the TMANet architecture also offers a promising avenue for improvement. Experimenting with more sophisticated attention mechanisms, such as multi-head or hierarchical attention layers, could better capture temporal and spatial dependencies within the data. Similarly, integrating convolutional neural networks (CNNs) with attention-based temporal layers may enhance feature extraction, particularly when dealing with spatially distributed inputs like satellite imagery. Transfer learning could further optimize the model by enabling pre-trained networks to be fine-tuned for specific crops or regions, improving both efficiency and adaptability. Moreover, incorporating multi-task learning to simultaneously predict multiple phenological stages can help capture the relationships between interdependent growth phases, providing a more comprehensive understanding of crop development.

Another significant direction is TMANet's integration into precision agriculture. By coupling the model with real-time technologies such as drones, IoT sensors, and weather monitoring systems, continuous phenology monitoring and timely interventions at critical growth stages can be facilitated. Developing user-friendly platforms to present TMANet's predictions in actionable formats would make the technology more accessible to farmers and agronomists, enabling data-driven decision-making and fostering sustainability in crop management.

Finally, TMANet's potential for addressing climate challenges opens avenues for exploring its application in dynamic phenology models that incorporate real-time data streams, enabling adaptive responses to evolving environmental conditions. Insights generated by the model can also inform agricultural policies aimed at mitigating the impacts of climate change, such as optimizing resource allocation and improving resilience in crop management practices. These advancements will enhance the model's practical relevance, contributing to sustainable and climate-resilient agricultural systems.

References

1. Adole, T., Dash, J., & Atkinson, P. M. (2016). A systematic review of vegetation phenology in Africa. *Ecological Informatics*, *34*, 117-128.
2. Adole, T., Dash, J., Rodriguez-Galiano, V., & Atkinson, P. M. (2019). Photoperiod controls vegetation phenology across Africa. *Communications biology*, *2*(1), 391.
3. Afzal, M. N., Tariq, M., Ahmed, M., Abbas, G., & Mehmood, Z. (2020). Managing planting time for cotton production. *Cotton Production and Uses: Agronomy, Crop Protection, and Postharvest Technologies*, 31-44.
4. Al-Kaisi, M. M., & Kwaw-Mensah, D. (2020). Quantifying soil carbon change in a long-term tillage and crop rotation study across Iowa landscapes. *Soil Science Society of America Journal*, *84*(1), 182-202.
5. Anandhi, A. (2016). Growing degree days–Ecosystem indicator for changing diurnal temperatures and their impact on corn growth stages in Kansas. *Ecological Indicators*, *61*, 149-158.
6. Angel, J. R., Widhalm, M., Todey, D., Massey, R., & Biehl, L. (2017). The U2U corn growing degree day tool: Tracking corn growth across the US Corn Belt. *Climate Risk Management*, *15*, 73-81.
7. Asokan, A., & Anitha, J. (2019). Change detection techniques for remote sensing applications: A survey. *Earth Science Informatics*, *12*, 143-160.
8. Ban, H. Y., Kim, K. S., Park, N. W., & Lee, B. W. (2016). Using MODIS data to predict regional corn yields. *Remote Sensing*, *9*(1), 16.
9. Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
10. Beck, P. S., Atzberger, C., Høgda, K. A., Johansen, B., & Skidmore, A. K. (2006). Improved monitoring of vegetation dynamics at very high latitudes: A new method using MODIS NDVI. *Remote sensing of Environment*, *100*(3), 321-334.
11.
12. Butler, E. E., & Huybers, P. (2015). Variations in the sensitivity of US maize yield to extreme temperatures by region and growth phase. *Environmental Research Letters*, *10*(3), 034009.
13. Butler, E. E., Mueller, N. D., & Huybers, P. (2018). Peculiarly pleasant weather for US maize. *Proceedings of the National Academy of Sciences*, *115*(47), 11935-11940.
14. Cao, G. (2022). Application of Remote Sensing Technology in Forest Resources Investigation. *Remote Sensing*, *11*(2), 46-49.
15. Čehulić, I., Sever, K., Katičić Bogdan, I., Jazbec, A., Škvorc, Ž., & Bogdan, S. (2019). Drought impact on leaf phenology and spring frost susceptibility in a Quercus robur L. provenance trial. *Forests*, *10*(1), 50.
16. Cui, L., Shi, J., Ma, Y., & Du, H. (2017). Distribution and trend in the thermal growing season in China during 1961–2015. *Physical Geography*, *38*(6), 506-523.
17. Chen, W., & Shi, K. (2021). Multi-scale attention convolutional neural network for time series classification. *Neural Networks*, *136*, 126-140.
18. Cui, L., & Shi, J. (2021). Evaluation and comparison of growing season metrics in arid and semi-arid areas of northern China under climate change. *Ecological Indicators*, *121*, 107055.
19. Czernecki, B., Nowosad, J., & Jabłońska, K. (2018). Machine learning modeling of plant phenology based on coupling satellite and gridded meteorological dataset. *International journal of biometeorology*, *62*, 1297-1309.
20. de Moura, Y. M., Galvão, L. S., Hilker, T., Wu, J., Saleska, S., do Amaral, C. H., ... & Aragão, L. E. (2017). Spectral analysis of amazon canopy phenology during the dry season using a tower hyperspectral camera and modis observations. *ISPRS Journal of Photogrammetry and Remote Sensing*, *131*, 52-64.
21. Diao, C. (2020). Remote sensing phenological monitoring framework to characterize corn and soybean physiological growing stages. *Remote Sensing of Environment*, *248*, 111960.
22. Donthu, N., Kumar, S., Mukherjee, D., Pandey, N., & Lim, W. M. (2021). How to conduct a bibliometric analysis: An overview and guidelines. *Journal of business research*, *133*, 285-296.
23. Ellegaard, O., & Wallin, J. A. (2015). The bibliometric analysis of scholarly production: How great is the impact?. *Scientometrics*, *105*, 1809-1831.

24. Gallinat, A. S., Ellwood, E. R., Heberling, J. M., Miller-Rushing, A. J., Pearse, W. D., & Primack, R. B. (2021). Macrophenology: insights into the broad-scale patterns, drivers, and consequences of phenology. *American journal of botany*, *108*(11), 2112-2126.
25. Gao, F., Anderson, M. C., & Hively, W. D. (2020). Detecting cover crop end-of-season using VENμS and sentinel-2 Satellite imagery. *Remote Sensing*, *12*(21), 3524.
26. Gao, F., & Zhang, X. (2021). Mapping crop phenology in near real-time using satellite remote sensing: Challenges and opportunities. Journal of Remote Sensing.
27. Gilmore, E.C., and J.S. Rogers. 1958. Heat units as a method of measuring maturity in corn. Agron. J. 50:611–615. doi:10.2134/agronj1958.00 021962005000100014x
28. GEE, 2024. https://earthengine.google.com/
29. GEE Data Catalog, 2024. https://developers.google.com/earth-engine/datasets/
30. Gong, Z., Kawamura, K., Ishikawa, N., Goto, M., Wulan, T., Alateng, D., ... & Ito, Y. (2015). MODIS normalized difference vegetation index (NDVI) and vegetation phenology dynamics in the Inner Mongolia grassland. *Solid Earth*, *6*(4), 1185-1194.
31. Gong, Z., Ge, W., Guo, J., & Liu, J. (2024). Satellite remote sensing of vegetation phenology: Progress, challenges, and opportunities. *ISPRS Journal of Photogrammetry and Remote Sensing*, *217*, 149-164.
32. Gupta, H., Kaur, L., Asra, M., Avtar, R., & Reddy, C. S. (2021). MODIS NDVI multi-temporal analysis confirms farmer perceptions on seasonality variations affecting apple orchards in Kinnaur, Himachal Pradesh. Agriculture, 11(8), 724.
33. Guan, K., Wood, E. F., Medvigy, D., Kimball, J., Pan, M., Caylor, K. K., ... & Jones, M. O. (2014). Terrestrial hydrological controls on land surface phenology of African savannas and woodlands. *Journal of Geophysical Research: Biogeosciences*, *119*(8), 1652-1669.
34. Guo, Y., Chen, S., Fu, Y. H., Xiao, Y., Wu, W., Wang, H., & Beurs, K. D. (2022). Comparison of multi-methods for identifying maize phenology using phenocams. *Remote Sensing*, *14*(2), 244.
35. Huang, J. (2016). Different sowing dates affected cotton yield and yield components. *International Journal of Plant Production*, *10*(1), 63-83.
36. Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM fully convolutional networks for time series classification. *IEEE access*, *6*, 1662-1669.
37. Kerner, H. R., Sahajpal, R., Pai, D. B., Skakun, S., Puricelli, E., Hosseini, M., ... & Becker-Reshef, I. (2022). Phenological normalization can improve in-season classification of maize and soybean: A case study in the central US corn belt. *Science of Remote Sensing*, *6*, 100059.
38. Kobayashi, S., & Ide, H. (2022). Rice crop monitoring using Sentinel-1 SAR data: A case study in Saku, Japan. *Remote Sensing*, *14*(14), 3254.
39. Kumudini, S., Andrade, F. H., Boote, K. J., Brown, G. A., Dzotsi, K. A., Edmeades, G. O., ... & Tollenaar, M. (2014). Predicting maize phenology: intercomparison of functions for developmental response to temperature. *Agronomy Journal*, *106*(6), 2087-2097.
40. Khaiter, P. A., & Erechtchoukova, M. G. (2022). Advanced Scientific Methods and Tools in Sustainable Forest Management: A Synergetic Perspective. In: Forest Dynamics and Conservation (Kumar M., Dhyani S., Kalra N. – Eds). Springer Singapore.: 279–312.
41. Leale, G., Cocitto, B., Cardoso, A. L., Lafluf, P., Tantucci, L., & Mendez, F. (2023). Predicting crop phenology: a simple logistic regression model approach. In *Congreso Argentino de AgroInformática (CAI 2023)-JAIIO 52 (Universidad Nacional de Tres de Febrero, 4 al 8 de septiembre de 2023)*.
42. Liao, L., Li, H., Shang, W., & Ma, L. (2022). An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *31*(3), 1-40.
43. Liu, G., Liu, H., & Yin, Y. (2013). Global patterns of NDVI-indicated vegetation extremes and their sensitivity to climate extremes. Environmental Research Letters, 8(2), 025009.
44. Liu, F., Wang, X., & Wang, C. (2019). Measuring vegetation phenology with near-surface remote sensing in a temperate deciduous forest: effects of sensor type and deployment. Remote Sensing, 11(9), 1063.
45. Liu, L., Zhang, X., Yu, Y., Gao, F., & Yang, Z. (2018). Real-time monitoring of crop phenology in the Midwestern United States using VIIRS observations. *Remote Sensing*, *10*(10), 1540.

46. Li, J., Burnham, J. F., Lemley, T., & Britton, R. M. (2010). Citation analysis: Comparison of web of science®, scopus™, SciFinder®, and google scholar. *Journal of electronic resources in medical libraries*, *7*(3), 196-217.

47. Li, P., Peng, C., Wang, M., Luo, Y., Li, M., Zhang, K., ... & Zhu, Q. (2018). Dynamics of vegetation autumn phenology and its response to multiple environmental factors from 1982 to 2012 on Qinghai-Tibetan Plateau in China. *Science of the Total Environment*, *637*, 855-864.
48. Li, X., Zhu, W., Xie, Z., Zhan, P., Huang, X., Sun, L., & Duan, Z. (2021). Assessing the effects of time interpolation of NDVI composites on phenology trend estimation. *Remote Sensing*, *13*(24), 5018.

49. Lechner, A. M., Foody, G. M., & Boyd, D. S. (2020). Applications in remote sensing to forest ecology and management. *One Earth*, *2*(5), 405-412.
50. Luedeling, E., Zhang, M., & Girvetz, E. H. (2009). Climatic changes lead to declining winter chill for fruit and nut trees in California during 1950–2099. *PloS one*, *4*(7), e6166.

51. Lobell, D. B., Hammer, G. L., McLean, G., Messina, C., Roberts, M. J., & Schlenker, W. (2013). The critical role of extreme heat for maize production in the United States. *Nature climate change*, *3*(5), 497-501.
52. Mahjoub, S., Chrifi-Alaoui, L., Marhic, B., & Delahoche, L. (2022). Predicting energy consumption using LSTM, multi-layer GRU and drop-GRU neural networks. *Sensors*, *22*(11), 4062.
53. Matongera, T. N., Mutanga, O., Sibanda, M., & Odindi, J. (2021). Estimating and monitoring land surface phenology in rangelands: A review of progress and challenges. *Remote Sensing*, *13*(11), 2060.
54. McMaster, G. S., & Wilhelm, W. W. (1997). Growing degree-days: one equation, two interpretations. *Agricultural and forest meteorology*, *87*(4), 291-300.
55. Morellato, L. P. C., Alberton, B., Alvarado, S. T., Borges, B., Buisson, E., Camargo, M. G. G., ... & Peres, C. A. (2016). Linking plant phenology to conservation biology. *Biological conservation*, *195*, 60-72.
56. Murguia-Cozar, A., Macedo-Cruz, A., Fernandez-Reynoso, D. S., & Salgado Transito, J. A. (2021). Recognition of Maize Phenology in Sentinel Images with Machine Learning. *Sensors*, *22*(1), 94.
57. Myers, E., Kerekes, J., Daughtry, C., & Russ, A. (2019). Assessing the impact of satellite revisit rate on estimation of corn phenological transition timing through shape model fitting. *Remote Sensing*, *11*(21), 2558.
58. NASA POWER, 2024. https://power.larc.nasa.gov/data-access-viewer/
59. NASA MODIS, 2024. https://developers.google.com/earth-engine/datasets/catalog/MODIS_061_MCD43A4
60. NASS CDL, 2024. https://www.nass.usda.gov/Research_and_Science/Cropland/metadata/meta.php
61. NASS CPR, 2024. https://www.nass.usda.gov/Publications/National_Crop_Progress/
62. Nielsen, R. L. (2002). Corn growth and development: What goes on from planting to harvest. *Purdue University, University Extension, West Lafayette, IN*.
63. Nieto, L., Schwalbert, R., Prasad, P. V., Olson, B. J., & Ciampitti, I. A. (2021). An integrated approach of field, weather, and satellite data for monitoring maize phenology. *Scientific Reports*, *11*(1), 15711.
64. NOAA State Climate Summaries. (2022). https://statesummaries.ncics.org/
65. NOAA National Climate Report (2009) . https://www.ncei.noaa.gov/access/monitoring/monthly-report/national/200908
66. Olesen, J. E., Børgesen, C. D., Elsgaard, L., Palosuo, T., Rötter, R. P., Skjelvåg, A. O., ... & Van Der Fels-Klerx, H. J. (2012). Changes in time of sowing, flowering and maturity of cereals in Europe under climate change. *Food Additives & Contaminants: Part A*, *29*(10), 1527-1542.
67. Park, T., Ganguly, S., Tømmervik, H., Euskirchen, E. S., Høgda, K. A., Karlsen, S. R., ... & Myneni, R. B. (2016). Changes in growing season duration and productivity of northern vegetation inferred from long-term remote sensing data. Environmental Research Letters, 11(8), 084001.
68. Potgieter, A. B., Zhao, Y., Zarco-Tejada, P. J., Chenu, K., Zhang, Y., Porker, K., ... & Chapman, S. (2021). Evolution and application of digital technologies to predict crop type and crop phenology in agriculture. *in silico Plants*, *3*(1), diab017.

69. Pritchard, A. (1969). Statistical bibliography or bibliometrics. *Journal of documentation*, *25*, 348.
70. Ritchie, S. W., Hanway, J. J., & Benson, G. O. (1989). *How a corn plant develops*. Iowa State University of Science and Technology, Cooperative Extension Service.

71. Ramadhani, F., Pullanagari, R., Kereszturi, G., & Procter, J. (2020). Mapping of rice growth phases and bare land using Landsat-8 OLI with machine learning. International Journal of Remote Sensing, 41(21), 8428-8452.
72. Reed, B. C., Brown, J. F., VanderZee, D., Loveland, T. R., Merchant, J. W., & Ohlen, D. O. (1994). Measuring phenological variability from satellite imagery. *Journal of vegetation science*, *5*(5), 703-714.
73. Richardson, A. D., Keenan, T. F., Migliavacca, M., Ryu, Y., Sonnentag, O., & Toomey, M. (2013). Climate change, phenology, and phenological control of vegetation feedbacks to the climate system. *Agricultural and Forest Meteorology*, *169*, 156-173.
74. Rubio-Cuadrado, Á., Camarero, J. J., Rodríguez-Calcerrada, J., Perea, R., Gómez, C., Montes, F., & Gil, L. (2021). Impact of successive spring frosts on leaf phenology and radial growth in three deciduous tree species with contrasting climate requirements in central Spain. *Tree physiology*, *41*(12), 2279-2292.
75. Seo, B., Lee, J., Lee, K. D., Hong, S., & Kang, S. (2019). Improving remotely-sensed crop monitoring by NDVI-based crop phenology estimators for corn and soybeans in Iowa and Illinois, USA. Field Crops Research, 238, 113-128.
76. Schepaschenko, D., See, L., Lesiv, M., Bastin, J. F., Mollicone, D., Tsendbazar, N. E., ... & Fritz, S. (2019). Recent advances in forest observation with visual interpretation of very high-resolution imagery. *Surveys in Geophysics*, *40*, 839-862.
77. Shen, Y., Zhang, X., & Yang, Z. (2022). Mapping corn and soybean phenometrics at field scales over the United States Corn Belt by fusing time series of Landsat 8 and Sentinel-2 data with VIIRS data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *186*, 55-69.
78. Streck, N. A., Lago, I., Gabriel, L. F., & Samboranha, F. K. (2008). Simulating maize phenology as a function of air temperature with a linear and a nonlinear model. Pesquisa Agropecuária Brasileira, 43, 449-455.
79. Stewart, D. W., Dwyer, L. M., & Carrigan, L. L. (1998). Phenological temperature response of maize. *Agronomy Journal*, *90*(1), 73-79.
80. Szantoi, Z., & Strobl, P. (2019). Copernicus Sentinel-2 calibration and validation. *European Journal of Remote Sensing*, *52*(1), 253-255.
81. Szomszor, M., Adams, J., Fry, R., Gebert, C., Pendlebury, D. A., Potter, R. W., & Rogers, G. (2021). Interpreting bibliometric data. *Frontiers in Research Metrics and Analytics*, 30.
82. Tang, J., Körner, C., Muraoka, H., Piao, S., Shen, M., Thackeray, S. J., & Yang, X. (2016). Emerging opportunities and challenges in phenology: a review. *Ecosphere*, *7*(8), e01436.
83. Thomson, L. J., Macfadyen, S., & Hoffmann, A. A. (2010). Predicting the effects of climate change on natural enemies of agricultural pests. *Biological control*, *52*(3), 296-306.
84. Tripathi, A. M., & Baruah, R. D. (2020, July). Multivariate time series classification with an attention-based multivariate convolutional neural network. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.
85. USDA Historical CPR, 2024. https://usda.library.cornell.edu/concern/publications/
86. USGS. (2011). The USA National Phenology Network. Retrieved October 205, 2024 from https://pubs.usgs.gov/fs/2011/3023/fs2011-3023.pdf
87. Vastaranta, M., Saarinen, N., Yrttimaa, T., & Tokola, T. (2020). Introduction to geoinformation science.
88. Venancio, L. P., Filgueiras, R., da Cunha, F. F., Silva, F. C. D. S., dos Santos, R. A., & Mantovani, E. C. (2020). Mapping of corn phenological stages using NDVI from OLI and MODIS sensors.
89. VOSViewer, 2023. https://www.vosviewer.com/
90. Vorobiova, N., & Chernov, A. (2017). Curve fitting of MODIS NDVI time series in the task of early crops identification by satellite images. *Procedia engineering*, *201*, 184-195.
91. Wang, Y., Luo, Y., & Shafeeque, M. (2019). Interpretation of vegetation phenology changes using daytime and night-time temperatures across the Yellow River Basin, China. *Science of the Total Environment*, *693*, 133553.
92. Wang, R., Cherkauer, K., & Bowling, L. (2016). Corn response to climate stress detected with satellite-based NDVI time series. *Remote Sensing*, *8*(4), 269.

93. Weiss, M., Jacob, F., & Duveiller, G. (2020). Remote sensing for agricultural applications: A meta-review. Remote sensing of environment, 236, 111402.
94. Wu, B., Zhang, M., Zeng, H., Tian, F., Potgieter, A. B., Qin, X., ... & Loupian, E. (2023). Challenges and opportunities in remote sensing-based crop monitoring: A review. *National Science Review*, *10*(4), nwac290.
95. Wu, H., Liu, B., Zhu, B., Zhen, Z., Song, K., & Ren, J. (2024). Combining Vegetation Indices to Identify the Maize Phenological Information Based on the Shape Model. *Agriculture*, *14*(9), 1608.
96. Yang, Y., Tao, B., Liang, L., Huang, Y., Matocha, C., Lee, C. D., ... & Ren, W. (2021). Detecting recent crop phenology dynamics in corn and soybean cropping systems of Kentucky. Remote Sensing, 13(9), 1615.
97. Yang, Z., Diao, C., & Gao, F. (2023). Towards Scalable Within-Season Crop Mapping With Phenology Normalization and Deep Learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 1390-1402.
98. Yang, Y., Ren, W., Tao, B., Ji, L., Liang, L., Ruane, A. C., ... & Tian, Q. (2020). Characterizing spatiotemporal patterns of crop phenology across North America during 2000–2016 using satellite imagery and agricultural survey data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *170*, 156-173.
99. Ye, J., Bao, W., Liao, C., Chen, D., & Hu, H. (2023). Corn phenology detection using the derivative dynamic time warping method and sentinel-2 time series. *Remote Sensing*, *15*(14), 3456.
100. Zai, F. H., Hamers, H., & McSharry, P. E. (2019). Evaluating the predictive accuracy of corn phenological durations in Iowa and Illinois. *Agronomy Journal*, *111*(3), 1128-1140.
101. Zainuddin, Z., EA, P. A., & Hasan, M. H. (2021). Predicting machine failure using recurrent neural network-gated recurrent unit (RNN-GRU) through time series data. *Bulletin of Electrical Engineering and Informatics*, *10*(2), 870-878.
102. Zeng, L., Wardlow, B. D., Wang, R., Shan, J., Tadesse, T., Hayes, M. J., & Li, D. (2016). A hybrid approach for detecting corn and soybean phenology with time-series MODIS data. *Remote Sensing of Environment*, *181*, 237-250.
103. Zeng, L., Wardlow, B. D., Xiang, D., Hu, S., & Li, D. (2020). A review of vegetation phenological metrics extraction using time-series, multispectral satellite data. *Remote Sensing of Environment*, *237*, 111511.
104. Zhang, X., Friedl, M. A., Schaaf, C. B., Strahler, A. H., Hodges, J. C., Gao, F., ... & Huete, A. (2003). Monitoring vegetation phenology using MODIS. *Remote sensing of environment*, *84*(3), 471-475.
105. Zhu, Z., & Woodcock, C. E. (2012). Object-based cloud and cloud shadow detection in Landsat imagery. Remote sensing of environment, 118, 83-94.
106. Zhao, W., Qu, Y., Zhang, L., & Li, K. (2022). Spatial-aware SAR-optical time-series deep integration for crop phenology tracking. *Remote Sensing of Environment*, *276*, 113046.
107. Zhong, L., Gong, P., & Biging, G. S. (2014). Efficient corn and soybean mapping with temporal extendability: A multi-year experiment using Landsat imagery. *Remote Sensing of Environment*, *140*, 1-13.
108. Zhong, L., Hu, L., Yu, L., Gong, P., & Biging, G. S. (2016). Automated mapping of soybean and corn using phenology. *ISPRS Journal of Photogrammetry and Remote Sensing*, *119*, 151-164.

Appendices

Appendix A. Code for feature extraction

This section contains sample code for retrieving NDVI and LSWI values from google earth engine. To run this code, you need to specify the data range and area of interest (AOI) for which you want to retrieve the indices. In this example, I have set the coordinates of my AOI as per the corn region shown in figure 15 and my sample data range is from January 1, 2022 to January 1, 2023.

Code to retrieve NDVI values:

```
var year = 2022;
// Load the USDA Cropland Data Layer for the specified year
var cdl = ee.ImageCollection('USDA/NASS/CDL');
var cdlYear = cdl.filter(ee.Filter.calendarRange(year, year, 'year')).first();
var cropland_band = cdlYear.select('cropland')
//# Define the crop code for corn (e.g., 1 for corn; check CDL documentation for the correct code)
var corn_code = 1
//# Create a mask for corn fields
var corn_mask = cropland_band.eq(corn_code)
//# Apply the mask to the CDL image
var corn_fields = cropland_band.updateMask(corn_mask)
// Get the coordinates of corn fields
var cornCoords = corn_fields.reduceToVectors({
 reducer: ee.Reducer.countEvery(),
 geometry: aoi,
 scale: 30,
 maxPixels: 1e9
});
print(cornCoords)
// Define the time period
var startDate = '2022-01-01';
var endDate = '2023-01-01';
```

```
// Function to add NDVI band
var addNDVI = function(image) {
  var ndvi = image.normalizedDifference(['Nadir_Reflectance_Band2', 'Nadir_Reflectance_Band1']);
  return image.addBands(ndvi.rename('NDVI'));
};
// Function to apply a more relaxed QA mask
var filterqa = function(image) {
  // Allowing quality values less than or equal to 3
  var mask1 = image.select("BRDF_Albedo_Band_Mandatory_Quality_Band1").lte(3);
  var mask2 = image.select("BRDF_Albedo_Band_Mandatory_Quality_Band2").lte(3);
  return image.updateMask(mask1).updateMask(mask2);
};
// Load the MODIS NDVI data
var modisNDVI = ee.ImageCollection('MODIS/006/MCD43A4')
  .filterDate(startDate, endDate)
  .map(filterqa)
  .map(addNDVI);
// Function to extract daily NDVI values for each feature
var extractNDVI = function(image) {
  // Reduce the image collection by region
  var ndviValues = cornCoords.map(function(feature) {
    var geom = feature.geometry();
    var mean = image.reduceRegion({
      reducer: ee.Reducer.mean(),
      geometry: geom,
      scale: 500
    }).get('NDVI');
    return ee.Feature(null, {NDVI: mean});
  });
  var meanNDVI = ndviValues.reduceColumns(ee.Reducer.mean(), ['NDVI']).get('mean');
```

```
  return image.set('meanNDVI', meanNDVI);
};
// Map the function over the image collection
var dailyNDVI = modisNDVI.map(extractNDVI);
print(dailyNDVI)
// Convert to a feature collection for exporting
var ndviFeatureCollection = dailyNDVI.map(function(image) {
  return ee.Feature(null, {
    date: image.date().format('YYYY-MM-dd'),
    meanNDVI: image.get('meanNDVI')
  });
});
// Print the NDVI time series for verification
print(ndviFeatureCollection);
// Export the NDVI time series to Google Drive
Export.table.toDrive({
  collection: ndviFeatureCollection,
  description: 'Daily_Averaged_Corn_NDVI_TimeSeries_2022_NE',
  fileFormat: 'CSV'
});
```

Code to retrieve LSWI values:

```
var data = dataset
    .filter(ee.Filter.date('2022-01-01', '2023-12-31'))
    .first().select('cropland');
var extractedProperty = data.eq(1);
var maskedFeature = extractedProperty.selfMask();
Map.addLayer(maskedFeature, {min: 0, max: 1, palette: 'FF0000'}, 'Feature at Value 1');
var year = 2022;

// Load the USDA Cropland Data Layer for the specified year
```

```
var cdl = ee.ImageCollection('USDA/NASS/CDL');
var cdlYear = cdl.filter(ee.Filter.calendarRange(year, year, 'year')).first();
var cropland_band = cdlYear.select('cropland')
//# Define the crop code for corn (e.g., 1 for corn; check CDL documentation for the correct code)
var corn_code = 1
//# Create a mask for corn fields
var corn_mask = cropland_band.eq(corn_code)
//# Apply the mask to the CDL image
var corn_fields = cropland_band.updateMask(corn_mask)
// Get the coordinates of corn fields
var cornCoords = corn_fields.reduceToVectors({
 reducer: ee.Reducer.countEvery(),
 geometry: aoi,
 scale: 30,
 maxPixels: 1e9
});
print(cornCoords)
// Define the time period
var startDate = '2022-01-01';
var endDate = '2023-01-01';
// Function to add NDVI band
var addNDVI = function(image) {
 var ndvi = image.normalizedDifference(['Nadir_Reflectance_Band2', 'Nadir_Reflectance_Band6']);
 return image.addBands(ndvi.rename('LSWI'));
};
// Function to apply a more relaxed QA mask
var filterqa = function(image) {
 // Allowing quality values less than or equal to 3
 var mask1 = image.select("BRDF_Albedo_Band_Mandatory_Quality_Band6").lte(3);
 var mask2 = image.select("BRDF_Albedo_Band_Mandatory_Quality_Band2").lte(3);
```

```
  return image.updateMask(mask1).updateMask(mask2);
};
// Load the MODIS NDVI data
var modisNDVI = ee.ImageCollection('MODIS/006/MCD43A4')
  .filterDate(startDate, endDate)
  .map(filterqa)
  .map(addNDVI);
// Function to extract daily NDVI values for each feature
var extractNDVI = function(image) {
  // Reduce the image collection by region
  var ndviValues = cornCoords.map(function(feature) {
    var geom = feature.geometry();
    var mean = image.reduceRegion({
      reducer: ee.Reducer.mean(),
      geometry: geom,
      scale: 500
    }).get('LSWI');
    return ee.Feature(null, {LSWI: mean});
  });
  var meanNDVI = ndviValues.reduceColumns(ee.Reducer.mean(), ['LSWI']).get('mean');
  return image.set('meanLSWI', meanNDVI);
};
// Map the function over the image collection
var dailyNDVI = modisNDVI.map(extractNDVI);
print(dailyNDVI)
// Convert to a feature collection for exporting
var ndviFeatureCollection = dailyNDVI.map(function(image) {
  return ee.Feature(null, {
    date: image.date().format('YYYY-MM-dd'),
    meanLSWI: image.get('meanLSWI')
```

```
  });
});
// Print the NDVI time series for verification
print(ndviFeatureCollection);
// Export the NDVI time series to Google Drive
Export.table.toDrive({
  collection: ndviFeatureCollection,
  description: 'Daily_Averaged_Corn_LSWI_TimeSeries_2022_NE',
  fileFormat: 'CSV'
});
```

Appendix B. Code for data preprocessing and data exploration

This code shows the preprocessing of data obtained at previous step (see section 3.1.4 and Appendix A) and POWER dataset, its transformation (see section 3.2.1) and its structuring (see section 3.2.2) for ML modelling.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pandas import Grouper
!pip install plotly_express
import plotly
import plotly_express as px
from scipy.signal import savgol_filter
import plotly_express as px
import plotly.graph_objects as go
data = pd.read_csv("iowa_story_city.csv")
"""###processing ndvi"""
fig = px.scatter(data, x = 'DOY', y='meanNDVI', opacity = 0.8,
size_max=30,hover_name='date',animation_frame='Year')
fig.show(renderer="colab")
```

```
ndvi_data = data.iloc[54:]

ndvi_data

from scipy.interpolate import UnivariateSpline

# Convert the list to a pandas Series

ndvi_series = pd.Series(ndvi_data.meanNDVI)

# Get the indices of non-missing values

valid_indices = np.where(~np.isnan(ndvi_series))[0]

# Get the corresponding NDVI values

valid_values = ndvi_series.dropna()

# Create a UnivariateSpline interpolation function

spline = UnivariateSpline(valid_indices, valid_values, k=3,s=0)  # s is the smoothing factor

# Generate the full range of indices

full_indices = np.arange(len(ndvi_series))

# Interpolate missing values

interpolated_values = spline(full_indices)

# Replace missing values in the original series with interpolated values

ndvi_series[np.isnan(ndvi_series)] = interpolated_values[np.isnan(ndvi_series)]

# Plot the original and interpolated NDVI values

plt.figure(figsize=(10, 5))

plt.plot(full_indices, ndvi_data.meanNDVI, 'o', label='Original Data (with NaNs)')

plt.plot(full_indices, ndvi_series, '-', label='Univariate Spline Interpolation')

plt.legend()

plt.xlabel('Time')

plt.ylabel('NDVI')

plt.title('Univariate Spline Interpolation of NDVI Data')

plt.show()

ndvi_data['NDVI_interpolated'] = ndvi_series

ndvi_data.isnull().sum()

fig = px.line(ndvi_data, x="DOY", y="NDVI_interpolated",
animation_frame="Year",color_discrete_sequence=['red'],range_x=[1,368])

fig
```

```
# Specify the list of months to drop
months_to_drop = [1, 2, 3,12]
# Drop rows where the 'month' column is in the specified list of months
ndvi_data = ndvi_data[~ndvi_data['Month'].isin(months_to_drop)]
from scipy.ndimage import median_filter
ndvi_series = pd.Series(ndvi_data.NDVI_interpolated)
# Step 2: Apply median filtering with a specified window size
window_size = 5  # Window size can be adjusted based on the level of smoothing needed
filtered_ndvi = median_filter(ndvi_series, size=window_size)
ndvi_data['NDVI_median_filtered'] = filtered_ndvi
line = px.line(ndvi_data, x="DOY", y="NDVI_median_filtered",
animation_frame="Year",color_discrete_sequence=['red'],range_x=[1,368])
scatter = px.scatter(ndvi_data, x="DOY", y="NDVI_interpolated", animation_frame="Year",range_x=[1,368])
# now integrate, traces, frames and layout
fig = go.Figure(
   data=line.data + scatter.data,
   frames=[
      go.Frame(data=fr1.data + fr2.data, name=fr1.name)
      for fr1, fr2 in zip(line.frames, scatter.frames)
   ],
   layout=scatter.layout,
)
fig
ndvi_series = pd.Series(ndvi_data.NDVI_median_filtered)
# Step 1: Fit a spline to the data
x = np.arange(len(ndvi_series))
spline = UnivariateSpline(x, ndvi_series, s=1)  # Initial fit with a smoothing factor
smoothed_ndvi = spline(x)
# Step 2: Calculate residuals
residuals = ndvi_series - smoothed_ndvi
# Step 3: Identify outliers
```

```
threshold = 3.0  # Threshold for identifying outliers (this value can be adjusted)
outliers = np.abs(residuals) > threshold
# Step 4: Remove outliers and refit spline
cleaned_x = x[~outliers]
cleaned_ndvi = ndvi_series[~outliers]
cleaned_spline = UnivariateSpline(cleaned_x, cleaned_ndvi, s=1)
cleaned_smoothed_ndvi = cleaned_spline(x)
ndvi_data['NDVI_smoothened'] = cleaned_smoothed_ndvi
line = px.line(ndvi_data, x="DOY", y="NDVI_smoothened",
animation_frame="Year",color_discrete_sequence=['red'],range_x=[1,368])
scatter = px.scatter(ndvi_data, x="DOY", y="NDVI_median_filtered", animation_frame="Year",range_x=[1,368])
# now integrate, traces, frames and layout
fig = go.Figure(
   data=line.data + scatter.data,
   frames=[
      go.Frame(data=fr1.data + fr2.data, name=fr1.name)
      for fr1, fr2 in zip(line.frames, scatter.frames)
   ],
   layout=scatter.layout,
)
fig
"""###processing lswi"""
fig = px.scatter(data, x = 'DOY', y='LSWI', opacity = 0.8, size_max=30,hover_name='date',animation_frame='Year')
fig.show(renderer="colab")
LSWI_data = data.iloc[54:]
from scipy.interpolate import UnivariateSpline
# Convert the list to a pandas Series
ndvi_series = pd.Series(ndvi_data.LSWI)
# Get the indices of non-missing values
valid_indices = np.where(~np.isnan(ndvi_series))[0]
# Get the corresponding NDVI values
```

```
valid_values = ndvi_series.dropna()

# Create a UnivariateSpline interpolation function

spline = UnivariateSpline(valid_indices, valid_values, k=3,s=0)  # s is the smoothing factor

# Generate the full range of indices

full_indices = np.arange(len(ndvi_series))

# Interpolate missing values

interpolated_values = spline(full_indices)

# Replace missing values in the original series with interpolated values

ndvi_series[np.isnan(ndvi_series)] = interpolated_values[np.isnan(ndvi_series)]

# Plot the original and interpolated NDVI values

plt.figure(figsize=(10, 5))

plt.plot(full_indices, ndvi_data.meanNDVI, 'o', label='Original Data (with NaNs)')

plt.plot(full_indices, ndvi_series, '-', label='Univariate Spline Interpolation')

plt.legend()

plt.xlabel('Time')

plt.ylabel('NDVI')

plt.title('Univariate Spline Interpolation of NDVI Data')

plt.show()

ndvi_data['LSWI_interpolated'] = ndvi_series

ndvi_data.isnull().sum()

fig = px.line(ndvi_data, x="DOY", y="LSWI_interpolated",
animation_frame="Year",color_discrete_sequence=['red'],range_x=[1,368])

fig

# Specify the list of months to drop

months_to_drop = [1, 2, 3,12]

# Drop rows where the 'month' column is in the specified list of months

ndvi_data = ndvi_data[~ndvi_data['Month'].isin(months_to_drop)]

fig = px.line(ndvi_data, x="DOY", y="LSWI_interpolated",
animation_frame="Year",color_discrete_sequence=['red'],range_x=[1,368])

fig

from scipy.ndimage import median_filter
```

```
ndvi_series = pd.Series(ndvi_data.LSWI_interpolated)
# Step 2: Apply median filtering with a specified window size
window_size = 5  # Window size can be adjusted based on the level of smoothing needed
filtered_ndvi = median_filter(ndvi_series, size=window_size)
ndvi_data['LSWI_median_filtered'] = filtered_ndvi
line = px.line(ndvi_data, x="DOY", y="LSWI_median_filtered",
animation_frame="Year",color_discrete_sequence=['red'],range_x=[1,368])
scatter = px.scatter(ndvi_data, x="DOY", y="LSWI_interpolated", animation_frame="Year",range_x=[1,368])
# now integrate, traces, frames and layout
fig = go.Figure(
   data=line.data + scatter.data,
   frames=[
      go.Frame(data=fr1.data + fr2.data, name=fr1.name)
      for fr1, fr2 in zip(line.frames, scatter.frames)
   ],
   layout=scatter.layout,
)
fig
ndvi_series = pd.Series(ndvi_data.LSWI_median_filtered)
# Step 1: Fit a spline to the data
x = np.arange(len(ndvi_series))
spline = UnivariateSpline(x, ndvi_series, s=1)  # Initial fit with a smoothing factor
smoothed_ndvi = spline(x)
# Step 2: Calculate residuals
residuals = ndvi_series - smoothed_ndvi
# Step 3: Identify outliers
threshold = 3.0  # Threshold for identifying outliers (this value can be adjusted)
outliers = np.abs(residuals) > threshold
# Step 4: Remove outliers and refit spline
cleaned_x = x[~outliers]
cleaned_ndvi = ndvi_series[~outliers]
```

```
cleaned_spline = UnivariateSpline(cleaned_x, cleaned_ndvi, s=1)

cleaned_smoothed_ndvi = cleaned_spline(x)

ndvi_data['LSWI_smoothened'] = cleaned_smoothed_ndvi

line = px.line(ndvi_data, x="DOY", y="LSWI_smoothened",
animation_frame="Year",color_discrete_sequence=['red'],range_x=[1,368])

scatter = px.scatter(ndvi_data, x="DOY", y="LSWI_median_filtered", animation_frame="Year",range_x=[1,368])

# now integrate, traces, frames and layout

fig = go.Figure(

    data=line.data + scatter.data,

    frames=[

        go.Frame(data=fr1.data + fr2.data, name=fr1.name)

        for fr1, fr2 in zip(line.frames, scatter.frames)

    ],

    layout=scatter.layout,

)

fig

"""###calculation of gdd & chu

"""

ndvi_data = ndvi_data[ndvi_data['Stage']!='NA']

def calculate_gdd(row, base_temp=10):

    # Ensure Tmin and Tmax are within the specified range (10°C to 30°C)

    Tmax = min(max(row['T2M_MAX'], base_temp), 30)

    Tmin = min(max(row['T2M_MIN'], base_temp), 30)

    # Calculate GDD

    GDD = ((Tmax + Tmin) / 2) - base_temp

    return max(GDD, 0)  # Ensure GDD is not negative

# Apply the calculate_gdd function to each row

ndvi_data['GDD1030'] = ndvi_data.apply(calculate_gdd, axis=1)

ndvi_data

def calculate_daytime_chu(Tmax):

    if Tmax < 10:
```

```
        Tmax = 10
    return (3.33 * (Tmax - 10) - 0.084 * (Tmax - 10) ** 2) / 2
def calculate_nighttime_chu(Tmin):
    if Tmin < 4.4:
        Tmin = 4.4
    return (1.8 * (Tmin - 4.4)) / 2
def calculate_chu(row):
    Tmax = row['T2M_MAX']
    Tmin = row['T2M_MIN']
    # Calculate daytime and nighttime CHUs
    CHU_day = calculate_daytime_chu(Tmax)
    CHU_night = calculate_nighttime_chu(Tmin)
    # Total CHU
    CHU = CHU_day + CHU_night
    return CHU
# Apply the calculate_chu function to each row
ndvi_data['CHU'] = ndvi_data.apply(calculate_chu, axis=1)
ndvi_data
"""###calculating  cumsum for gdd and chu"""
ndvi_data['Stage'] = ndvi_data['Stage'].fillna('NA')
ndvi_data = ndvi_data[ndvi_data.Stage != 'NA']
ndvi_data
ndvi_data['sum_GDD1030'] = ndvi_data.groupby('Year')['GDD1030'].cumsum()
ndvi_data['sum_CHU'] = ndvi_data.groupby('Year')['CHU'].cumsum()
ndvi_data
ndvi_data.to_csv('ndvi_data_NE.csv')
ndvi_data.to_csv('ndvi_data_central.csv', index=False)
f = pd.read_csv('ndvi_data_central.csv', parse_dates=['date'])


df['Stage'] = df['Stage'].fillna('NA')
```

```
df = df[df.Stage != 'NA']
df = df[df.Stage != 'Harvested']
#df = df[df.Stage != 'Planted']
df.Stage.unique()
# Encode the growth stages as integers
label_encoder = LabelEncoder()
df['growth_stage'] = label_encoder.fit_transform(df['Stage'])
# Features and target variable
features = ['NDVI_smoothened','sum_GDD1030','sum_CHU','LSWI_smoothened']
transform_features = ['sum_GDD1030','sum_CHU']
target = 'growth_stage'
# Scale the features
scaler = StandardScaler()
df[features] = scaler.fit_transform(df[features])
train_data = df[(df['date'] < '2017-01-01')]
test_data = df[(df['date'] >= '2017-01-01')]
X_train = train_data[features].values
y_train = train_data[target].values
dates_train = train_data['date'].values
X_test = test_data[features].values
y_test = test_data[target].values
dates_test = test_data['date'].values
train_data
test_data
# Convert target variable to categorical
from tensorflow.keras.utils import to_categorical
y_train_categorical = to_categorical(y_train)
y_test_categorical = to_categorical(y_test)
timesteps = 1
num_features = 4
```

```
X_train_reshaped = X_train.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test_reshaped = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))
y_train_reshaped = y_train_categorical.reshape((y_train_categorical.shape[0], 1, y_train_categorical.shape[1]))
y_test_reshaped = y_test_categorical.reshape((y_test_categorical.shape[0], 1, y_test_categorical.shape[1]))
```

Appendix C. Code for Machine learning experimentation

C.1. TMANet Model

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.layers import Layer, Bidirectional, LSTM, Dense, Input, GlobalAveragePooling1D, Multiply
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
# Positional Encoding Layer
class PositionalEncoding(Layer):
    def __init__(self, max_len, d_model):
        super(PositionalEncoding, self).__init__()
        self.positional_encoding = self.get_positional_encoding(max_len, d_model)

    def get_positional_encoding(self, max_len, d_model):
        # Create a matrix of shape (max_len, d_model)
        pos = np.arange(max_len)[:, np.newaxis]  # Shape: (max_len, 1)
        i = np.arange(d_model)[np.newaxis, :]  # Shape: (1, d_model)

        # Compute the positional encoding
        angle_rates = 1 / np.power(10000, (2 * (i // 2)) / np.float32(d_model))
        angle_rads = pos * angle_rates  # Shape: (max_len, d_model)

        # Apply sine to even index and cosine to odd index
        angle_rads[:, 0::2] = np.sin(angle_rads[:, 0::2])
        angle_rads[:, 1::2] = np.cos(angle_rads[:, 1::2])
```

```
        return tf.cast(angle_rads[np.newaxis, ...], dtype=tf.float32)  # Shape: (1, max_len, d_model)

    def call(self, inputs):
        seq_len = tf.shape(inputs)[1]
        return inputs + self.positional_encoding[:, :seq_len, :]  # Add encoding to inputs

# Spatial Attention Layer
class SpatialAttentionLayer(Layer):
    def __init__(self, feature_dim):
        super(SpatialAttentionLayer, self).__init__()
        self.dense = Dense(feature_dim, activation='softmax')

    def call(self, inputs):
        avg_pool = GlobalAveragePooling1D()(inputs)  # Shape: (None, features)
        attention_weights = self.dense(avg_pool)  # Shape: (None, features)
        attention_weights = tf.reshape(attention_weights, (-1, 1, inputs.shape[-1]))  # Shape: (None, 1, features)
        attended_output = Multiply()([inputs, attention_weights])  # Shape: (None, timesteps, features)
        return attended_output, attention_weights

# Temporal Attention Layer
class TemporalAttentionLayer(Layer):
    def __init__(self):
        super(TemporalAttentionLayer, self).__init__()

    def call(self, inputs):
        attention_scores = tf.nn.softmax(tf.reduce_mean(inputs, axis=-1), axis=1)  # Shape: (None, timesteps)
        attention_scores = tf.expand_dims(attention_scores, axis=-1)  # Shape: (None, timesteps, 1)
        attended_output = Multiply()([inputs, attention_scores])  # Shape: (None, timesteps, features)
        return attended_output, attention_scores
```

```
# Complete LSTM Model with Spatial, Temporal Attention, and Positional Encoding
def build_lstm_model_with_attention(input_shape, num_classes):
    inputs = Input(shape=input_shape)

    # Add Positional Encoding
    max_len = input_shape[0]  # Number of timesteps
    d_model = input_shape[1]  # Number of features
    pos_encoding = PositionalEncoding(max_len, d_model)(inputs)

    # Spatial Attention
    spatial_out, spatial_attention_weights = SpatialAttentionLayer(input_shape[-1])(pos_encoding)

    # Bidirectional LSTM Layer
    lstm_out = Bidirectional(LSTM(128, return_sequences=True))(spatial_out)  # Return sequences=True for temporal attention

    # Temporal Attention
    temporal_out, temporal_attention_weights = TemporalAttentionLayer()(lstm_out)

    # Global Average Pooling after Temporal Attention
    global_avg_out = GlobalAveragePooling1D()(temporal_out)

    # Fully Connected Output Layer
    outputs = Dense(num_classes, activation='softmax')(global_avg_out)

    # Build model
    model = Model(inputs=inputs, outputs=[outputs, spatial_attention_weights, temporal_attention_weights])
    return model

# Example input shape and number of classes
input_shape = (1, 4)  # 30 timesteps, 4 features
```

```
num_classes = 6


# Build and compile the model
model = build_lstm_model_with_attention(input_shape, num_classes)
model.compile(optimizer=Adam(learning_rate = 0.001), loss='categorical_crossentropy',
metrics=['accuracy',None,None])
model.summary()
y_train_reshaped=y_train_reshaped.reshape((-1, 6))


y_test_reshaped=y_test_reshaped.reshape((-1, 6))


y_train_reshaped.shape


# Train the model
history = model.fit(X_train_reshaped, y_train_reshaped, epochs=100, batch_size=64)


# Evaluate the model on the test data
test_loss, test_accuracy = model.evaluate(X_test_reshaped, y_test_reshaped)
print(f'Test Loss: {test_loss}')
print(f'Test Accuracy: {test_accuracy}')


import pickle
pickle.dump(model, open('tmanet_central.sav', 'wb'))


# Predictions on test set
y_pred, multivariate_attention_weights, temporal_attention_weights= model.predict(X_test_reshaped)


#y_pred_classes = np.argmax(y_pred, axis=1)


y_pred_classes = np.argmax(y_pred, axis=1)
```

```
y_pred_classes = y_pred_classes.reshape(-1)

y_pred_classes

y_true_classes = np.argmax(y_test_categorical, axis=1)

# Convert predictions and true labels back to original labels
y_pred_labels_decoded = label_encoder.inverse_transform(y_pred_classes)
y_test_labels_decoded = label_encoder.inverse_transform(y_true_classes)

# Evaluate the model
accuracy = (y_pred_labels_decoded == y_test_labels_decoded).mean()
print(f'Accuracy: {accuracy}')

# You can also print a classification report for more detailed evaluation
from sklearn.metrics import classification_report
print(classification_report(y_test_labels_decoded, y_pred_labels_decoded))

multivariate_attention_weights.shape

import numpy as np

def compute_feature_importance_per_class(model, inputs):
    # Get model outputs (including attention weights)
    preds, multivariate_attn_weights, temporal_attn_weights = model.predict(inputs)

    # Store feature importance per class
    class_feature_importance = {}

    # Iterate over each input sample
```

```
for i in range(inputs.shape[0]):
    # Get the predicted class for the sample
    predicted_class = np.argmax(preds[i])

    # Extract the multivariate attention weights for the current sample (shape: timesteps, features)
    multivariate_weights = multivariate_attn_weights[i]

    # Extract the temporal attention weights for the current sample (shape: timesteps, 1)
    temporal_weights = temporal_attn_weights[i]

    # Reshape temporal weights to broadcast across features
    temporal_weights = np.reshape(temporal_weights, (-1, 1))

    # Calculate combined attention weights (temporal * multivariate)
    combined_attention = multivariate_weights * temporal_weights

    # Sum combined attention over timesteps to get importance per feature
    feature_importance = np.sum(combined_attention, axis=0)  # shape: (features,)

    # Add feature importance for this sample to the corresponding class
    if predicted_class not in class_feature_importance:
        class_feature_importance[predicted_class] = []
    class_feature_importance[predicted_class].append(feature_importance)

# Average the feature importance for each class
for class_label in class_feature_importance:
    class_feature_importance[class_label] = np.mean(class_feature_importance[class_label], axis=0)

return class_feature_importance
```

```
# Get the feature importance per class
class_feature_importance = compute_feature_importance_per_class(model, X_test_reshaped)
l = np.array(list(class_feature_importance.keys()))
l = label_encoder.inverse_transform(l)
print(l)
i=0
# Print feature importance for each class
for class_label, importance in class_feature_importance.items():
    print(f"Class {l[i]} feature importance:")
    print(f"NDVI: {importance[0]}, GDD: {importance[1]}, CHU: {importance[2]}, LSWI: {importance[3]}")
    print(f'sum: {sum(importance)}')
    i=i+1

results = pd.DataFrame({
    'date': test_data['date'],
    'year': test_data['Year'],
    'doy': test_data['DOY'],
    'ndvi': test_data['NDVI_smoothened'],
    'gdd': test_data['sum_GDD1030'],
    'CHU': test_data['sum_CHU'],
    'T2M_MAX': test_data['T2M_MAX'],
    'T2M_MIN': test_data['T2M_MIN'],
    'actual': test_data['Stage'],
    'predicted': y_pred_labels_decoded
})

# Mark correct and incorrect predictions
results['correct'] = results['actual'] == results['predicted']

results
```

```
incorrect = results[results.correct == False]


incorrect


line = px.scatter(results, x = 'doy', y='ndvi',hover_name='doy',animation_frame='year', color='actual',size_max=100)

scatter = px.scatter(incorrect, x='doy', y='ndvi',
animation_frame='year',color_discrete_sequence=['black'],size_max=10,symbol_sequence=['circle-
open'],hover_name='predicted')

# now integrate, traces, frames and layout

fig = go.Figure(

   data=line.data + scatter.data,

   frames=[

      go.Frame(data=fr1.data + fr2.data, name=fr1.name)

      for fr1, fr2 in zip(line.frames, scatter.frames)

   ],

   layout=scatter.layout,

)

fig.update_layout(yaxis_range=[-2,2])

fig


def get_first_occurrence(df, stage_col, date_col):

   first_occurrence = df.groupby(['year', stage_col])[date_col].min().reset_index()

   first_occurrence['doy'] = first_occurrence[date_col].dt.dayofyear

   return first_occurrence.pivot(index='year', columns=stage_col, values='doy')


# Get first occurrence of each stage for actual and predicted stages

first_occurrence_actual = get_first_occurrence(results, 'actual', 'date')

first_occurrence_predicted = get_first_occurrence(results, 'predicted', 'date')


# Flatten the dataframes to compare DOY values directly
```

```
actual_doy = first_occurrence_actual.values.flatten()
predicted_doy = first_occurrence_predicted.values.flatten()

# Remove NaN values if any
mask = ~np.isnan(actual_doy) & ~np.isnan(predicted_doy)
actual_doy = actual_doy[mask]
predicted_doy = predicted_doy[mask]

# Calculate RMSE
rmse = np.sqrt(mean_squared_error(actual_doy, predicted_doy))
print(f'RMSE of DOY between actual and predicted stages: {rmse}')

# Calculate RMSE
rmse = np.sqrt(mean_absolute_error(actual_doy, predicted_doy))
print(f'MAE of DOY between actual and predicted stages: {rmse}')

# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
```

```
    rmse_stage = np.sqrt(mean_squared_error(actual_doy_stage, predicted_doy_stage))
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'RMSE for stage {stage}: {rmse}')

from sklearn.metrics import mean_absolute_error
# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = mean_absolute_error(actual_doy_stage, predicted_doy_stage)
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'MAE for stage {stage}: {rmse}')
```

C.2. Comparison with state-of-the-art ML models

```
"""##LSTM"""

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Bidirectional

# Build the LSTM model
model = Sequential()
model.add(LSTM(128, activation='relu', input_shape=(1, len(features))))
model.add(Dense(y_train_categorical.shape[1], activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train_reshaped, y_train_categorical, epochs=100, batch_size=64)

# Predict the growth stages
y_pred = model.predict(X_test_reshaped)
y_pred_labels = np.argmax(y_pred, axis=1)

# Convert predictions and true labels back to original labels
y_pred_labels_decoded = label_encoder.inverse_transform(y_pred_labels)
y_test_labels_decoded = label_encoder.inverse_transform(np.argmax(y_test_categorical, axis=1))

# Evaluate the model
accuracy = (y_pred_labels_decoded == y_test_labels_decoded).mean()
print(f'Accuracy: {accuracy}')

# You can also print a classification report for more detailed evaluation
from sklearn.metrics import classification_report
print(classification_report(y_test_labels_decoded, y_pred_labels_decoded))

results = pd.DataFrame({
    'date': test_data['date'],
```

```
    'year': test_data['Year'],
    'doy': test_data['DOY'],
    'ndvi': test_data['NDVI_smoothened'],
    'gdd': test_data['sum_GDD1030'],
    'CHU': test_data['sum_CHU'],
    'T2M_MAX': test_data['T2M_MAX'],
    'T2M_MIN': test_data['T2M_MIN'],
    'actual': test_data['Stage'],
    'predicted': y_pred_labels_decoded
})

# Mark correct and incorrect predictions
results['correct'] = results['actual'] == results['predicted']


line = px.scatter(results, x = 'doy', y='ndvi',hover_name='doy',animation_frame='year', color='actual',size_max=100)
scatter = px.scatter(incorrect, x='doy', y='ndvi',
animation_frame='year',color_discrete_sequence=['black'],size_max=10,symbol_sequence=['circle-
open'],hover_name='predicted')
# now integrate, traces, frames and layout
fig = go.Figure(
    data=line.data + scatter.data,
    frames=[
        go.Frame(data=fr1.data + fr2.data, name=fr1.name)
        for fr1, fr2 in zip(line.frames, scatter.frames)
    ],
    layout=scatter.layout,
)
fig.update_layout(yaxis_range=[-2,2])
fig


import numpy as np
from sklearn.metrics import accuracy_score

# Function to calculate permutation feature importance
```

```
def permutation_feature_importance(model, X, y, metric=accuracy_score, n_repeats=10):
    baseline_score = metric(y, np.argmax(model.predict(X), axis=1))
    importances = np.zeros(X.shape[2])

    for i in range(X.shape[2]):
        scores = np.zeros(n_repeats)
        for n in range(n_repeats):
            X_permuted = X.copy()
            np.random.shuffle(X_permuted[:, :, i])
            scores[n] = metric(y, np.argmax(model.predict(X_permuted), axis=1))
        importances[i] = baseline_score - np.mean(scores)

    return importances

# Calculate permutation feature importance
feature_importances = permutation_feature_importance(model, X_test_reshaped, y_test)
feature_names = features

# Print feature importances
for feature, importance in zip(feature_names, feature_importances):
    print(f'Feature: {feature}, Importance: {importance}')

def get_first_occurrence(df, stage_col, date_col):
    first_occurrence = df.groupby(['year', stage_col])[date_col].min().reset_index()
    first_occurrence['doy'] = first_occurrence[date_col].dt.dayofyear
    return first_occurrence.pivot(index='year', columns=stage_col, values='doy')

# Get first occurrence of each stage for actual and predicted stages
first_occurrence_actual = get_first_occurrence(results, 'actual', 'date')
first_occurrence_predicted = get_first_occurrence(results, 'predicted', 'date')

# Flatten the dataframes to compare DOY values directly
actual_doy = first_occurrence_actual.values.flatten()
```

```
predicted_doy = first_occurrence_predicted.values.flatten()

# Remove NaN values if any
mask = ~np.isnan(actual_doy) & ~np.isnan(predicted_doy)
actual_doy = actual_doy[mask]
predicted_doy = predicted_doy[mask]

# Calculate RMSE
rmse = np.sqrt(mean_squared_error(actual_doy, predicted_doy))
print(f'RMSE of DOY between actual and predicted stages: {rmse}')

# Calculate RMSE
rmse = np.sqrt(mean_absolute_error(actual_doy, predicted_doy))
print(f'MAE of DOY between actual and predicted stages: {rmse}')

# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = np.sqrt(mean_squared_error(actual_doy_stage, predicted_doy_stage))
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
```

```
for stage, rmse in rmse_per_stage.items():
    print(f'RMSE for stage {stage}: {rmse}')

from sklearn.metrics import mean_absolute_error
# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = mean_absolute_error(actual_doy_stage, predicted_doy_stage)
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'MAE for stage {stage}: {rmse}')

"""##GRU"""

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Bidirectional

# Build the LSTM model
model = Sequential()
model.add(GRU(128, activation='relu', input_shape=(1, len(features))))
```

```
model.add(Dense(y_train_categorical.shape[1], activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train_reshaped, y_train_categorical, epochs=100, batch_size=64)

# Predict the growth stages
y_pred = model.predict(X_test_reshaped)
y_pred_labels = np.argmax(y_pred, axis=1)

# Convert predictions and true labels back to original labels
y_pred_labels_decoded = label_encoder.inverse_transform(y_pred_labels)
y_test_labels_decoded = label_encoder.inverse_transform(np.argmax(y_test_categorical, axis=1))

# Evaluate the model
accuracy = (y_pred_labels_decoded == y_test_labels_decoded).mean()
print(f'Accuracy: {accuracy}')

# You can also print a classification report for more detailed evaluation
from sklearn.metrics import classification_report
print(classification_report(y_test_labels_decoded, y_pred_labels_decoded))

results = pd.DataFrame({
    'date': test_data['date'],
    'year': test_data['Year'],
    'doy': test_data['DOY'],
    'ndvi': test_data['NDVI_smoothened'],
    'gdd': test_data['sum_GDD1030'],
    'CHU': test_data['sum_CHU'],
    'T2M_MAX': test_data['T2M_MAX'],
    'T2M_MIN': test_data['T2M_MIN'],
    'actual': test_data['Stage'],
    'predicted': y_pred_labels_decoded
```

```
})

# Mark correct and incorrect predictions
results['correct'] = results['actual'] == results['predicted']

line = px.scatter(results, x = 'doy', y='ndvi',hover_name='doy',animation_frame='year', color='actual',size_max=100)
scatter = px.scatter(incorrect, x='doy', y='ndvi',
animation_frame='year',color_discrete_sequence=['black'],size_max=10,symbol_sequence=['circle-
open'],hover_name='predicted')
# now integrate, traces, frames and layout
fig = go.Figure(
    data=line.data + scatter.data,
    frames=[
        go.Frame(data=fr1.data + fr2.data, name=fr1.name)
        for fr1, fr2 in zip(line.frames, scatter.frames)
    ],
    layout=scatter.layout,
)
fig.update_layout(yaxis_range=[-2,2])
fig

import numpy as np
from sklearn.metrics import accuracy_score

# Function to calculate permutation feature importance
def permutation_feature_importance(model, X, y, metric=accuracy_score, n_repeats=10):
    baseline_score = metric(y, np.argmax(model.predict(X), axis=1))
    importances = np.zeros(X.shape[2])

    for i in range(X.shape[2]):
        scores = np.zeros(n_repeats)
        for n in range(n_repeats):
            X_permuted = X.copy()
            np.random.shuffle(X_permuted[:, :, i])
```

```
            scores[n] = metric(y, np.argmax(model.predict(X_permuted), axis=1))
        importances[i] = baseline_score - np.mean(scores)

    return importances

# Calculate permutation feature importance
feature_importances = permutation_feature_importance(model, X_test_reshaped, y_test)
feature_names = features

# Print feature importances
for feature, importance in zip(feature_names, feature_importances):
    print(f'Feature: {feature}, Importance: {importance}')

def get_first_occurrence(df, stage_col, date_col):
    first_occurrence = df.groupby(['year', stage_col])[date_col].min().reset_index()
    first_occurrence['doy'] = first_occurrence[date_col].dt.dayofyear
    return first_occurrence.pivot(index='year', columns=stage_col, values='doy')

# Get first occurrence of each stage for actual and predicted stages
first_occurrence_actual = get_first_occurrence(results, 'actual', 'date')
first_occurrence_predicted = get_first_occurrence(results, 'predicted', 'date')

# Flatten the dataframes to compare DOY values directly
actual_doy = first_occurrence_actual.values.flatten()
predicted_doy = first_occurrence_predicted.values.flatten()

# Remove NaN values if any
mask = ~np.isnan(actual_doy) & ~np.isnan(predicted_doy)
actual_doy = actual_doy[mask]
predicted_doy = predicted_doy[mask]

# Calculate RMSE
rmse = np.sqrt(mean_squared_error(actual_doy, predicted_doy))
```

```
print(f'RMSE of DOY between actual and predicted stages: {rmse}')

# Calculate RMSE
rmse = np.sqrt(mean_absolute_error(actual_doy, predicted_doy))
print(f'MAE of DOY between actual and predicted stages: {rmse}')

# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = np.sqrt(mean_squared_error(actual_doy_stage, predicted_doy_stage))
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'RMSE for stage {stage}: {rmse}')

from sklearn.metrics import mean_absolute_error
# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
```

```
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = mean_absolute_error(actual_doy_stage, predicted_doy_stage)
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'MAE for stage {stage}: {rmse}')

"""##Random Forest"""

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

target="Stage"

X_train = train_data[features].values
y_train = train_data[target].values
dates_train = train_data['date'].values
X_test = test_data[features].values
y_test = test_data[target].values
dates_test = test_data['date'].values

RFC = RandomForestClassifier(n_estimators = 200, random_state = 42)
RFC.fit(X_train , y_train)
RFC_preds = RFC.predict(X_test)
```

```
print('Classification Report:\n',classification_report(y_test , RFC_preds))
print('Accuracy Score:',accuracy_score(y_test , RFC_preds)*100)

import matplotlib.pyplot as plt
feature_importances = RFC.feature_importances_

# Create a DataFrame for better visualization
features_df = pd.DataFrame({
   'Feature': features,
   'Importance': feature_importances
}).sort_values(by='Importance', ascending=False)

# Print feature importance
print(features_df)

# Optional: Plot feature importance
plt.figure(figsize=(10, 6))
plt.barh(features_df['Feature'], features_df['Importance'], color='skyblue')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance from Decision Tree')
plt.gca().invert_yaxis()  # Invert y-axis to show the most important feature on top
plt.show()

test=test_data
results = pd.DataFrame({
   'date': test['date'],
   'year': test['Year'],
   'doy': test['DOY'],
   'ndvi': test['NDVI_smoothened'],
   'gdd': test['sum_GDD1030'],
   'CHU': test['sum_CHU'],
   'T2M_MAX': test['T2M_MAX'],
```

```
    'T2M_MIN': test['T2M_MIN'],
    'actual': test['Stage'],
    'predicted': RFC_preds
})

# Mark correct and incorrect predictions
results['correct'] = results['actual'] == results['predicted']

results

incorrect = results[results.correct == False]

line = px.scatter(results, x = 'doy', y='ndvi',hover_name='doy',animation_frame='year', color='actual',size_max=100)
scatter = px.scatter(incorrect, x='doy', y='ndvi',
animation_frame='year',color_discrete_sequence=['black'],size_max=10,symbol_sequence=['circle-
open'],hover_name='predicted')
# now integrate, traces, frames and layout
fig = go.Figure(
    data=line.data + scatter.data,
    frames=[
        go.Frame(data=fr1.data + fr2.data, name=fr1.name)
        for fr1, fr2 in zip(line.frames, scatter.frames)
    ],
    layout=scatter.layout,
)
fig.update_layout(yaxis_range=[-2,2])
fig

growth_stages = ['Planted', 'Emerged', 'Silked', 'Dough', 'Dent', 'Mature']

# Map the stages to their position in the list
stage_mapping = {stage: idx for idx, stage in enumerate(growth_stages)}

def enforce_sequential_constraint(df, stages):
```

```
    # Apply the logic year by year
    df['stage_order'] = df['predicted'].map(stage_mapping)  # Assign a numerical order to stages
    for year in df['year'].unique():
        year_data = df[df['year'] == year]
        current_stage = -1  # Initialize to a stage before 'Planted'

        # Iterate through the predictions of the year
        for idx in year_data.index:
            predicted_stage = df.at[idx, 'stage_order']
            # If the predicted stage is earlier than the current stage, enforce the current stage
            if predicted_stage < current_stage:
                df.at[idx, 'stage_order'] = current_stage  # Enforce the latest stage
            else:
                current_stage = predicted_stage  # Update the current stage

        # Map the corrected numerical stage back to the original stage name
        df.loc[year_data.index, 'predictions'] = df.loc[year_data.index, 'stage_order'].map(
            {v: k for k, v in stage_mapping.items()}
        )

    # Drop the stage_order column used for processing
    df.drop(columns=['stage_order'], inplace=True)
    return df

# Apply the function to enforce the sequential constraint
df = enforce_sequential_constraint(results, growth_stages)

df

df['correct'] = df['actual'] == df['predictions']

incorrect = df[df.correct == False]
```

```
line = px.scatter(df, x = 'doy', y='ndvi',hover_name='doy',animation_frame='year', color='actual',size_max=100)
scatter = px.scatter(incorrect, x='doy', y='ndvi',
animation_frame='year',color_discrete_sequence=['black'],size_max=10,symbol_sequence=['circle-
open'],hover_name='predicted')
# now integrate, traces, frames and layout
fig = go.Figure(
   data=line.data + scatter.data,
   frames=[
      go.Frame(data=fr1.data + fr2.data, name=fr1.name)
      for fr1, fr2 in zip(line.frames, scatter.frames)
   ],
   layout=scatter.layout,
)
fig.update_layout(yaxis_range=[-2,2])
fig

print('Classification Report:\n',classification_report(df.actual , df.predictions))
print('Accuracy Score:',accuracy_score(df.actual , df.predictions)*100)

results=df

def get_first_occurrence(df, stage_col, date_col):
   first_occurrence = df.groupby(['year', stage_col])[date_col].min().reset_index()
   first_occurrence['doy'] = first_occurrence[date_col].dt.dayofyear
   return first_occurrence.pivot(index='year', columns=stage_col, values='doy')

# Get first occurrence of each stage for actual and predicted stages
first_occurrence_actual = get_first_occurrence(results, 'actual', 'date')
first_occurrence_predicted = get_first_occurrence(results, 'predictions', 'date')

# Flatten the dataframes to compare DOY values directly
actual_doy = first_occurrence_actual.values.flatten()
predicted_doy = first_occurrence_predicted.values.flatten()
```

```
# Remove NaN values if any
mask = ~np.isnan(actual_doy) & ~np.isnan(predicted_doy)
actual_doy = actual_doy[mask]
predicted_doy = predicted_doy[mask]

# Calculate RMSE
rmse = np.sqrt(mean_squared_error(actual_doy, predicted_doy))
print(f'RMSE of DOY between actual and predicted stages: {rmse}')

# Calculate RMSE
rmse = np.sqrt(mean_absolute_error(actual_doy, predicted_doy))
print(f'MAE of DOY between actual and predicted stages: {rmse}')

# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = np.sqrt(mean_squared_error(actual_doy_stage, predicted_doy_stage))
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'RMSE for stage {stage}: {rmse}')
```

```
from sklearn.metrics import mean_absolute_error
# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = mean_absolute_error(actual_doy_stage, predicted_doy_stage)
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'MAE for stage {stage}: {rmse}')

"""##SVR"""

from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

target="Stage"

X_train = train_data[features].values
y_train = train_data[target].values
dates_train = train_data['date'].values
```

```
X_test = test_data[features].values
y_test = test_data[target].values
dates_test = test_data['date'].values

# Support Vector Machine
svm = SVC(probability=True, kernel='rbf',)
svm.fit(X_train , y_train)
svm_preds = svm.predict(X_test)
print('Classification Report:\n',classification_report(y_test , svm_preds))
print('Accuracy Score:',accuracy_score(y_test , svm_preds)*100)

test=test_data
results = pd.DataFrame({
    'date': test['date'],
    'year': test['Year'],
    'doy': test['DOY'],
    'ndvi': test['NDVI_smoothened'],
    'gdd': test['sum_GDD1030'],
    'CHU': test['sum_CHU'],
    'T2M_MAX': test['T2M_MAX'],
    'T2M_MIN': test['T2M_MIN'],
    'actual': test['Stage'],
    'predicted': svm_preds
})

# Mark correct and incorrect predictions
results['correct'] = results['actual'] == results['predicted']

results

incorrect = results[results.correct == False]

line = px.scatter(results, x = 'doy', y='ndvi',hover_name='doy',animation_frame='year', color='actual',size_max=100)
```

```
scatter = px.scatter(incorrect, x='doy', y='ndvi',
animation_frame='year',color_discrete_sequence=['black'],size_max=10,symbol_sequence=['circle-
open'],hover_name='predicted')

# now integrate, traces, frames and layout

fig = go.Figure(

   data=line.data + scatter.data,

   frames=[

      go.Frame(data=fr1.data + fr2.data, name=fr1.name)

      for fr1, fr2 in zip(line.frames, scatter.frames)

   ],

   layout=scatter.layout,

)

fig.update_layout(yaxis_range=[-2,2])

fig


growth_stages = ['Planted', 'Emerged', 'Silked', 'Dough', 'Dent', 'Mature']


# Map the stages to their position in the list

stage_mapping = {stage: idx for idx, stage in enumerate(growth_stages)}


def enforce_sequential_constraint(df, stages):

   # Apply the logic year by year

   df['stage_order'] = df['predicted'].map(stage_mapping)  # Assign a numerical order to stages

   for year in df['year'].unique():

      year_data = df[df['year'] == year]

      current_stage = -1  # Initialize to a stage before 'Planted'


      # Iterate through the predictions of the year

      for idx in year_data.index:

         predicted_stage = df.at[idx, 'stage_order']

         # If the predicted stage is earlier than the current stage, enforce the current stage

         if predicted_stage < current_stage:

            df.at[idx, 'stage_order'] = current_stage  # Enforce the latest stage

         else:
```

```
            current_stage = predicted_stage  # Update the current stage

        # Map the corrected numerical stage back to the original stage name
        df.loc[year_data.index, 'predictions'] = df.loc[year_data.index, 'stage_order'].map(
            {v: k for k, v in stage_mapping.items()}
        )

    # Drop the stage_order column used for processing
    df.drop(columns=['stage_order'], inplace=True)
    return df

# Apply the function to enforce the sequential constraint
df = enforce_sequential_constraint(results, growth_stages)

df

df['correct'] = df['actual'] == df['predictions']

incorrect = df[df.correct == False]

line = px.scatter(df, x = 'doy', y='ndvi',hover_name='doy',animation_frame='year', color='actual',size_max=100)
scatter = px.scatter(incorrect, x='doy', y='ndvi',
animation_frame='year',color_discrete_sequence=['black'],size_max=10,symbol_sequence=['circle-
open'],hover_name='predicted')
# now integrate, traces, frames and layout
fig = go.Figure(
    data=line.data + scatter.data,
    frames=[
        go.Frame(data=fr1.data + fr2.data, name=fr1.name)
        for fr1, fr2 in zip(line.frames, scatter.frames)
    ],
    layout=scatter.layout,
)
fig.update_layout(yaxis_range=[-2,2])
```

```
fig

print('Classification Report:\n',classification_report(df.actual , df.predictions))
print('Accuracy Score:',accuracy_score(df.actual , df.predictions)*100)

results=df

def get_first_occurrence(df, stage_col, date_col):
    first_occurrence = df.groupby(['year', stage_col])[date_col].min().reset_index()
    first_occurrence['doy'] = first_occurrence[date_col].dt.dayofyear
    return first_occurrence.pivot(index='year', columns=stage_col, values='doy')

# Get first occurrence of each stage for actual and predicted stages
first_occurrence_actual = get_first_occurrence(results, 'actual', 'date')
first_occurrence_predicted = get_first_occurrence(results, 'predictions', 'date')

# Flatten the dataframes to compare DOY values directly
actual_doy = first_occurrence_actual.values.flatten()
predicted_doy = first_occurrence_predicted.values.flatten()

# Remove NaN values if any
mask = ~np.isnan(actual_doy) & ~np.isnan(predicted_doy)
actual_doy = actual_doy[mask]
predicted_doy = predicted_doy[mask]

# Calculate RMSE
rmse = np.sqrt(mean_squared_error(actual_doy, predicted_doy))
print(f'RMSE of DOY between actual and predicted stages: {rmse}')

# Calculate RMSE
rmse = np.sqrt(mean_absolute_error(actual_doy, predicted_doy))
print(f'MAE of DOY between actual and predicted stages: {rmse}')
```

```
# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = np.sqrt(mean_squared_error(actual_doy_stage, predicted_doy_stage))
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'RMSE for stage {stage}: {rmse}')

from sklearn.metrics import mean_absolute_error
# Ensure both dataframes have the same columns (growth stages)
common_stages = first_occurrence_actual.columns.intersection(first_occurrence_predicted.columns)
first_occurrence_actual = first_occurrence_actual[common_stages]
first_occurrence_predicted = first_occurrence_predicted[common_stages]
# Calculate RMSE for each stage
rmse_per_stage = {}
for stage in common_stages:
    actual_doy_stage = first_occurrence_actual[stage].dropna()
    predicted_doy_stage = first_occurrence_predicted[stage].dropna()

    # Align the indices
```

```
    actual_doy_stage, predicted_doy_stage = actual_doy_stage.align(predicted_doy_stage, join='inner')

    # Calculate RMSE for the current stage
    rmse_stage = mean_absolute_error(actual_doy_stage, predicted_doy_stage)
    rmse_per_stage[stage] = rmse_stage

# Print RMSE for each stage
for stage, rmse in rmse_per_stage.items():
    print(f'MAE for stage {stage}: {rmse}')
```