Smart Farming: Computer Simulation and Predictive Model for Cassava

by

Amadou Coulibaly

Thesis submitted to the School of Electrical Engineering and Computer Science in partial Fulfillment of the requirements for the degree of Master of Applied Science

School of Electrical Engineering and Computer Science Faculty of Engineering University of Ottawa

Abstract

The widespread adoption of technologies has made digital transformation relevant to almost every sector of the economy, including agriculture. Thanks to technologies such as the Internet of Things (IoT), the farming industry now has access to tools that enable a shift from precision agriculture to farming. Smart farming has progressed from precision agriculture, which relied on technologies like satellites and planes for precise product applications, such as pesticides. These improvements allowed for the collection of information and assisting farmers at a reduced cost. The research examined farming architecture systems and their different levels. It introduced commonly used machine learning models in agricultural data management. By applying mathematical methodologies, a simulation model was created to study the growth of crops, particularly focusing on cassava as the plant of interest. Additionally, various machine learning models were constructed and evaluated using the available data.

Acknowledgments

I express my gratitude to the almighty and extend heartfelt appreciation to my amazing parents for their unwavering support. I express gratitude to my patient supervisors for their guidance, tolerance, and assistance. It has been a blessing to have my partner and my siblings enduring the stress and the challenges alongside me. Last but not least, I would like to take this opportunity to thank myself for my hard work and for not quitting.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent decades, technology has made remarkable progress, leading to significant changes in how humans connect and how they interact with machines. These advancements have resulted in various notable developments. The development of telecommunications, particularly the advent of 5G, and the emergence of the Internet of Things (IoT) have expanded the scope of telecommunication services. These advancements have not only pushed the boundaries of telecommunications but have also extended their influence to various industries, including agriculture.

The agricultural and livestock industries have experienced notable shifts due to technological innovations. Progress has been made from an era when farmers had to rely on their experience and intuition to deal with the unpredictable forces of nature to a time when it is now possible to anticipate and model the potential impact of environmental changes with greater accuracy.

Undeniably, the integration of the Internet of Things has unlocked substantial agricultural potential. Within the scope of our research, we will specifically explore the technological advantages provided by the modeling and management of environmental variables.

1.1 Motivation

Smart farming, which refers to the adoption of advanced autonomous technologies, is experiencing rapid growth and has great potential to revolutionize agriculture. It can improve crop productivity and contribute to the long-term sustainability of farms. Through

the adoption of innovative tools such as sensors, drones, and data analytics, farmers can manage their crops more efficiently, conserve resources, and improve yields.

Given the shortage of the world's food resources by 2050 (according to FAO), it becomes crucial to investigate viable agricultural methods as a possible remedy. This presents smart farming as an intriguing area of research. By investigating the application of cutting-edge technologies and data analysis in agriculture, this study aims to contribute to the development and adoption of smart farming practices. The main objective is to use smart farming methods to improve the management of crops at the field level and use data analytics to optimize resource allocation.

Among the various crops, cassava has garnered significant attention due to its exceptional tolerance to acidic soils and adaptability to changing environmental conditions. These attributes make it well-suited for cultivation in developing countries and regions with challenging climates. The variety of data sources allows us to use computer simulations and predictive models to study how crops grow. We will Analyze data from different sources to create models that can predict crop yields and help us make the best use of resources.

Research in smart farming costs money, and very often developing countries find themselves left behind. The crop used in this research is cassava, a crop often used in developing countries and tolerant to environmental variations. One of our goals is to help these farmers neglected by science. Our research with UAVs and complementary sensors provides substantial assistance to cassava farmers, irrespective of their technological expertise. It equips them with the means to discern optimal planting periods, select suitable cassava varieties considering anticipated weather patterns and soil attributes (including nutrient levels), and address challenges such as nutrient deficiencies, weed proliferation, diseases, and water scarcity promptly as they arise. This research will help us gain an understanding of how advanced technologies and data analytics can be used in farming, particularly when it comes to cassava. This knowledge will play a role in promoting agricultural growth and tackling the issues surrounding food scarcity in the future.

1.2 Problem

Despite the benefits, the adoption of farming comes with a set of unresolved challenges that need to be tackled for its complete realization. The primary challenge is the cost involved in researching, developing, and sustaining the required technologies. Small-scale farmers, who typically operate with limited resources, face obstacles when it comes to investing in

expensive equipment, like drones, automated tractors, and other technological tools that are essential for smart farming practices.

Moreover, the absence of uniformity and interoperability in technologies poses an additional substantial obstacle. The current landscape of smart farming involves diverse sensor systems, platforms, and data formats, which poses obstacles in integrating and analyzing data across multiple systems. The absence of consistency in data makes it difficult to efficiently exchange and utilize information to make optimal decisions and manage resources.

Continuous research and investigation continue to delve into the potential advantages and ramifications of smart farming technologies. Important questions remain about how they impact productivity and resource usage and how these effects vary across different crop types, climates, and soil conditions. Our study will primarily focus on addressing these uncertainties. We aim to explore how smart farming, which seeks to maximize output-to-input ratios, can empower farmers by providing them with data-driven insights for informed decision-making.

Furthermore, our choice to focus on the cassava crop is motivated by several factors. Cassava is predominantly cultivated in developing countries and serves as a vital staple food. It possesses a remarkable tolerance to drought conditions and exhibits extended growth periods ranging from eight to eighteen months. These unique characteristics have led to cassava being labeled as a "crop for the poor" due to its potential to alleviate food security challenges in resource-constrained regions. However, existing crop modeling software for cassava is insufficiently updated and lacks the necessary accuracy. Another important goal of our research is to overcome this limitation and improve the precision and reliability of crop modeling for cassava.

1.3 Solution

Data acquisition becomes crucial to maximizing the output-to-input ratio in the context of smart farming. However, raw data alone holds little value unless properly interpreted and analyzed. While the collection of environmental variables through sensing mechanisms is an advantageous approach, the lack of knowledge regarding their use results in missed opportunities. In light of this, our research aims to address these challenges by analyzing various process-based models specific to cassava cultivation and developing a comprehensive model. This model will serve as a foundation for generating data that can be used for machine learning purposes.

The primary objective of our study is to conduct a comparative analysis of existing

models and evaluate the potential benefits of integrating machine learning techniques into smart farming practices, particularly in improving crop yield. To achieve this, we propose modifying the existing models and introducing a predictive model within the simulation framework. This enhanced simulation model will encompass crucial factors such as soil conditions, weather patterns, water stress, and nutrient stress, enabling a comprehensive understanding of the dynamic interactions between these variables and their influence on cassava growth and productivity. Our model incorporates nutrient uptake, representing an enhancement over the approach used in the [22] model.

By combining process-based modeling, data generation, and machine learning methodologies, our research endeavors to contribute to the advancement of smart farming practices, optimize crop management strategies, and enhance the overall productivity and sustainability of cassava cultivation.

1.4 Methodology

Machine learning (ML) models fall into both categories. The categorization depends on how the model is constructed and selected. Mathematical models establishes linear relationships, making linear ML models suitable for this approach. On the other hand, machine learning process-based models can be built using non-linear ML techniques such as random forests, K-NN, and neural networks. Furthermore, each process can be established as a service to represent the dynamic interactions among various components within crop.

The proposed approach entails conducting an in-depth examination of the crop yield model and developing a predictive model for cassava cultivation using non-linear machine learning techniques. Efficient farming techniques aim to maximize the use of resources. Consequently, our research will primarily concentrate on identifying the optimal strategies for input application in the context of cassava cultivation.

1.5 Contributions

The existing models for Cassava crop development have predominantly focused on analyzing the plant's response to ecological factors such as solar radiation and water dissipation. Our model aims to determine optimal timings for applying nutrients and water, as well as the cessation of such inputs. It is well-established that the maximum yield of the Cassava crop is highly responsive to inputs up to a certain stage of its development. Beyond this

stage, unless subjected to substantial environmental stress, the crop's yield remains unchanged. Furthermore, adding supplementary nutrients beyond this critical stage does not impact the crop's maximum yield. We can summarize our contribution at two significant levels:

- The development of a dynamic model that incorporates nutrient considerations.
- The creation of comparative machine learning models for yield.

1.6 Outline of the Thesis

This thesis consists of four additional chapters following this one.

Chapter 2 will focus on the background work, providing an in-depth exploration of smart farming and a comprehensive review of the technologies employed in this field. Additionally, it will include a thorough examination of crop simulation techniques specific to cassava and present various examples of machine learning techniques employed for crop yield simulation. Furthermore, the chapter will briefly discuss the architectures used in smart farming.

Chapter 3 will delve into the characteristics of cassava, presenting a detailed description of the plant itself. Moreover, it will highlight a process-based simulation model that will be implemented as a fundamental component of this research.

Chapter 4 will encompass the practical implementations of the models, including both process-based simulation and machine-learning approaches. The chapter will feature an analysis of the results obtained, along with a comprehensive discussion of these findings.

Finally, Chapter 5 will serve as the concluding chapter, summarizing the key outcomes of the thesis and presenting recommendations based on the findings derived from the study.

Chapter 2

Related Work

2.1 Smart Farming Technologies

Precision agriculture, also known as smart farming, is an innovative farming technique that applies advanced technology to maximize crop yield and improve farming efficiency in a sustainable manner. By leveraging data analysis, sensors, and digital tools, farmers can make informed decisions about crop management with precision and accuracy.

While smart farming and precision agriculture are often used interchangeably, there are differences between them. Precision agriculture focuses on leveraging technologies to optimize crop management and conserve resources. It involves collecting and analyzing data related to soil moisture, nutrient levels, and weather conditions using sensors, drones, GPS, and other digital tools. This valuable information can then be used to create farm maps and apply inputs, like water, fertilizers, and pesticides accurately to specific areas of the farm. The primary objective of precision agriculture is to maximize yields while minimizing resource waste.

On the other hand, smart farming is a concept that goes beyond precision agriculture. While precision agriculture is one of its key components, it involves technologies like the Internet of Things (IoT), artificial intelligence (AI), robotics, and big data to automate tasks, reduce human involvement, reduce cost, and improve production. The goal is to create farming techniques that are efficient in their use of resources and promote productivity, all while minimizing waste and environmental harm. This includes aspects such as monitoring, automated machinery, predictive analysis, and making decisions based on real-time data to optimize farming operations. Taking an approach like this can address the pollution and environmental damage that comes with farming methods.

Additionally, there exists the concept of intelligent farming. The core principle of intelligent farming solutions lies in extracting actionable insights and meaningful value from the data collected through precision farming practices. Intelligent farming goes beyond simply collecting data; it involves leveraging advanced analytics and artificial intelligence to transform raw data into valuable insights and actionable intelligence. This approach encompasses the concepts of precision farming and smart farming, aiming to optimize agricultural operations and drive sustainable outcomes.

A primary objective of smart farming is to increase crop yields while optimizing the use of resources such as water, fertilizers, and pesticides. Through the utilization of sensors and other data collection technologies, farmers can monitor various environmental variables in real-time, including temperature, humidity, soil moisture, and nutrient levels. This information enables improvements in irrigation, fertilization, and pest management, leading to better resource management and higher crop yields. Smart farming also utilizes precision agriculture tools such as drones, robots, and GPS-guided tractors to automate various farming tasks. Drones, for instance, can assess crop conditions, detect disease or stress symptoms, and provide specialized care. Robotic systems can be deployed to automate processes like seeding, planting, and harvesting, reducing the need for manual labor.

In conclusion, smart farming represents a cutting-edge approach to farming that has the potential to enhance the efficiency, productivity, and sustainability of agricultural operations. Through the utilization of cutting-edge technologies and data analysis, farmers can make informed choices when it comes to managing their crops. This ultimately results in yields. Promotes the adoption of sustainable agricultural methods. We will explore the technologies utilized in smart farming.

2.1.1 Sensors and IoT Devices

The Internet of Things (IoT) constitutes a network of interconnected physical terminals, commonly referred to as "objects." The IoT integrates sophisticated technologies such as sensors and software, which in turn facilitate the smooth transmission and reception of data. Essentially this network model allows for the transformation from the physical dimension into the digital dimension, making communication and information exchange between these realms more efficient through the use of the Internet. Over the years, IoT has found numerous applications spanning from intelligent domiciles and wearable technologies like smartwatches to the optimization of supply chain management.

Despite its multiple applications, the present focus lies on its pertinence to the agricultural domain. The IoT assumes a pivotal role in the collection of data through an

interlinked network. The adoption of IoT comes with an array of advantageous outcomes, including but not limited to:

- Reduction of workforce cost, connectivity established by objects enables the sharing of status updates and environmental data, thus achieving a diminution in human intervention,
- Increase in income by improving the quality and quantity of crops.,
- Facilitation of agricultural modernization.

Within these terminals, sensors equipped with microcontrollers serve as pivotal instruments in the facilitation of data exchange. While the terminal architecture accommodates other devices for data regularization, sensors invariably occupy a central role in data acquisition. Sensors are conceptually defined as devices designed to evaluate and then transmute physical data emanating from their surrounding environment into intelligible data that can be interpreted by end users. For example, thermometers deployed for human body temperature measurement are equipped with sensors that provide numerical values on interfaces as they interact with the human body. In the field of agriculture, there are sensors that are easily distinguishable:

- Electrochemical sensors, with their ability to collect information on soil pH and nutrient concentrations. Some sensors have the ability to perform a nuanced analysis of soil composition, resulting in the generation of soil chemical maps. Optical sensors are designed to determine soil attributes through the prism of light spectra, encompassing variables such as clay content, raw materials, and soil moisture.
- Mechanical sensors are dedicated to the quantification of the mechanical resistance of soils. These sensors allow us to record the strength exerted by the roots when absorbing water, which gives us information for managing soil irrigation methods.
- Air flow sensors are responsible for the dynamic measurement of soil air permeability. Their primary role is to determine the air pressure required to infuse a specified amount of air into the ground at a stipulated level.

Over time, scientific minds have worked diligently to instantiate various technologies, programs, and applications aimed at improving agricultural techniques. The following section will present selected examples of triumph in this area.

The first story focuses on the concept of precision farming. In the context of India, researchers have designed an IoT-based agricultural production system designed to monitor an array of parameters encompassing temperature, humidity, and water consumption, all aimed at optimizing productivity. The deployment of this system has resulted in a substantial 30% reduction in water consumption, representing a substantial conservation feat. At the same time, crop yield recorded a commendable increase ranging from 15 to 20 percent, including crops such as wheat, rice, and maize. In essence, this innovative system announced the confluence of time, financial, and energy savings concomitant with a significant improvement in agricultural production [20].

Additionally, there is evidence of cost savings in terms of reduced labor expenses. An example from China highlights the impact of an emerging IoT-driven system centered around greenhouses, which has resulted in a decrease in labor costs. Direct transmission of data to end users via a remotely manageable real-time application avoids the need for physical crop monitoring by farmers. In the previous era, without IoT integration, the greenhouse required a workforce of around sixty people; after the implementation, a simple group of six was enough. This recalibration resulted in a 60% reduction in labor costs. Furthermore, analysis of data led to a decrease in the use of fertilizers while there was an 80% rise in the deployment of pesticides [20].

Finally, here is an example of how the Internet of Things (IoT) has been used in the livestock industry. There is this farm in Africa that implemented a system to monitor the well-being of their animals in real time. They did this by outfitting the animals with collars that can send information directly to a platform. With this setup, farmers are now able to keep track of the animals grazing habits and migration patterns [20].

2.1.2 Blockchain

Improving agriculture requires a multi-faceted approach since the agricultural sector relies on cooperation between different technologies. Blockchain, a relatively new innovation that has shown practical potential, has motivated researchers to investigate its usefulness in various agricultural areas.

Agriculture and livestock involve a variety of activities from production to end user. This complex process includes stages of production, transportation, and preservation of the products. One of the challenges is ensuring the quality of products. Establishing connections between parties involved, such as farmers, suppliers, stores, and consumers. It's worth noting that there is a lack of transparency in terms of where the products come

from and financial aspects. Instances like food epidemics emphasize the need for solutions like blockchain. [18].

A block includes data, a reference to the previous block's hash (encryption), and its hash, forming a cryptographically linked chain. If data is altered, the hash changes, invalidating the chain. Concepts like mining, distributed peer-to-peer networks, consensus ledgers, and cryptographic hashes further define blockchain.

Blockchain functions as a secure record-keeping system where individuals store transactions. It acts as a reference point for the state of an ecosystem. In the agriculture sector, important information regarding farms, contracts, and management is stored in the blockchain. This enables decentralization, traceability, indisputability, automated payments, and the exchange of commodities [49]. Blockchain technology is guided by principles such as governance, accountability, transparency, flexibility, availability, usability, manageability, and sustainability.

Using RFID and blockchain, we are able to maintain standards of food safety and quality throughout the supply chain. This system covers fresh produce and meat, offering transparency to stakeholders. Introduced by [101], this approach focuses on data collection and information management for every transaction in the agriculture supply chain, enhancing monitoring and tracking. Some researchers propose similar solutions, integrating blockchain in agriculture supply chains with slight variations [28] [100] [107].

IBM Food Trust, built on Hyperledger Fabric, is another example of such applications. Developed in collaboration with Walmart, it traces product provenance, leveraging blockchain's transparency from producer to consumer. Data accessibility throughout the supply chain is a hallmark of this solution [6].

2.2 Smart Farming Architecture

2.2.1 Data

The integration of big data in smart farming revolutionizes agriculture. This approach involves collecting, processing, and analyzing vast amounts of agricultural data generated from various sources such as sensors, satellites, drones, weather stations, and machinery. Big data technologies enable farmers to make data-driven decisions, enhance crop management, and improve overall productivity in several ways.

Predictive Analytics: Big data facilitates predictive models that forecast weather patterns, pest infestations, crop diseases, and optimal planting times.

Precision Agriculture: By employing big data analytics, farmers can perform precise field-level analysis. This helps optimize irrigation, fertilizer application, and pesticide use, leading to resource efficiency and reduced environmental impact.

Real-time Monitoring: Integration of big data enables real-time monitoring of farm activities. IoT devices and sensors collect data on soil moisture, temperature, humidity, and crop health, providing farmers with instant insights for timely interventions.

Supply Chain Optimization: Big data facilitates better traceability and transparency in the agricultural supply chain. It allows stakeholders to track produce from farm to market, ensuring quality control and reducing waste.

Decision Support Systems: Advanced analytics tools assist in developing decision support systems that offer personalized recommendations to farmers, enabling them to optimize inputs and increase productivity.

2.2.2 Communication Systems

Advancements in computing power, including the Internet of Things (IoT), have led to new capabilities in communication systems, transitioning from human-to-machine interactions to machine-to-machine interactions. However, the latter posed numerous challenges. When our mobile devices or computers take two seconds to load a webpage, it might seem acceptable from a human standpoint. Yet, in terms of machine operations, this delay is deemed excessively slow 100 times too sluggish and thus unacceptable. This delay in communication is termed "latency." Addressing this issue, among others, prompted the creation of 5G.

5G networks facilitate a vast number of connected devices, fostering the proliferation of IoT devices and applications across various sectors, such as smart cities, autonomous vehicles, healthcare monitoring, and industrial automation. Within smart farming communication systems, 5G introduces several key technologies, including:

Narrowband IoT (NB-IoT): Specifically designed for low-power IoT devices that require long battery life and extended coverage, NB-IoT allows for efficient transmission of small data packets [111] [84].

Edge Computing: In the context of IoT, Edge Computing in 5G networks involves processing data closer to the source (devices or sensors) rather than transmitting it to distant data centers. This reduces latency and speeds up decision-making for time-sensitive applications like autonomous vehicles and industrial automation [10].

Network Slicing: 5G networks can be divided into multiple virtual networks through network slicing. Each slice is tailored to meet the specific requirements of different IoT applications. For instance, a slice could be optimized for smart farming, while another could be for connected vehicles.

Cloud computing: It is a foundational technology that serves as a centralized hub for data storage, processing, and analysis. Within the context of IoT in 5G, cloud computing enables the aggregation and management of immense volumes of data generated by various connected devices. This infrastructure provides scalable and efficient storage solutions and allows for sophisticated data analytics [10].

Fog computing and edge computing are both distributed computing paradigms that aim to process data closer to the source, reducing latency and enhancing efficiency in handling vast amounts of data. However, they differ in certain key aspects. Fog computing is positioned between the edge devices and the cloud, closer to the network's edge but not directly on the devices themselves. It operates at an intermediary level, typically within a local area network or across multiple networks. Edge computing, as the name suggests, occurs at the immediate "edge" or endpoint devices, closer to where data is generated and where actions are taken. It operates directly on devices like sensors, gateways, or routers. Fog Computing is suitable for applications requiring intermediate processing, analytics, and coordination across multiple edge devices or networks. Edge Computing is ideal for applications needing immediate, localized processing, such as real-time analytics, autonomous systems, or IoT devices in remote or resource-constrained environments [91].

2.2.3 Architecture Models

[3] introduces an IoT-enabled stick designed to monitor various agricultural parameters in real time. The stick aids farmers in obtaining immediate data on temperature and soil moisture. Through its plug-and-play functionality, this agricultural IoT device enables quick setup for smart monitoring by placing it in the field, allowing access to live data on smart devices such as tablets and phones. Furthermore, the sensor-generated information can be easily analyzed and accessed by agricultural experts, even in remote areas, utilizing cloud computing technologies [11].

[54] introduced a monitoring system for farms using IoT technology. Various sensors, including those for CO2 levels, temperature, humidity, soil moisture, light intensity, and pH values, gather environmental data and transmit it to a central gateway node. Complementing these sensors, cameras were deployed to capture photos and videos of the monitored

area. Acting as the gateway device, a microcontroller collects information from the sensors and sends it onward to a web portal for further analysis [99].

[97] introduced a smart farming system leveraging IoT technology for monitoring key parameters such as light intensity, humidity, soil moisture, and temperature while also implementing automated irrigation for plants. They utilized specific sensors for diverse environmental measurements: thermistors for soil temperature, coplanar capacitors for soil moisture, photo-resistors for light intensity, and sensors to measure both humidity and air temperature [99].

Based on research from [74] and [**?**], it's evident that architectures in various studies are predominantly built using a layered approach. Authors tend to differ in their depiction of these layers, ranging from 3 to 6 in number. However, a common trend emerges, where most of these architectural designs can be effectively summarized or condensed into four primary levels or layers.

Perception layer: This layer includes various sensors and devices deployed in the field to collect data on soil moisture, temperature, humidity, light intensity, pH levels, crop health, and more. It also contains actuators that enable actions in response to data inputs, such as controlling irrigation systems, adjusting temperature, and activating pest control measures.

Network layer: It involves the communication protocols and networks connecting sensors, actuators, and devices to transmit data. This layer often utilizes wireless technologies such as Wi-Fi, LoRaWAN, NB-IoT, or Zigbee to enable seamless data transmission. It also includes gateways and devices that facilitate the connection between local sensors and the wider internet, providing aggregation and initial processing of data before transmitting it to the cloud. It is sometimes referred to as the transport layer.

Processing Layer: Data from various sensors and edge devices are transmitted to the cloud, where it undergoes further processing, storage, analytics, and integration with other data sources. Cloud-based platforms provide a scalable and robust infrastructure for data management and advanced analytics. Edge computing happens at this layer too. By extension, this layer can also be used for data analytics, applying various analytics techniques, such as machine learning algorithms, predictive modeling, and AI, to derive insights from the collected data.

Application Layer: This layer consists of user interfaces, dashboards, and applications that farmers, agricultural experts, or stakeholders' access to monitor, analyze, and act upon the collected data. It includes tools for crop monitoring, predictive analytics, decision support systems, farm management applications, and feedback and control. It completes the loop by taking actions based on the analyzed data.

2.3 Machine Learning

In today's rapidly changing agricultural landscape, smart farming powered by artificial intelligence offers highly effective solutions to meet the pressing challenges of sustainable agriculture. Machine learning, deep learning, and time series analysis play a critical role in the foundations of smart farming. These advanced techniques enable a wide range of agricultural strategies, including crop selection, yield prediction, soil compatibility allocation, and water management. With the integration of artificial intelligence, modern agricultural practices have the potential to revolutionize productivity and sustainability [1].

To start, let's delve into some explanations. Artificial intelligence (AI) pertains to the intelligence displayed by machines. In the domain of computer science, AI research focuses on the analysis of "intelligent agents", devices that can understand their surroundings and make decisions to optimize their chances of accomplishing objectives. Meanwhile, machine learning represents a subfield within computer science that, as defined by Arthur Samuel in 1959, empowers computers with the ability to learn without being programmed [89]. Machine learning involves studying algorithms that can learn from data and use that knowledge to make predictions. These algorithms go beyond programming instructions by using data-driven methods to make predictions or decisions based on examples. Machine learning is used in computing tasks where it is difficult or impractical to design and program algorithms for high performance [77].

Supervised learning and unsupervised learning are two of the most widely adopted machine learning methods. In addition, there are types of machine learning, such as reinforcement learning and semi-supervised learning. Supervised and Unsupervised learning make up roughly 80 percent of the field of machine learning [77]. That's why these methods will be analyzed in more detail in the following sections.

2.3.1 Supervised Learning

In supervised learning, the algorithm undergoes training using a labeled dataset. Each data point in this dataset is associated with a known outcome or label. Various algorithms are used to create a function that connects inputs with the desired outputs. In supervised learning, we often encounter what is known as a classification problem. The main objective for the learner is to approximate the behavior of a function that connects a set of input features to classes. To accomplish this, the learner relies on a collection of training examples that show how the function behaves when given inputs (represented as vectors) and their corresponding class labels. By observing these examples, the model tries to understand

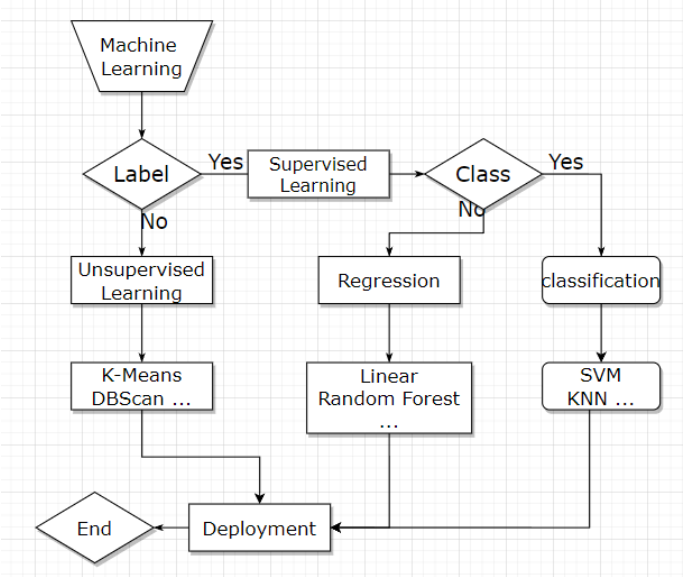


Figure 2.1: Machine learning techniques

the underlying patterns and connections between the input features and class labels. This knowledge enables it to accurately classify data points into their appropriate classes [73]. Let's say we're interested in creating a model that uses supervised learning to determine if an email is classified as "spam" or "not spam". We have a dataset of emails where each email is represented by features or attributes like word frequency, email length, and the sender's address. In addition to the email data, we also have labels for each email indicating whether it is categorized as "spam" or "not spam." These labels act as the reference points for our supervised learning algorithm. They Help it learn how to predict. Linear regression, logistic regression, support vector machine, decision trees, and Random Forests are some algorithms employed in supervised learning.

K-Nearest Neighbors (KNN)

K nearest neighbors (KNN) is a used and straightforward algorithm in supervised machine learning that can be applied to both classification and regression tasks [59]. It falls under the category of case-based learning methods [27], where it retains all the training data for classification purposes. KNN is considered a lazy learning algorithm because it doesn't construct a model during training; instead, it directly utilizes the stored training data for predictions.

During the model's construction process, each data point identifies its neighborhood, which consists of the maximum number of data points sharing the same class label. By analyzing these neighborhoods, we can determine the global neighborhood in each cycle. This largest global neighborhood acts as a representative that encompasses all the data points within its range [27]. In KNN, "K" refers to the number of neighbors considered when making predictions for data points. The fundamental concept behind KNN involves identifying the K data points from the training set in terms of distance or similarity (using metrics) to predict outcomes for new data points. In classification, the k neighbors vote to decide the class, while in regression, they contribute to the prediction. Although KNN is simple and easy to understand it may not be suitable for large datasets due to computational costs when storing and searching through the entire training set. It is considered a parametric and instance-based learning method as it relies solely on the training data without any assumptions about the underlying data distribution.

These are the steps for KNN conception:

1. Determine the value of K. Decide how many neighbors (K) should be taken into account when making predictions. Usually, an odd number is chosen to avoid ties.

2. Calculate distances. Use a distance metric to measure the distance between the data point and all other points in the training set.

3. Find K neighbors. Select K data points with the distances from the new data point.

4. Class vote. In classification tasks, these K nearest neighbors vote collectively to determine which class label should be assigned to the data point. The predicted class for

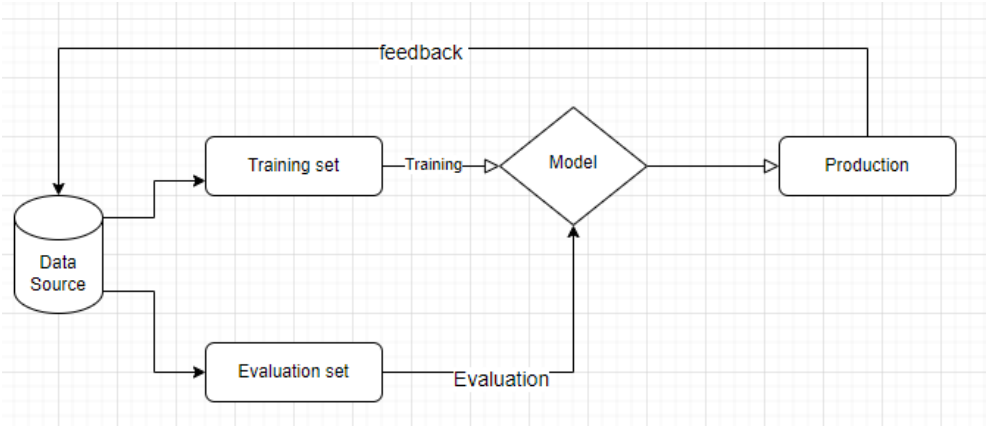


Figure 2.2: Supervised learning Workflow [59]

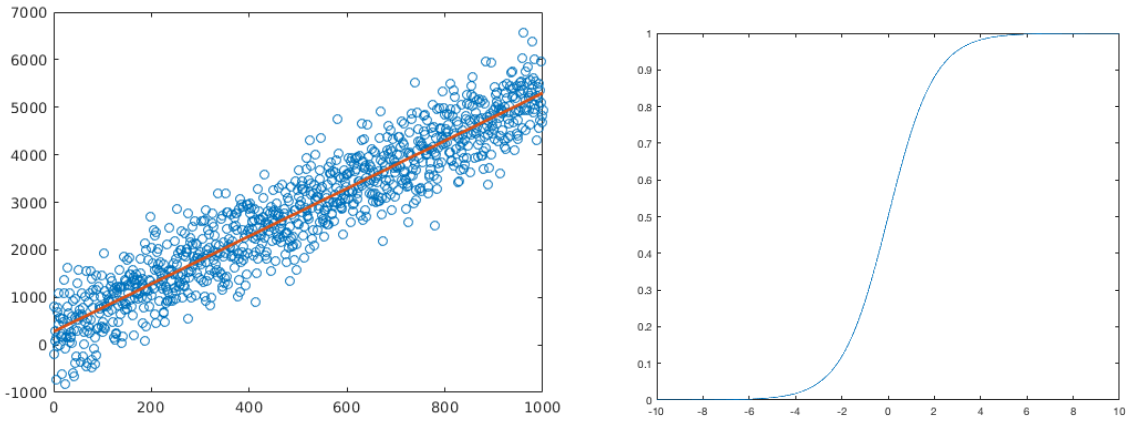
a data point is determined by selecting the class with the number of votes from the K nearest neighbors.

In regression tasks, a similar process is followed, but instead of voting, the values contributed by the K neighbors are considered. The predicted value for the data point is often calculated as the average of these contributed values. KNN is a used and straightforward algorithm in machine learning that can be applied to both classification and regression tasks.

Regression

In regression, the objective is to develop an understanding of the relationship between input characteristics and output values by utilizing the information provided in the training dataset. This enables the model to generate predictions for data points that have not been previously encountered.

Linear Regression. The primary goal of linear regression is to find the line (or hyperplane in higher dimensions) that shows the direct line relationship between the independent variables and the dependent variable. It establishes a modeling connection between a variable (also called the response variable) denoted as y and one or more independent variables (also known as predictor variables), which can be represented as X. We can think of these variables as a vector of D dimensions [73]. This relationship is expressed using an equation



(a) Linear regression

(b) Logistic Regression

Figure 2.3: Regression function's display

in this form:

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + ... + b_n * x_n \tag{2.1}$$

Where:
y is the value we want to predict (dependent variable)
X is the vector of features (independent variables)
b is the vector for bias or the function's slope

The method of regression is used to make predictions by taking into account a set of input features. It has applications in fields such as predicting stock prices forecasting weather, and projecting sales.

Logistic Regression. Logistic regression is commonly utilized for tasks involving classification. Its purpose is to predict one of two outcomes, such as true or false, yes or no. Despite its name, logistic regression is a classification algorithm than a regression algorithm.

The underlying concept of regression revolves around predicting the probability of an event occurring by data to a logistic function. Logistic regression takes into account predictor variables that can be either numerical or categorical [73]. The logistic function, which is sometimes called the sigmoid function, is used to represent this probability. It guarantees that the estimated probabilities will always be between 0 and 1.

The logistic regression model takes a combination of variables and their corresponding coefficients. Uses the sigmoid function to convert them into probabilities. We can represent this regression model as shown below:

$$P(y = 1|X) = \frac{1}{1 + e^{-z}} \tag{2.2}$$

Where:
$P(y = 1|X)$ is the probability that the dependent variable y is equal to 1 given the independent variables X
X is the vector of features (independent variables)
z is the linear combination of the independent variables and their coefficients

Support Vector Machines (SVM)

Support Vector Machines (SVM) are a type of model used in data analysis to classify information or predict output values. They do this by identifying the hyperplane that

separates classes. SVMs are effective for both classification and regression tasks. They excel at solving classification problems where data points need to be binary-categorized.

The main goal of SVM is to find the hyperplane that separates data points of classes in the feature space. In a two-feature space, a hyperplane is just a line that divides the data into two classes. SVM strives to maximize the margin, which is the distance between the support vectors (representative points) of each class and the decision boundary.

Moreover, SVMs possess the ability to efficiently handle non-linear classification tasks through a technique referred to as the kernel trick. This trick implicitly maps inputs into feature spaces, allowing for more complex classifications by drawing margins between classes [59].

Overall, Support Vector Machines offer solutions for classifying data and making predictions while leveraging concepts like hyperplanes to achieve accurate results. On the other hand, SVM might necessitate adjustment of hyperparameters, and its training duration can be computationally demanding when dealing with extensive datasets. The margins are designed in a way that maximizes the distance between them and the classes, thus reducing classification errors. They are applied in fields such as image classification, text classification, and bioinformatics.

Decision Trees

Decision trees apply a tree structure to model decisions and their potential consequences. They are commonly applied in customer segmentation, product recommendations, and medical diagnosis.

It is a type of model that divides data recursively into subsets based on input features to predict the target variable. These models, known as decision trees, are constructed using a sequence of tests. Each test compares an attribute to a value (threshold) or set of values. Decision trees have an advantage because they offer greater clarity and understandability in their decision-making process [43].

The decision tree begins with the dataset as the starting point. Makes splits at internal nodes based on features that effectively separate the data into different classes or groups. The choice of feature and threshold for each split is determined by maximizing information gain or reducing impurity at that node.

In a decision tree, when a data point falls within a region, it is classified as belonging to the most recurrent class in that region. The error rate is calculated by comparing misclassified points to the number of data points while the accuracy rate is derived by

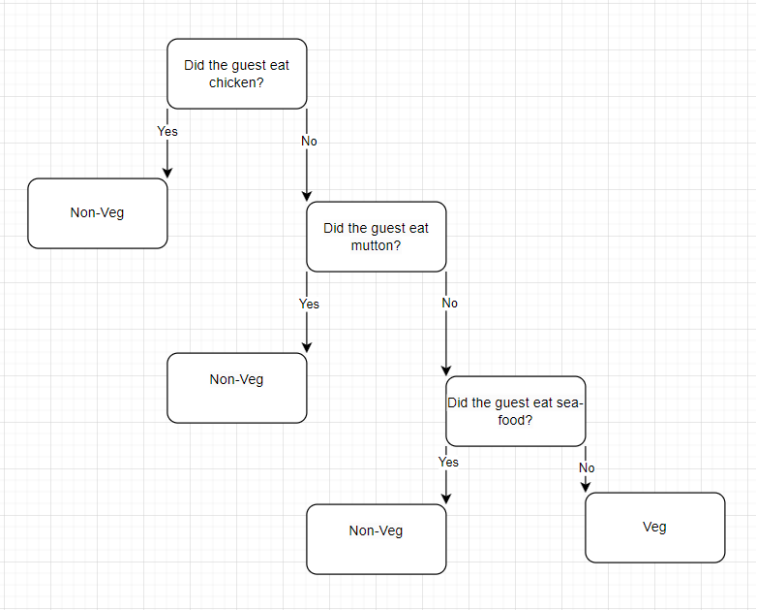


Figure 2.4: Decision Tree [59]

subtracting the error rate from one. Once the Decision Tree has been constructed, it can be employed to make predictions on data by traversing the trees' branches until it reaches a leaf node. The leaf node represents the anticipated class (for classification) or value (for regression).

Random Forests

Random Forest is a widely used learning technique in machine learning. It combines decision trees to create a robust and accurate model. This approach is commonly employed for both classification and regression tasks.

The main concept behind Random Forest involves creating decision trees during the training phase and then combining their predictions to make the final prediction. Each decision tree in the Random Forest is trained on a subset of the training data and a random subset of the features. This randomness introduces diversity among the trees, which helps prevent overfitting and enhances the performance of the ensemble model.

To elaborate further, based on [54] [78], during training, a random subset of the training data is selected with a replacement called bootstrapping or bagging. Some data points may be repeated, while others may be left out. For each tree in the Random Forest, a random

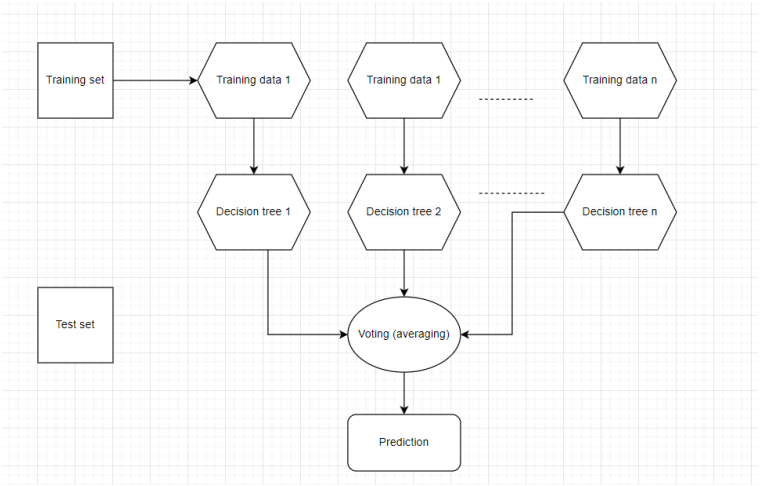


Figure 2.5: Random Forest Algorithm [83]

subset of features (variables) is chosen at each node to make decisions within that tree, thereby introducing diversity among them. Multiple decision trees are constructed using these bootstrapped samples.

Trees, in a Random Forest model, are cultivated until they meet certain conditions, such as reaching a depth or having a minimum number of data points at each node. In classification tasks, each tree provides its input on the class of a data point. The final prediction is determined by selecting the majority class. In regression tasks, the predictions made from each tree are combined to calculate the predicted value.

Random Forests can be constructed using either input variables or output [54]. Their usage can be found in tasks such as image classification, text classification, and fraud detection.

2.3.2 Unsupervised Learning

According to [77] [59] [1], in unsupervised learning, an algorithm is trained on a dataset where the outcomes are unknown and no labels. Unlike supervised learning, which provides answers, unsupervised learning operates without guidance. The goal of the algorithm is to identify patterns or structures in the data, such as clusters or associations, without any previous knowledge or clear indication of these patterns.

The algorithm explores the data to discover structures or groupings and aims to understand how the data is distributed and correlated. By finding similarities or differences

among data points, the algorithm strives to uncover underlying patterns.

Unsupervised learning includes two types of tasks: clustering and dimensionality reduction.

Clustering algorithms group similar data points together based on their features with the objective of creating partitions where data points within the same cluster are more similar to each other than those in other clusters. Clustering finds applications in areas like customer segmentation, image segmentation, and anomaly detection.

On the other hand, dimensionality reduction algorithms aim to reduce the number of features (dimensions) in the data while preserving information. This is particularly useful for datasets with high dimensions as it simplifies data representation and addresses challenges caused by having many dimensions. Used methods for reducing dimensions include techniques such as Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding.

For instance, marketers can use unsupervised learning to identify groups of customers who share attributes, allowing them to target these segments effectively in their campaigns.

K-Means

The study of [80] [96] [52] reveals the following. K-means is a type of machine learning algorithm that falls under the category of unsupervised learning. Its purpose is to group data points into clusters, making it easier to analyze and process them.

The K-Means algorithm operates through the following steps:

1. Initially, K data points are randomly chosen from the dataset to serve as the centroids for each cluster. These centroids act as the center points of their clusters.

2. Each data point in the dataset is assigned to the cluster whose centroid is closest to it using a distance metric.

3. The centroids of each cluster are recalculated by taking the mean of all data points assigned to that cluster. This updated centroid becomes the center point for its cluster.

4. Steps 2 and 3 are repeated until there is no significant change in the centroids or a maximum number of iterations has been reached.

The outcome of applying the K-Means algorithm is a set of K clusters, where each cluster is represented by its centroid. Data points within each cluster share characteristics are grouped based on their proximity to their corresponding centroid. Determining the

number of clusters K is an aspect of the K-Means algorithm. It can be quite challenging to find the value. Often requires domain knowledge or techniques, like the elbow method or silhouette score.

It's important to note that depending on the centroids the K-Means algorithm may converge to different solutions. To address this issue, it is practice to run the algorithm times with various initializations and select the best outcome based on cluster quality.

K-Means is widely used for clustering tasks due to its simplicity and effectiveness. However, it does have some limitations, such as being sensitive to centroids struggling with clusters of different shapes and sizes and requiring prior knowledge of the number of clusters. For more complex clustering problems, other algorithms like DBSCAN may be more appropriate.

Density-based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a known learning technique used for clustering data. It stands out as a pioneer for density-based clustering algorithms that are specially designed to cluster data points of various shapes while also accounting for noise, in both non-spatial and spatial datasets with high dimensions [42]. Unlike partition-based clustering algorithms like K-Means, which assume clusters and require the number of clusters as input, DBSCAN has the ability to discover clusters of shapes without needing the number of clusters to be specified beforehand.

The core concept behind DBSCAN revolves around the idea that for each object within a cluster, its neighborhood defined by a radius (epsilon) should encompass at least a minimum number of other objects (MinPts). Therefore, the cardinality of the neighborhood needs to exceed a predefined threshold. In other terms, DBSCAN works based on density connectivity. The algorithm groups together data points that are close to each other and have neighboring points within a given distance (epsilon) [93]. Based on [31], it classifies data points into three categories; core points, border points, and noise points.

Core Points: A data point is considered a core point if it has the required number of other data points within its epsilon neighborhood. To put it simply a core point is a point that has neighboring points to form a cluster.

Border points are the ones that fall within the epsilon neighborhood of a core point but don't have enough neighbor points to be considered core points. They are still part of a cluster.

On the other hand, noise points (or outliers) are the ones that don't belong to any cluster because they aren't core or border points.

The DBSCAN algorithm starts with a data point and gradually expands the cluster by adding core and border points connected to each other based on their density connectivity. Once no points can be added, the algorithm moves on to another point to create the next cluster. This process continues until all data points are assigned to clusters or identified as noise.

DBSCAN has advantages such as its ability to handle data with cluster densities, its resilience against outliers, and its capability to discover clusters in various shapes. However, it may encounter challenges when dealing with data or data with significantly varying densities. DBSCAN is a method utilized in fields including spatial data analysis, image segmentation, and anomaly detection. It is known for its ability to accurately detect clusters and outliers within the data without needing the number of clusters to be specified beforehand.

Principle Component Analysis (PCA)

Principal Component Analysis (PCA) is a utilized technique that aims to reduce the dimensions of a dataset while preserving information. Its purpose is to identify patterns within the data and represent them effectively [48]. Based on [58] and [37], Here is a logical step-by-step process for building PCA:

1. Standardize the Data. If the features in the dataset have varying scales or units it is important to standardize them so that they have an average of zero and a standard deviation of one. This ensures that all features contribute equally during PCA analysis.

2. Calculate the Covariance Matrix. Determine the covariance matrix of the data. This matrix represents the relationships between features. Plays an essential role in PCA.

3. Compute Eigenvectors and Eigenvalues. Find the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors indicate the principal components within the data, while eigenvalues represent the variance of each principal component.

4. Arrange Eigenvectors. The eigenvectors should be in descending order based on their corresponding eigenvalues. This step is crucial as it determines which principal components are most important.

5. Select the Number of Principal Components (K). Decide on the number of components (K) to retain for representing the lower dimension in data. One common approach is to select the value of K based on the variance, aiming to retain a percentage of the total variance.

6. Data Projection. We can take the K eigenvectors. Create a projection matrix using them. By projecting the data onto this lower dimensional space using the projection matrix, we obtain a reduced representation of the data.

To illustrate this better, let us consider an example [7]. Imagine we have a dataset of houses with features like room count, square footage, bathroom count, house age, and price. Each house is represented as a data point in feature space with five dimensions (five features). Now, suppose we want to visualize this dataset on two plots to understand how the features of the house are related. However, directly plotting the data in two dimensions may not provide information since we have five dimensions. PCA can help us reduce the dataset's dimensionality from five to two. This allows us to project the data onto a plane in two dimensions while preserving information. The algorithm achieves this by identifying directions (principal components) that exhibit data variation. The first principal component represents the direction that shows the highest variance in the data. The second principal component, which is perpendicular to the first one, represents the direction that shows the second highest variance. Together these main factors create a coordinated system that captures the information in the data. By projecting the data points onto these principal components, we effectively reduce the complexity of the dataset from 5 dimensions to 2 dimensions. This resulting 2D representation allows us to visualize and understand the data easily, giving us insights into how houses are related based on their most important features.

Principal Component Analysis (PCA) is commonly used for tasks such as preparing data for analysis, creating visualizations, and extracting features. It helps to simplify datasets and facilitates analysis and machine learning tasks.

2.3.3 Deep Learning

Deep learning is a branch of machine learning that focuses on training networks to carry out complex tasks. It draws inspiration from the structure and functioning of the brain, where interconnected neurons collaborate to process information and make decisions. Deep learning models, in this case, deep neural networks, are composed of multiple layers of interconnected artificial neurons, also referred to as nodes or units.

Let us have an overview of Neural Networks (NN). They comprise interconnected neurons organized in layers. Each neuron takes inputs, performs computations on them, and generates an output. Neural networks can be used for tasks such as identifying patterns, classifying data, making predictions, and making decisions. The structure of NN is as follows:

Input Layer. This is the layer where data is fed into the network. Each neuron in this layer corresponds to a feature or input variable. **Hidden Layers**. These intermediate layers exist between the input and output layers. Their purpose is to process and transform information received from the input layer using their weights and biases. **Output Layer**. This final layer produces the desired output based on the required task, whether it is classification or regression. The number of neurons, in this layer depends on that specific objective.

The learning process consists of two steps: forward propagation and backpropagation. During forward propagation, the neural network receives input data, which flows through its layers from the input to the output layer. At each neuron the weighted sum of inputs is computed, followed by applying an activation function to generate the output of the neuron. After propagation, the neural network compares its predictions with the target values and calculates the prediction error or loss. This error is then propagated back through the network during backpropagation to update weights and biases, minimize loss and enhance performance.

Neural networks can be utilized for supervised and unsupervised learning. In supervised learning, a labeled dataset is used to train the network where both input data and corresponding target outputs (labels) are provided. Throughout training, the network adjusts its weights and biases to accurately map input data to correct output labels.

An example of unsupervised learning in NN is autoencoders. Autoencoders are designed to learn a condensed representation of input data by encoding it into a lower dimensional space and then decoding it back into its original form. The network is trained to minimize reconstruction error compelling it to capture features of the data. Clustering is another method of unsupervised learning. It involves using networks to group data points together based on their features or similarities. One used clustering algorithm in NN is the Self Organizing Map (SOM). This algorithm organizes data points in a two-dimensional map, taking into account their similarity in the original dimensional space.

The term "deep", in deep learning refers to the network depth, which means it has multiple hidden layers between the input and output layers. These hidden layers allow the network to understand levels of abstraction in the data. The learning process involves adjusting the weights and biases of the neurons to minimize the difference between predicted outputs and actual target values during training.

Deep learning is particularly good at learning features, where the model learns to extract patterns or characteristics directly from raw data without needing manual feature engineering. This ability to learn representations helps the model uncover patterns and relationships within the data.

During training, errors are propagated backward through the network, allowing adjustments to be made to minimize prediction errors by updating parameters. Effective training of deep learning algorithms requires labeled datasets. Moreover, training deep neural networks requires power. This is possible thanks to advancements in hardware like graphical processing units (GPUs) and specialized hardware such as tensor processing units (TPUs).

2.3.4 Literature Review: Machine Learning in Smart Farming

In the field of smart farming, artificial intelligence has been used to improve the overall goal of increasing crop production.

Pest, disease, and weed management: Systems powered by AI have the capacity to analyze plant images to swiftly identify initial indications of pests and diseases. This timely identification allows for prompt interventions to circumvent the spread of issues and diminish crop losses. AI-driven robotic systems, equipped with cameras and machine learning algorithms, are proficient in discerning and selectively eradicating weeds, thus reducing the necessity for herbicides. [24] adopted an object-oriented approach to establish a rule-based framework while creating TEAPEST, an expert system tailored to elevate pest management. [33] introduced an integrated model that combines image processing and artificial neural networks for proficient disease classification in phalaenopsis seedlings. [9] employed machine vision coupled with a neural network to distinguish between weeds of distinct species.

Weather impact: AI algorithms meticulously analyze weather patterns and historical climate data to predict the optimal planting time for crops. Additionally, these algorithms assist farmers in adapting to fluctuating climate conditions, making informed decisions, and mitigating drought impacts. This includes recommending ideal crop rotation patterns based on soil health, pest history, and nutrient depletion to sustain soil fertility and reduce disease vulnerabilities. There are applications that involve data analysis to choose the most suitable crops for various seasons while considering soil attributes to align crop varieties. [47] proposed a method for crop selection aimed at optimizing choices and amplifying yield rates. [13] introduced a methodology involving Support Vector Machines (SVM) for feature selection followed by a decision tree for classification. [56] showed the importance of crop classification through SVM and highlighted the role of feature selection in enhancing model accuracy.

Yield, irrigation, nutrient, and soil management: AI-fueled sensors and data analysis determine the timing and quantity of irrigation based on real-time soil moisture levels, weather forecasts, and crop water requirements, thereby curbing over-irrigation and

conserving water resources. AI models process data concerning soil nutrient levels and crop necessities, offering precise recommendations for fertilizer application rates, thus decreasing excessive fertilizer usage and environmental impacts. Leveraging historical and real-time data, AI models proficiently predict crop yields, facilitating more strategic planning of harvesting, storage, and marketing efforts. [62] performed a comparative analysis of diverse neural network architectures for rainwater prediction, employing distinct input variables. [41] provided an exemplary instance of data utilization for weather forecasting and yield prediction using deep neural networks.

In the realm of bioinformatics analyses, there are machine learning tools and algorithms across various domains, including identifying protein-coding genes, identifying cis-regulatory elements, analyzing gene expression, studying protein interactions, determining subcellular localization, exploring gene ontology, examining metabolic pathways, predicting phenotypic traits, and making genomic predictions—all extensively reviewed by [61].

2.4 Crop Simulation

A model is a representation or simulation of a system typically created using equations or replication. Its purpose is to help people understand and enhance the performance of the system. In crop science crop models play a role in expanding our knowledge of how crops interact with agricultural practices under different conditions and timeframes. These models are highly regarded as tools that assist agronomists, farmers, policymakers, and researchers in making decisions and providing recommendations based on solid information [85].

Creating a crop model requires achieving equilibrium between theoretical concepts and practical application, intricacy and reliability, and the extent and depth of coverage. The main goal is to develop a model that captures both the processes that govern plant growth and the dynamic interactions among these processes within a farming system, which includes plants, soil, climate, and management practices. Ultimately, the objective of a crop model is to simulate and comprehend plant growth as a consequence of intricate interactions within the crop system [79].

Various models have emerged over time, categorized into groups ranging from empirical to explanatory models. Empirical models primarily rely on direct descriptions of data, usually presented as regression equations that involve one or a few factors. Their main goal is to predict the final yield. This method involves examining the data selecting equations or sets of equations, and fitting them to the data. However, these models do not provide

insights into the underlying mechanisms behind the observed response. They are also commonly referred to as statistical models. On the other hand, mechanistic models, which are also called dynamic models, not only explain how weather parameters affect crop yield but also delve into the underlying mechanisms that drive these relationships [79].

2.4.1 Process-based Model

Dynamic models can vary in complexity from representations that track the growth and development of a cultivar under specific conditions to more intricate models that consider factors like weather, soil, crop management, and different cultivars. These models use equations to capture changes over time, ranging from weekly evaluations to the most recent models that provide hourly assessments.

Cassava doesn't have defined growth stages, and the time it takes to harvest can vary significantly depending on how farmers handle it in the field. Some models rely on determined growth phases, but experts don't unanimously agree on these classifications.

The effective leaf area in models is influenced by both the growth of leaves and the shedding of old ones during senescence. Leaf area can increase through the development of node cohorts or through patterns of leaf growth and biomass distribution among different parts of the plant.

Different techniques are used in models to evaluate biomass assimilation. One approach involves establishing a connection between the growth rate of crops and the leaf area index (LAI). In cassava, there are two methods used to simulate biomass distribution. The first method uses derived factors (empirical method) to determine how biomass accumulates in different parts of the plant. The second method is known as the spillover model, which prioritizes directing nutrients to the upper part of the plant. When there is an excess of assimilates beyond what's needed for the upper part, it extends into the tubers [70]. Here are a few models.

cock model: Introduced by Cock et al. (1979) at CIAT [14], this was the inaugural dynamic model dedicated to cassava. This model doesn't account for soil and weather variations nor offers multiple management options, focusing solely on plant spacing. The model parameters were determined based on data collected from field experiments conducted near the equator. Its main function is to simulate the growth of leaf area index (LAI) by taking into account factors such as the rate of leaf formation per apex, the number of apices per plant, the density of shoots, per land unit, individual leaf area, and leaf longevity [70].

Fukai and Hammer: The Fukai-Hammer model [22] marked a significant progression in cassava modeling, introducing variations in weather, soil fertility, and soil conditions. One notable contribution was the introduction of thermal time, which involved setting a base temperature and later became a practice in subsequent dynamic models. This model was pivotal in predicting yields across a wide spectrum of environmental conditions in tropical and subtropical regions, representing the first instance where weather and soil variability were taken into account. Using empirical relationships, it accurately depicted leaf area growth along with stem development. Biomass production estimates were derived by factoring in crop growth rate to LAI (Leaf Area Index), which is further adjusted by stress factors. The distribution of biomass was determined through derived partitioning factors [70].

Gumcas: The GUMCAS model [64], which is one of the leading cassava models, introduced the consideration of Vapor Pressure Deficit (VPD). It takes into account the daily average VPD as a factor that influences the growth of crops. Similar to the spillover model used in the Cock model, it determines how biomass is distributed. Stem growth is a defined fraction derived from leaf growth, which is itself contingent on crop age determined by thermal time. This comprehensive model encompasses 23 crop parameters involving aspects such as photoperiod sensitivity, time taken for development from emergence to branching, rate of leaf appearance, branch count, VPD sensitivity, crop growth rate, leaf characteristics, duration of leaves, and allocation of biomass to stems and fibrous root growth [70].

SIMANIHOT: SIMANIHOT [102] emerged as an extension of the original GUMCAS model, serving the purpose of comparing and integrating certain functions into the existing DSSAT model available at the time. The integration process involved evaluating both models in the conditions found in Southern Brazil, taking into account the unique characteristics of local cassava varieties. This phenological model, which was created in the region of Brazil, stands out as a pioneer among cassava models by considering the impact of winter seasons. Furthermore, SIMANIHOT is designed to incorporate responses to increased CO2 levels, allowing for assessments of how climate change affects cassava production in Rio Grande do Sul state in Brazil [70].

SIMCAS: The SIMCAS model [69] emphasizes the balance between cassava's ability to absorb nutrients and its ability to produce them, acknowledging the adjustments induced by field conditions. Additionally, SIMCAS acknowledges the constraints of empirical models that necessitate recalibration of parameters when assessing new varieties or environments. This is done by building upon knowledge acquired from previous models, particularly the GUMCAS model [70].

In addition to the previous models we have seen, there are more models such as HyCAS [65], SUCROS [25], DSSAT [36], LINTUL [19], DYNCAS [16], and FAO agroecological zone [75]. We focused on the models that had the most impact and presented them in chronological order [70].

Certain models impose time limitations on simulations: Fukai and Hammer and DYNCAS limit simulations to around 52 weeks. In contrast, the FAO Agroecological Zone method gives the option to simulate a growing season lasting between 12 to 18 months. Some models assume cassava as a determinate crop. For example, GUMCAS defines a phase where no new leaves emerge and has a maturity stage that stops growth. It is unclear from the literature whether the maturity stage in the SIMANIHOT model was adapted from the GUMCAS model. Complex dynamic models, although comprehensive, harbor a drawback due to their numerous variables, parameters, and processes. This complexity may lead to small inaccuracies accumulating across multiple processes, potentially resulting in significant errors, particularly in yield estimations [70].

2.4.2 Statistical-based Model

Static models estimate the yield of roots by considering conditions during the growing season at a fixed harvest time. However, these models fail to take into account the interactions between components of the plant during its growth cycle. Here are a few milestone models in this category.

Boerboom: Boerboom's cassava model [8], established in 1978, stands as one of the earliest published models, coinciding with the development timeline of the Cock model. According to this model, tuber growth is preceded by a phase where no growth occurs, followed by a prolonged period characterized by the development of roots. After initiation, the growth of roots consistently accounts for a portion of biomass, which is known as the efficiency of root production (ESRP).

Manrique: In 1992, Manrique proposed a model [63] that used linear regression to establish a relationship between biomass production and allocation with temperature and solar radiation. This model was based on experiments conducted across three distinct elevations in Hawaii. It functions as a static model, generating simulated LAI and biomass using distinct equation sets for 60-day intervals. Despite incorporating specific coefficients for temperature and solar radiation responses in the equations, it does not account for the interactions between these interconnected input factors.

QUEFTS: Byju adapted in 2012 for cassava the static model initially designed for maize, known as Quantitative Evaluation of the Fertility of Tropical Soils (QUEFTS) by

Janssen et al. in 1990 [35]. This adaptation was evaluated in regions of India. Showed significant improvements in fertilizer recommendations for nitrogen, phosphorus, and potassium (NPK) usage in cassava farming. QUEFTS works by establishing a relationship between uptake and crop yield. It predicts crop yields by considering the contributions from soil and fertilizer supplies of N, P, and K, the internal nutrient use efficiencies as achievable yields based on geographical location and climate conditions.

When it comes to situations where accurate predictions are crucial, for activities that rely on planning yields at the end of the season, you may prefer to choose static or simpler dynamic models. However, in scenarios demanding models to provide insights into uncharted environments, novel management approaches, or the performance of new phenotypes, more complex dynamic models are better suited. Additionally, these advanced models give us an opportunity to perform experiments for circumstances like predicting how crops will behave in future climates with higher levels of carbon dioxide.

Chapter 3

Cassava and Simulation Models

Cassava plays a role as a food crop for millions of small-scale farmers in tropical developing nations. It is cultivated primarily in regions by low-income farmers who rely on minimal resources. Cassava's resilience allows it to yield harvests in challenging conditions where other crops struggle. However, due to the growing demand for cassava products, there has been an increase in production intensity. This includes the shift towards monoculture farming cultivation of yielding varieties and the use of agrochemicals and irrigation. Unfortunately, this intensification poses risks such as outbreaks of pests and diseases, depletion of soil nutrients, and environmental degradation. And using models can help reduce the environmental impact.

Scientists have categorized cassava models into two groups based on how assimilated nutrients are distributed among the plant's parts, affecting their growth. The first group consists of models that follow a fixed pattern of distribution. For example, a model developed by [8] established a linear relationship between the weight of the storage roots and the total weight of the plant. However, this approach overlooks dynamic physiological processes in the plant that contribute to its final yield. As a result, it has a limited ability to respond to environmental conditions and doesn't accurately account for factors like the lifespan of leaves affected by the environment, which only impacts leaf growth and not the other parts. This limitation leads us to consider the second category.

In this category, which is the focus of our study, there are models such as [14] that have been developed. In these models, the Crop Growth Rate (CGR) is assumed to be a function of the Leaf Area Index (LAI). However, one notable drawback of this model is its inability to study crop performance under varying radiation and temperature conditions. Therefore, it fails to consider variations in environmental factors.

This initial model didn't consider the effects of temperature, solar radiation, and water stress. Although branching was taken into account, its regulation wasn't fully integrated. In a related study in [22], the concept of crop growth rate (CGR) was expanded to include the effects of light as observed in [14] along with temperature and water stress. The allocation of dry matter was determined based on relationships using data from [38], a study on the MAusl0 (cassava variety) cultivar considering factors like temperature, photoperiod, and influences from leaf and shoot size. Leaf senescence was calculated by multiplying the senescence rate with modifiers related to temperature, water status, or shading. Adapting this model to other environments would require calibration efforts. The model is from Australia.

Earlier models had limitations that were effectively addressed by the cassava simulation model GUMCAS [64]. This particular model takes into account the potential CGR, which is a varietal trait used to compute dry matter production. To calculate the impact of stress caused by factors, like radiation, temperature, and water deficit on CGR, multipliers are used in a manner as in [22]. SIMCAS was developed to incorporate considerations for supply. While we may not have control over factors (except for irrigation), this model aims to minimize the wastage of moisture, nitrogen, and potassium while optimizing yield through the timely application of required amounts. Additionally, it also evaluates how shortages of moisture, nitrogen, and potassium can affect crop growth and yield.

3.1 Cassava Crop

Based on [32], cassava, scientifically known as Manihot esculenta Crantz, is a plant that has been cultivated for over 9,000 years primarily for its roots. It is believed to have originated in the Brazilian Amazon region. It is now grown by small-scale farmers in subtropical areas. One of its qualities is its ability to thrive in challenging environments. It can be propagated from stem cuttings, withstands acidic soils, and efficiently utilizes water and nutrients. The cassava plant is made up of three parts: leaves, stem, and storage roots. The leaves of the cassava plant contain glycosides as a defense mechanism against herbivores. The roots, which consist of 60% water content, are packed with carbohydrates. It Serves as an essential source of energy. They are typically harvested at around 8 to 10 months for maximum starch yield. Additionally, cassava exhibits versatility beyond its roots; its leaves are used for animal feed, while the root starch finds applications in industries.

Despite its earlier reputation as difficult to intensify due to its bulkiness and susceptibility to pests and diseases, in the past, cassava has experienced substantial growth. From 1980 to 2011, there was a 44% expansion in the area dedicated to cassava cultivation,

resulting in doubling production. Over the years, this growth trend has accelerated due to increased demand. Intensified farming methods aimed at enhancing productivity. Although current yields still fall short of their potential, with management practices and the development of drought-resistant varieties, it's possible for cassava crops to produce over 23.2 tonnes per hectare.

3.1.1 Farming System

The "Save and Grow" approach to farming systems focuses on three recommendations for cassava cultivation. Firstly, farmers are encouraged to take care of the soil by minimizing disturbance. Instead of plowing, they should consider using conservation techniques like strip or minimum tillage and zero tillage. These practices help maintain the structure of the soil, preserve matter, and prevent erosion.

Secondly, it is suggested that farmers keep a layer of material on the soil surface by using crop residues and mulches. This improves the properties of the soil. Also promotes beneficial microorganisms while conserving water and nutrients. In zero tillage systems, crops are planted directly into a layer of crop residues.

Thirdly, diversifying plant species through mixed cropping and rotations is recommended. This approach reduces risks, allows for adaptation to market changes, and helps mitigate climate impacts. By combining nutrient-demanding crops with legumes that enrich the soil, farmers can enhance fertility while preventing pests and diseases from spreading. Implementing these practices leads to increases in yield production as well as environmental benefits. The decision between tillage and no tillage depends on the conditions of the soil; while degraded soils may require some level of plowing, healthy soils generally benefit from zero tillage.

Cover crops and mulching play a role in protecting the soil during cassava's initial growth phase while also suppressing weed growth. Mulching also helps to stabilize temperatures, retain moisture, and foster the growth of microorganisms. Growing cassava alongside crops like maize, legumes, or trees improves the use of land, prevents erosion, and provides a wider range of food options.

In areas with limited resources and where cereals are grown, it is recommended to practice agroforestry, crop rotation, and intercropping. By combining these methods, we can promote intensification while utilizing high-yield varieties of crops, managing water efficiently, maintaining nutrition for plants, and integrating pest control measures.

3.1.2 Varieties and Planting Materials

Promising advancements in cassava breeding are evident in the identification of beneficial mutations by scientists at CIAT. One such mutation involves root starch with little to no amylose content, known as "waxy starch," which has valuable industrial applications. The Thai Tapioca Development Institute is incorporating this trait into high-yielding commercial cassava varieties. Additionally, a mutation resulting in smaller starch granules with a rough outer surface has potential in the fuel-ethanol industry, as it simplifies starch-to-sugar conversion for ethanol fermentation.

Ongoing research at CIAT involves the use of molecular tools for cassava genetic enhancement. Molecular markers are being employed to trace the inheritance of resistance to pests and diseases like whiteflies, green mites, and bacterial blight. Molecular markers linked to a gene for cassava mosaic disease resistance are facilitating the selection of resistant varieties. This approach has enabled the transfer of Latin American cassava genotypes to African breeding programs, enhancing resistance to CMD.

Quality planting materials are essential for cassava production. Thailand has achieved success in disseminating improved varieties through a program launched in 1994. This program, involving various departments and institutes, led to almost 90 percent of the nation's cassava area being cultivated with recommended cultivars by 2000.

Efforts to enhance cassava stem production efficiency are noteworthy. IITA and Nigeria's National Root Crops Research Institute have developed a rapid multiplication technology involving shorter stem cuttings with 2 to 3 nodes. With proper management, these stems can be harvested twice a year, resulting in a significant increase in stem yield. This technology has empowered farmers, with many in Nigeria using it to multiply stems of improved varieties, generating average annual earnings of $ 750 from stem sales.

3.1.3 Pests and Diseases

[32] advocates for a balanced agroecosystem as the primary line of defense against crop pests and diseases. It emphasizes reducing synthetic pesticide use and promoting integrated pest management (IPM) strategies that enhance natural processes and biodiversity, supporting crop production. This involves deploying resistant plant varieties, encouraging biological control agents, and managing crop nutrient levels for insect reduction.

Cassava, susceptible to pests and diseases, particularly in Africa, is best safeguarded through non-chemical means. Using healthy planting material, hot water treatment, and crop rotation helps control bacterial blight, a severe disease. For viral diseases like cassava

mosaic disease (CMD) and brown streak disease (CBSD), practicing quarantine procedures and using virus-free planting material are recommended. Innovative research has led to CMD- and CBSD-resistant cassava varieties.

Weeds pose another challenge to cassava. Cultural practices like using vigorous planting material, planting under mulch, and intercropping with fast-growing plants help reduce weed competition. Mechanical methods like hoeing and using animals or tractors for cultivation are effective. Herbicides can also be used with caution, with pre- and post-emergence options available.

The success of biological control is evident in countering pests like mealybugs and green mites, which harm cassava in various regions. The introduction of natural predators and parasites effectively curbs their population. Additionally, targeted intercropping, such as cassava with cowpeas, has proven beneficial.

Strategies to control pests and diseases while ensuring sustainable cassava production involve a holistic approach that respects ecological balances and local conditions. These approaches, encompassing various non-chemical interventions and carefully considered use of pesticides, can secure cassava yield while minimizing environmental harm.

3.2 Model Example

This model has been implemented and has been inspired by [14]

3.2.1 Initialization

The initialization part is made of two subsections. The first subsection is the initialization of all environmental conditions. Because the model is made on a weekly basis, we need a second initialization to set the crop establishment parameters better known as the plant onset.
The first initialization requires 52 sets of values (representing 12 months and 52 weeks of growth) of all environment inputs. The environmental inputs will be assessed once. The parameters taken into consideration are solar radiation (Sr), air temperature, rainfall, pan evaporation, and day length (D). In order to determine the day length, we need the culture location and latitude. Pan evaporation is necessary to establish soil water balance, and for that, we also specify the following soil parameters for the three first soil layers: depth,

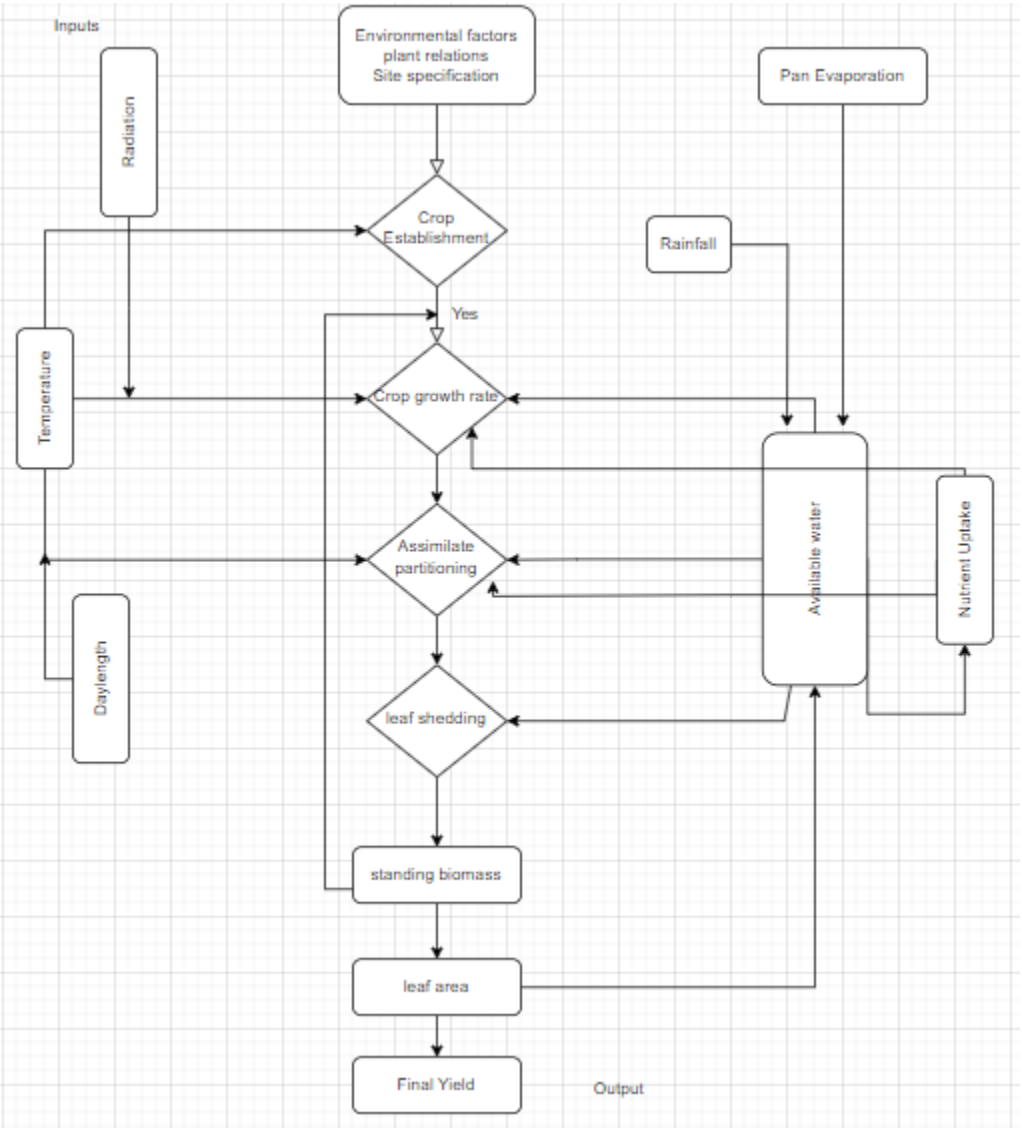


Figure 3.1: Cassava process-based flowchart [14]

average bulk density, field capacity, wilting point, and moisture content.
The second initialization represents the crop establishment. It has been verified that the minimum accumulated temperature needed for establishment is 16.7. For our crop to start growing, we need to calculate the heat sum. When the heat sum reaches 16.7, then the plant onset is met.

$$heat\_sum = \sum_{n=1}^{m} \sqrt{T - 16} \tag{3.1}$$
$$heat\_sum \geq 16.7$$

$$m : number\ of\ minimum\ weeks\ required\ for\ establishment.$$

For example, a constant temperature of 25°C will take 6 weeks to establish.
We do not account for water stress prior to onset. Following [22]'s methodology, our model initializes parameters at establishment. The initial parameters are as follows: leaf area index (LAI) is set at 0.5, stem at 7, petiole at 5, lamina at 20, tuber at 0, planting piece at 37, and total dry matter at 69. The data for these parameters were sourced from [22]. Then we make a loop for 52 iterations of the following blocks.

3.2.2 Leaf Area Index (LAI)

To calculate the LAI daily, we use leaf Lamina dry weight and create a polynomial extrapolation of a specific leaf area.

$$LAI = lldw \cdot sla \tag{3.2}$$

where
lldw: leaf lamina dry weight
sla: specific leaf area.

3.2.3 Soil Water Balance

In plant growth, environmental conditions like water availability and pan evaporation can add stress to the plant. To determine the amount of stress these conditions add, we calculate a stress index (SI). The stress index is a function of the potential extraction and potential transpiration.

$$SI = \frac{Pt - Pex}{Pt} \tag{3.3}$$

Where
Pt: Potential transpiration
Pex: Potential extraction
Note: SI should always be positive. If the result is negative, which means that Potential extraction is superior to potential transpiration, we assume $SI = 0$. In that case, there is no water stress on the crop.
Soil evaporation is needed to calculate the potential transpiration. It is related to the ground cover proportion and days separating rainfalls.
The ground cover proportion is:

$$scov = 1 - e^{(-K \cdot LAI)} \tag{3.4}$$

where
K=0.8
and LAI is calculated daily

From this, we can obtain the potential soil evaporation:

$$PE = epan \cdot (1 - scov) \tag{3.5}$$

where
epan: pan evaporation
With this information, we use the Ritchie procedure to determine the soil evaporation amount.
The soil evaporation amount helps to determine if the soil is dry or not. According to this

factor, the potential transpiration calculus method may change To calculate the potential transpiration we deduce:

$$Pt = epan \cdot scov \tag{3.6}$$

For a dryer soil surface, we take another path:

$$Pt = epan \cdot Gcov \tag{3.7}$$

where

$$Gcov = 1 - e^{(-cf \cdot LAI)} \tag{3.8}$$

$$For\ this\ particular\ case,\ cf = 1$$

We should keep in mind that the root depth starts at 70 cm at the establishment and is increased by 7 cm each week until the maximum depth of the third layer, which is approximately 120cm. With this information in mind, we can calculate the uptake, "the amount of moisture that can be extracted. The sum gives the potential extraction.

$$Pex = \sum_{n=1}^{3} UPTAKE_i \tag{3.9}$$

$$UPTAKE_i = \alpha_i \cdot (PAWP_i)^{\beta} \tag{3.10}$$

Where i is for the i^{th} layer, We calculate the uptake for each soil layer.
$\beta = 1.67$
$\alpha_1 = 3.2$; $\alpha_2 = 5.6$; $\alpha_3 = 6.7$
$PAWP_i$ is the plant's available water proportion for each layer.

3.2.4 Crop Growth Rate

The crop growth rate (CGR) is a function of LAI, solar radiation, and temperature. Under optimum conditions which mean $temperature > 24^{\circ}C$ and solar radiation $> 22MJ \cdot m^2$, the i deal CGR is as follow:

$$ICGR = a - b \cdot e^{(-c \cdot LAI)} \tag{3.11}$$

Where
$a = 21.7$; $b = 20.5$; $c = 0.27$

Because perfect conditions do not exist, the model needs to take into consideration the stress induced by water, temperature, and radiation. For that reason, we did a polynomial extrapolation to determine the multiplier index for each parameter. With this update, the CGR functions change.

$$CGR = PCGR \cdot (tm \cdot rm \cdot wsm) \tag{3.12}$$

Where
tm: temperature index multiplier
rm: radiation index multiplier
wsm: water stress index multiplier

3.2.5 Assimilate Distribution

After CGR, we estimate the assimilate distribution among the different organs. This distribution is affected by temperature, day length, LAI, and water stress index. The water stress polynomial extrapolation is unique for assimilating distribution. This relationship is expressed as follows:

$$DRS = [0.011T + 0.0136LAI + 0.0637 \cdot (D - 10)] \cdot wsm \cdot fm \tag{3.13}$$

Where:
DRS: Distribution ratio to shoot
T: Temperature
D: Day length
wsm: water stress multiplier
fm: fertilizer multiplier

$$fm = 0.65$$

if no fertilizer is applied and

$$fm = 1$$

if there is fertilizer

3.2.6 Nutrients

The determination of the plant's requirements involves combining the needs of each part (leaves, stem, roots). This calculation is based on the difference between the minimum levels and the maximum levels in each part of the plant.

$$Act_{NPK} = DRS \cdot 0.3 \cdot CaL_{NPK} + 0.7 \cdot DRS \cdot CaST_{NPK} + (1 - DRS)CaRT_{NPK} \quad (3.14)$$

CaL_{NPK} = Concentration of actual NPK in leaves.
$CaST_{NPK}$ =Concentration of actual NPK in the stem.
$CaRT_{NPK}$ =Concentration of actual NPK in roots.

The changes seen in the highest levels of plant parts concentrations depend on the impact of temperature. It's crucial to have the recorded data indicating the largest amounts of nitrogen (N), phosphorus (P), and potassium (K) in leaves, stems, and storage organs for examination.

$$Max_{NPK} = DRS \cdot 0.3 \cdot CmL_{NPK} + 0.7 \cdot DRS \cdot CmST_{NPK} + (1 - DRS)CmRT_{NPK} \quad (3.15)$$

CmL_{NPK} = Concentration maximum for leaves.

$CmST_{NPK}$ = Concentration maximum for the stem.

$CmRT_{NPK}$ = Concentration maximum for roots.

To determine the requirement and provision of nutrients, we calculate the combined absorption of nutrients by the plant, taking into account that the uptake of each nutrient depends on the presence of other nutrients. The actual uptake is as follows:

$$Uptake = 0.012 \cdot NED \tag{3.16}$$

NED : Nutrient equivalent demand

$$NED = Max_N \cdot [1 - Act_N + \frac{Mac_P - Act_P}{Max_P} + \frac{Max_K + Act_K}{Max_K}] \tag{3.17}$$

From all these equations, we can determine the stress index induced by nutrient uptake:

$$NI = \frac{Act_{NPK} - Min_{NPK}}{0.8 \cdot (Max_{NPK} - Min_{NPK})} \tag{3.18}$$

Min_{NPK} : Minimum available nutrient

Chapter 4

Implementation and Results

Machine learning (ML) models can be classified into two simulation categories: mathematical and process-based, depending on their construction and selection criteria. Mathematical models are characterized by linear relationships, making linear ML models well-suited for this approach. On the other hand, process-based ML models are made of non-linear techniques like random forests, K-NN, and neural networks. Moreover, each process can be implemented as a service to accurately depict dynamic interactions within a crop system. In our experiment, we employ non-linear models to capture the intricate relationships among various processes. While not strictly process-based, it serves as a valuable benchmark for comparison.

4.1 Process-based Model: Results and Observations

The objective of this section entails the formulation of a simulation model for the growth of cassava, grounded in [22]'s research. To execute this, a Python-based software employing [22]'s acquired data will be employed for thorough testing. Presently, the prevalent software for akin modeling purposes is developed in FORTRAN. The essence of smart farming resides in the judicious allocation of resources vis-a-vis the attained outcomes. Hence, the current endeavor is singularly directed at devising the most effective methodology for emulating [22]'s study while systematically integrating factors such as temperature, water stress, and radiation.

Data. We conducted two distinct simulations. The initial simulation was executed using the process-based model. In this implementation, we utilized environmental data

from [26] over a period of ten years, enabling us to perform the procedures outlined in the preceding chapter to construct the model.

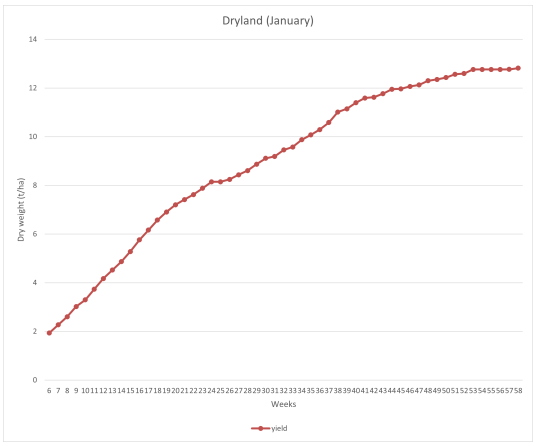
Within the scope of this experimentation, we have diligently reconstructed a model in Python with maximal precision. Subsequent to its development, a series of simulations were executed, furnishing discernible outcomes. The critical parameters to be observed during these simulations encompass the timing of sowing (winter, summer, etc.) and the intricate interplay of irrigation. It is presupposed that in each of these test scenarios, the fertilizer quantity remains consistent.

4.1.1 No Irrigation

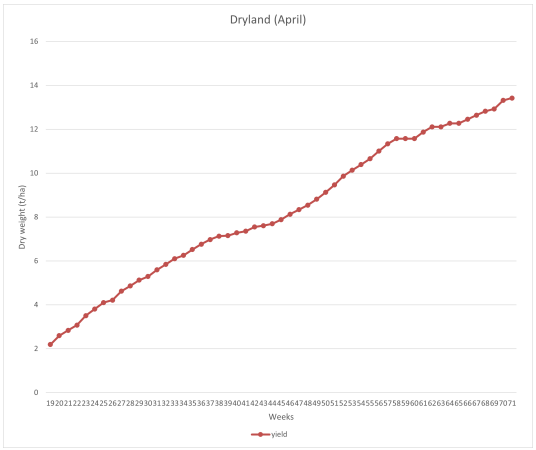
The conducted experiment exhibited a notable resemblance to the outcomes derived from [22] research. While the graphical representations differ, they fundamentally align in terms of the underlying trend. In our specific case, in the absence of irrigation, a dry weight of 12.5 tons per hectare was attained. [22], in his experimental work, had yielded a value of 13 tons per hectare. Notably, this value exhibited fluctuations contingent upon geographical location and season, spanning a range from 11 to 15 tons per hectare. The discernible variance between our outcomes and [22] findings can be readily attributed to differences in calibration protocols. Specifically, our experiment employed rainfall data from 2010, whereas [22] data dates back to 1975. Additionally, certain aspects pertaining to soil type and its capacity were approximated in our study.

Two distinct simulations were conducted, one during the Australian summer, precisely in January, and the other during the winter month of May. In the summer simulation, the plant initiated growth within 6 weeks and matured over approximately 40 weeks to reach a stabilized state. Conversely, in the winter simulation, the target stabilization point was not fully realized, necessitating a duration of 70 weeks to approach outcomes akin to those achieved during the summer simulation. The model's capacity to effectively mirror the contextual conditions is indeed a promising facet of this research endeavor.

Cassava, predominantly grown in tropical regions, our study encountered limitations regarding data availability and model specifications. While the Fukai model proved advantageous in terms of simplicity for modifications, adapting it to non-Australian environments would required substantial parameter adjustments, therefore comprehensive environmental data was needed. Fortunately, the Australian government provided comprehensive environmental datasets for our study. We also tested during the Australian winter to evaluate the model's capacity to accommodate extreme environmental stress. Notably, no onset



(a) simulation for summer



(b) simulation for Winter

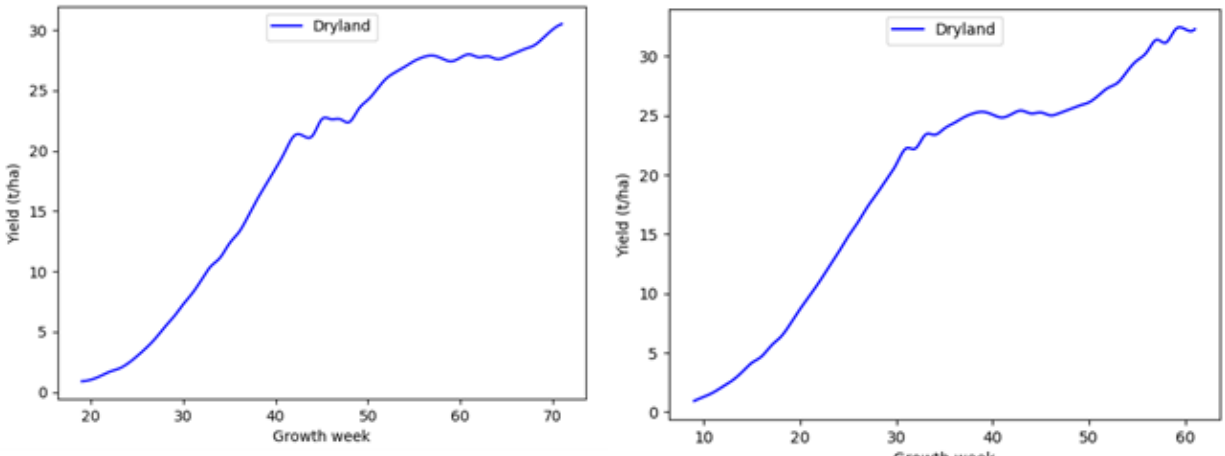
Figure 4.1: Dryland Simulation

occurred during the winter season, with the onset process starting after 19 weeks of simulation. Winter stress levels were also assessed between weeks 20 and 40 of the summer simulation.

4.1.2 Irrigated

In the context of irrigation, the influence of water-related stress is omitted from consideration. Instead, the model factors in stress arising from solar radiation and temperature. The resultant simulation data yields an outcome that is notably consistent. While [22] findings stood at 26 tons per hectare, our simulation generated a close figure of 25.8 tons per hectare. Notably, the trajectory of the curve exhibits some divergence from [22] rendition. Our analysis attributes this disparity primarily to the calculation of the Leaf Area Index (LAI). [22] LAI data were meticulously measured and computed daily at the site. Conversely, we resorted to an approximation technique to estimate LAI based on acquired data. In the majority of similar studies, LAI was determined through manual measurements and on-site calculations.

The model's capacity to achieve results almost on par with [22] outcomes is certainly a promising aspect.



(a) simulation for winter

(b) simulation for spring

Figure 4.2: Fully irrigated simulation

4.1.3 Analysis by Season

Our observations have yielded the following findings. In our simulation, the point of maximum growth is attained at the 40-week mark. Any growth beyond this juncture is extrapolated rather than grounded in data. This distinction arises due to [22] data, which involved certain aspects calculated on-site, as opposed to our approach of estimation. Consequently, the curve in [22] reflects a higher degree of precision. Regrettably, the formulas and estimates provided by [22] lack the capacity to incorporate negative feedback.

To elaborate further, the plant exhibits a growth period of approximately 40 weeks. Subsequent to this phase, growth stagnates temporarily before declining. However, the model inadequately captures this degenerative trend. The model's representation of the negative feedback caused by degeneration is insufficient to curtail the overall growth of the plant. Weekly growth calculations are appended to the total growth, but circumstances necessitating a negative weekly growth significant enough to impact total growth are uncommon and extreme in reality. Nonetheless, this limitation doesn't significantly undermine the model's utility. The plant's inherent resilience to extreme conditions means that this degeneration effectively illustrates the plant's response to reasonable drought conditions. This explains why our simulations show some development even after 40 weeks. We've retained these results as they facilitate the analysis of the duration of plant development under varying conditions.

The subsequent figures depict the plant's development under irrigated conditions at different times of the year. The simulation conducted in spring reveals that a plant achieves its peak yield of 26 tons per hectare between 28 and 35 weeks. Conversely, a winter planting attains its maximum yield of approximately 21 tons per hectare at the 40-week mark, requiring around 65 weeks to virtually reach a similar outcome. The summer simulation also converges at a result of about 26 tons per hectare within 40 weeks. These findings align with observations made by [22].

4.2 Machine Learning: Results and Analysis

The second implementation is rooted in a machine learning model. The data fed into these models primarily originate from our earlier simulation.

In preparation for the machine learning simulation, we applied data normalization to the acquired dataset. The primary goal of data normalization is to bring all features to a similar scale without distorting their original distributions. This ensures that no single feature dominates the analysis or affects the performance of machine learning algorithms disproportionately. Additionally, normalization can improve the convergence speed and accuracy of machine learning models, leading to better performance overall. Min-max normalization was used in our case. This dataset was acquired after an initial simulation throughout **ten years**. Among the numerous factors influencing crop yield prediction, temperature, radiation, and rainfall have a considerable impact in general. The key features considered here as inputs for this simulation encompassed week, temperature, average weekly rainfall, average weekly evaporation, and radiation. The output variable focused on yield.

4.2.1 Method

For this exercise, we used different models to compare the results of the predictions. The mean squared error method has been used for performance review. You can see below the algorithms used and their parameters:

1) **Neural Networks (ANN):** Neural Networks are made up of interconnected processing units called neurons, which collaborate to process information and make predictions or decisions. These networks are composed of layers of interconnected neurons, allowing them to capture relationships in data that linear models may struggle with. One key advantage of Neural Networks is their ability to automatically learn features from raw

data, reducing the need for manual feature engineering. This becomes especially beneficial when dealing with large data. Additionally, Neural Networks can be scaled up to handle large datasets and complex tasks by increasing the number of neurons, layers, or model parameters.

In the context of this study, we have developed two Neural Network models for comparison. The first model was constructed utilizing the Keras library, whereas we manually crafted all the libraries for the second model. The training process took approximately ten minutes for the Keras model and forty minutes for the other model. The parameters are presented in the table below.

Table 4.1: Model Keras: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 360)	2160
$\text{dense}_1(Dense)$	(None, 1)	361
Total params: 2,521 Trainable params: 2,521 Non-trainable params: 0 Best Optimizer: SGD*	learning rate=0.01,	momentum=0.7

SGD: Stochastic gradient descent

Table 4.2: Model from scratch: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 1024)	6144
$\text{dense}_1(Dense)$	(None, 1)	1025
Total params: 7169 Trainable params: 7169 Non-trainable params: 0 Best Optimizer: adam	learning rate=0.01, $\text{beta}_1 = 0.9$,	$\text{beta}_2 = 0.999, \epsilon = 1e-8$

2) **K-nearest neighbors (KNN):** is a type of machine learning algorithm that is commonly used for both classification and regression tasks. It is an easy-to-understand algorithm that works by predicting the class of a data point based on the classes of its neighbors. The choice of a value for K, which represents the number of neighbors to consider, is crucial as it can significantly impact the algorithm's performance. If K is set small, it may result in noisy predictions, whereas if it is set too large, it may lead to overly smoothed predictions. One advantage of KNN is that it does not make any assumptions about the distribution of the underlying data. Due to its simplicity and ease of implementation, KNN often serves as a choice for simple regression tasks but may not be ideal for large and complex datasets.

In this case, we implemented a KNN model based on our dataset. After multiple tries, it has been concluded that the best value for K is 12. You can find below our parameters:

parameters: K=12

$$\{'algorithm' :' auto',$$
$$'leaf\_size' : 30,$$
$$'metric' :' minkowski',$$
$$'metric\_params' : None,$$
$$'n\_jobs' : None,$$
$$'n\_neighbors' : 12,$$
$$'p' : 2,$$
$$'weights' :' uniform'\}$$

3) **Random Forest (RF):** is a method used in machine learning that combines multiple models to enhance performance. It's well regarded for its ability to provide reliable predictions. Random Forest employs a combination of decision trees and randomization techniques to make predictions. In regression tasks, each tree predicts a numeric value, and the final prediction is determined by averaging these values. This approach tends to yield results while mitigating the risk of overfitting that can arise with decision trees. Additionally, Random Forest is effective in handling data, outliers, and missing values. It also offers insights into feature importance, which can be valuable for feature selection and gaining an understanding of the data. During the training, it was revealed that the best parameters are n_estimator=100. Below is an extensive list of parameters.

Parameters:

$$\{'bootstrap' : True,$$
$$'ccp\_alpha' : 0.0,$$
$$'criterion' :' squared\_error',$$
$$'max\_depth' : None,$$
$$'max\_features' : 1.0,$$
$$'max\_leaf_nodes' : None,$$
$$'max\_samples' : None,$$
$$'min\_impurity\_decrease' : 0.0,$$
$$'min\_samples\_leaf' : 1,$$
$$'min\_samples\_split' : 2,$$
$$'min\_weight\_fraction\_leaf' : 0.0,$$
$$'n\_estimators' : 100,$$
$$'n\_jobs' : None,$$
$$'oob\_score' : False,$$
$$'random\_state' : 42,$$
$$'verbose' : 0,$$
$$'warm\_start' : False\}$$

4.2.2 Observations

- **Irrigated sensitivity test result.**

The two Neural Networks were implemented in Python using the Keras software library for one and We constructed the other model from the ground up. The aim was to access additional parameters and evaluate their potential impact. We also implemented two other popular prediction models for comparison: Random Forest Regression and KNN regression. All of these models were implemented in Python in the most efficient manner that we were capable of and tested under the same software and hardware environments to ensure fair comparisons. The following hyper parameters were used for the random forest. The maximum depth of the tree was not set, due to the amount of data, we were not particularly worried about overfitting. We set the number of estimators at 100. All

features were used to train the random forest. The tables below compare the performances of the four models on validation datasets with respect to the Mean Squared Error (MSE), Root Mean Squared Error(RMSE), and Mean Absolute Error (MAE). The motivation stems from the fact that for certain datasets, relying solely on MSE may not suffice. While MSE is indeed valuable, it may have certain limitations that need to be considered. These results suggest that the random forest outperformed the other three models to varying extents. We note that the random forest model achieves the lowest MSE value of 1.35, outperforming all other models. Specifically, KNN has an MSE of 1.89, while the neural network with Keras achieves 1.78, and the other neural network records 1.95. In our case, the analysis based on MSE values is backed up by RMSE and MAE values (Table 4.3).

Despite the superior performance of the random forest model in yield prediction, it did not exhibit superiority across all aspects. Notably, Neural Networks (NNs) demonstrated better prediction accuracy at extreme values. It was observed that, overall, all models struggled with precision in predicting central values, a deficiency notably accentuated in the case of KNN. These observations can be derived from the data presented in Figure 4.3.

The judicious selection of optimal parameters has proven immensely beneficial in reducing the learning and validation losses within the Keras-based model. Additionally, the analysis of feature impact within the random forest model shed light on the significant influence of the weeks of growth. This particular parameter was intentionally integrated into the model to regulate the growth process.

Table 4.3: Sensitivity test result

MSE Values						
	no test	week	Temperature	Radiation	evaporation	Daily rain
NN Keras	1.78	54.98	2.00	1.77	1.63	1.77
NN other	1.95	74.62	1.81	1.78	1.65	1.64
Random Forest	1.35	69.13	1.61	1.46	1.38	1.34
KNN	1.89	67.23	2.22	2.06	1.64	1.88
RMSE Values						
	no test	week	Temperature	Radiation	evaporation	Daily rain
NN Keras	1.33	7.42	1.42	1.33	1.28	1.33
NN other	1.40	8.64	1.34	1.34	1.29	1.28
Random Forest	1.16	8.31	1.27	1.21	1.17	1.16
KNN	1.37	8.20	1.49	1.43	1.28	1.37
MAE Values						

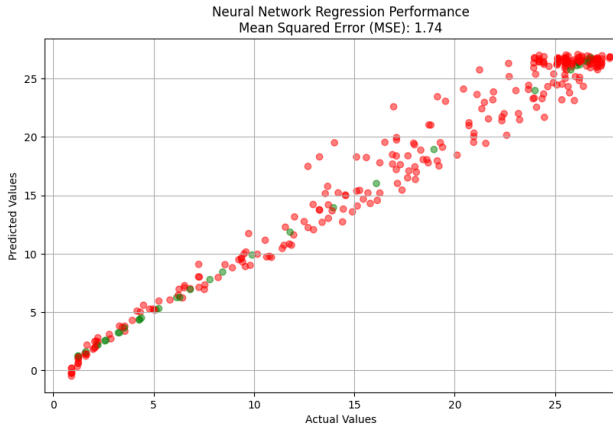
	no test	week	Temperature	Radiation	evaporation	Daily rain
NN Keras	1.01	6.06	1.09	1.02	0.94	1.03
NN other	1.05	5.58	1.00	1.00	0.93	0.98
Random Forest	0.85	5.44	0.90	0.88	0.85	0.85
KNN	1.04	5.10	1.10	1.07	0.98	1.04

We conducted supplementary assessments to analyze the influence of individual features within each model. Our objective was to ascertain the sensitivity of each model to specific features by systematically modifying or excluding certain parameters and observing their respective reactions to these alterations.

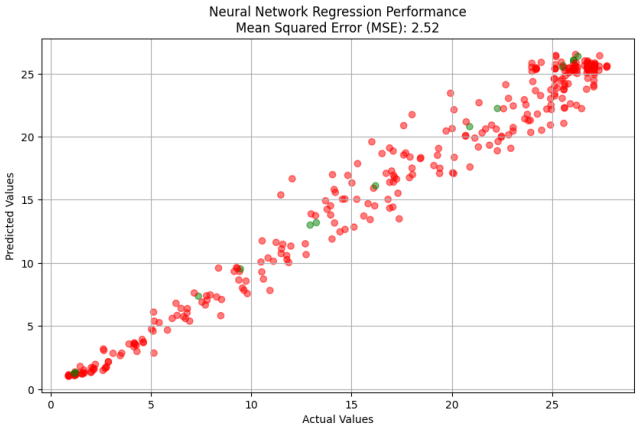
Upon varying the values associated with the week of growth and even removing it entirely, we noted a pronounced sensitivity of all models to this particular feature. Eliminating this feature significantly compromised the predictive capacity of the models, rendering them less effective in their predictive abilities. On the other hand, the alternative Neural Network (NN) shows sensitivity primarily to the growth week, followed by temperature, radiation, evaporation, and rainfall. The absence of each variable contributes to enhanced precision in its predictions. The previous deduction was done based on table 4.3 data.

Upon analyzing the data, we can re-evaluate the models from another perspective. In particular, when it comes to the Keras network, we notice that the growth week has a significant influence on the precision of the model. Following are factors such as radiation, evaporation, temperature, and lastly rainfall. compared to the values in the no sensitivity test (mse=1.78) It is worth noting that both rainfall and temperature have a minimal impact on the overall results. Random Forest, however, displays limited sensitivity to daily rainfall and evaporation variables. It demonstrates a modest sensitivity to temperature, followed by radiation, while maintaining a significant sensitivity to the growth week. KNN reveals substantial sensitivity to the growth week, temperature, evaporation, and radiation in that order compared to the values in the no sensitivity test (mse=1.89) Its sensitivity to rainfall shows minimal to no variations.

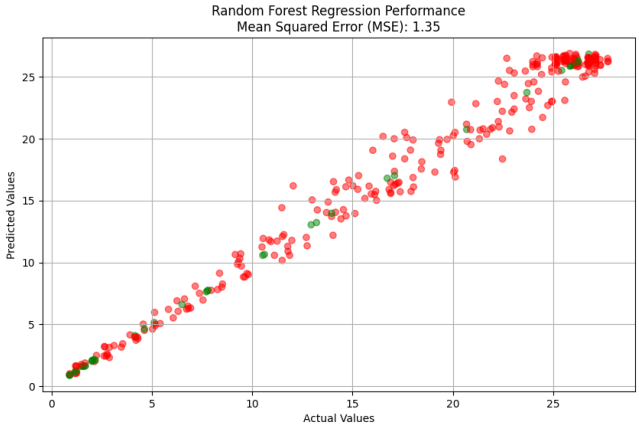
The evaporation component consistently ranks third as the influential factor after the week of growth. The primary position immediately following the growth week is shared by both temperature and radiation. Temperature exerted a moderate impact on prediction accuracy across all models. In general, each model experienced a marginal decrease in precision, except for the NN constructed from alternative libraries, which exhibited a slight



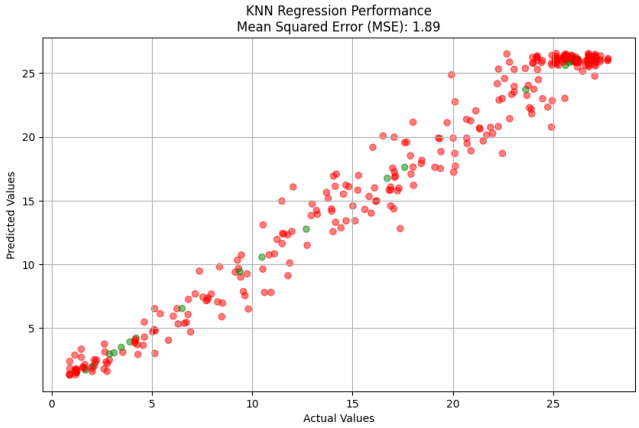
(a) Neural Network regression performance (Keras)



(b) Neural Network regression performance (from scratch)



(c) Random forest regression performance



(d) KNN regression performance

Figure 4.3: Mean Squared error evaluation

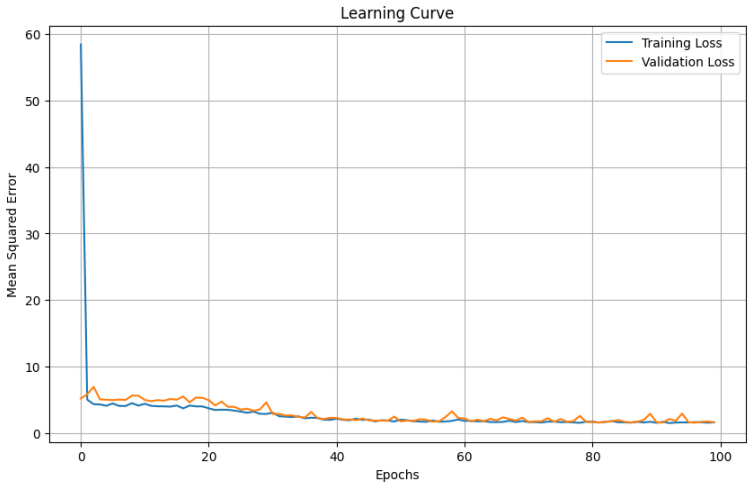


Figure 4.4: NN learning curve (Keras)

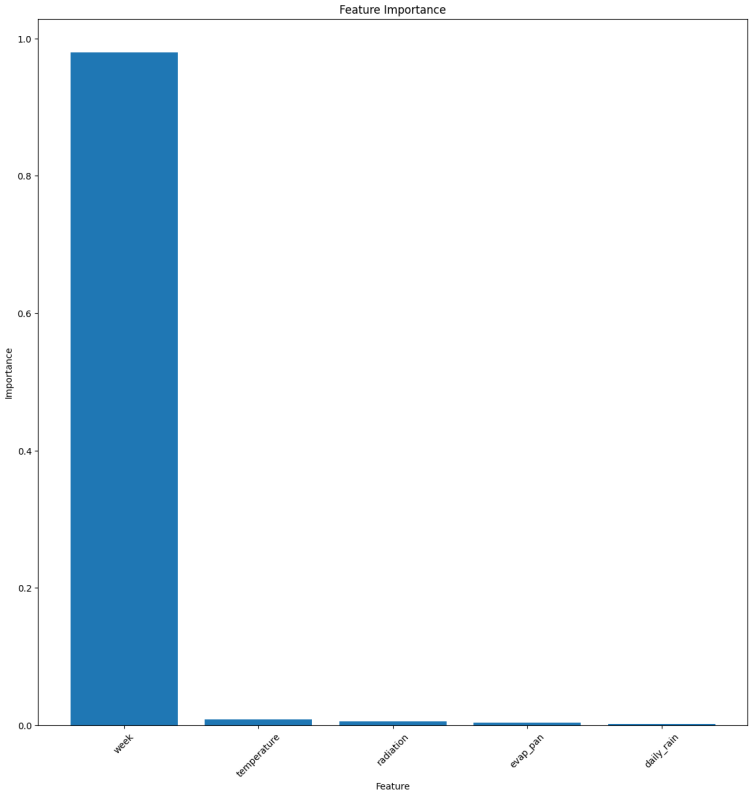


Figure 4.5: The Random Forest feature's importance

improvement in accuracy. Notably, the NN generated from Keras showcased a heightened sensitivity to the radiation feature, resulting in a larger mean square error (MSE). Moreover, nearly all models exhibited improved precision in the absence of daily rainfall. For those models that did not notably improve in precision, there was no marked decline either. This observation aligns with the Fukai experiment, upon which our dataset is based, which frequently implemented irrigation to mitigate the impact of drought conditions. As a result, when the daily rainfall is excluded from the model, we observe an improvement in accuracy. This highlights the situations in which irrigation was used to mitigate the impact of drought.

- **Drought sensitivity test result.**

Table 4.4: drought sensitivity test result

MSE Values						
	No test	week	Temperature	Radiation	evaporation	Daily rain
NN Keras	1.20	8.42	1.06	1.11	1.44	1.19
NN other	1.47	10.67	1.90	1.47	0.86	1.98
Random Forest	1.37	9.30	1.30	1.25	1.07	1.47
KNN	1.24	8.54	1.32	1.22	0.95	1.27
RMSE Values						
	No test	week	Temperature	Radiation	evaporation	Daily rain
NN Keras	1.10	2.90	1.03	1.07	1.19	1.08
NN other	1.21	3.27	1.38	1.21	0.94	1.39
Random Forest	1.17	3.05	1.14	1.12	1.03	1.21
KNN	1.11	2.92	1.15	1.11	0.97	1.13
MAE Values						
	No test	week	Temperature	Radiation	evaporation	Daily rain
NN Keras	0.76	2.23	0.73	0.87	0.78	0.74
NN other	0.90	2.40	1.0.3	0.89	0.70	1.06
Random Forest	0.85	2.32	0.84	0.84	0.77	0.87
KNN	0.82	2.14	0.89	0.84	0.73	0.82

In this section, it becomes evident that the growth week significantly influences the

models. Its absence leads to challenges in accurate prediction across all models. Notably, the Keras NN model demonstrates sensitivity primarily influenced by evaporation, followed by temperature and radiation. Rainfall has minimal impact on this specific model. For the other Neural Network model, it appears that evaporation is the most significant factor, followed by rainfall, temperature, and then radiation. The random forest is affected by evaporation, radiation, rainfall, and temperature, while the KNN model responds to evaporation, temperature, rainfall, and radiation, although the distinctions in the effects of rainfall, temperature, and radiation are relatively negligible. The consistent impact of the growth week and evaporation on all models is apparent from the observations. These factors significantly affect the performance of all models. Following this trend, radiation and rainfall share a subsequent position in terms of impact.

In conclusion, based on the analysis, the growth week has a predominant impact on predicting all models regardless of other factors due to the cumulative effect of plant growth. In scenarios of drought moderated by irrigation, models tend to rely more on radiation (or, in specific cases, temperature) and evaporation, with rainfall exerting minimal influence. Random forest was the best performer in these conditions. During unmitigated drought conditions, models prioritize evaporation, followed by radiation and rainfall, while temperature appears to have minimal impact in these conditions. Neural Network built with Keras was a better performer.

The substantial impact of this parameter in the model can be rationalized by the data portraying a crop relatively resilient to drought conditions. Consequently, the influence of environmental variables was less pronounced. Nevertheless, these acquired data provide valuable insights into the plant's development under stable conditions.

4.3 Application

After investing considerable effort, it would be valuable to elaborate on our vision for implementing this research. As previously mentioned, unmanned aerial vehicles (UAVs) play a pivotal role in this regard. UAVs offer the capability to generate detailed maps of crops, enabling the assessment of various factors such as crop yield and Leaf Area Index (LAI). The LAI, being a crucial parameter for process-based models, underscores the significance of UAVs in this context.

During different stages of crop development, UAVs can be used by farmers for targeted spraying and precise monitoring of crop progress. Equipped with suitable sensors, UAVs facilitate the identification of areas within a crop that require additional watering, thereby positively impacting overall crop productivity. Among the array of sensors,

thermal cameras stand out as commonly employed tools for this purpose. By leveraging thermal imaging to map water potential variability across a field, irrigation practices can be optimized for enhanced efficiency. Moreover, the creation of accurate weed coverage maps through UAV imaging enables targeted herbicide application and facilitates disease detection.

Our research endeavors with UAVs and complementary sensors offer significant benefits to cassava farmers, regardless of their level of technological proficiency. They empower farmers to make informed decisions regarding optimal planting times, suitable crop varieties based on predicted weather patterns and soil characteristics (including nutrient content), and the timely mitigation of development issues such as nutrient deficiencies, weed infestations, diseases, and water stress. Handling these variables undoubtedly translates into higher crop yields.

Chapter 5

Conclusion and Future Work

As part of this thesis, a comprehensive discussion was undertaken regarding a dynamic model formulated to forecast cassava yield. In addition, various machine learning models were deliberated upon, aiming at optimizing yield prediction and facilitating comparisons. The dynamic model accounted for several environmental factors, encompassing weather parameters (such as temperature, radiation, rainfall, and evaporation) as well as soil parameters, which included the application of fertilizers. The discussion surrounding this model illuminated the intricate interactions inherent in the developmental stages of diverse components of the cassava plant. The development of this model facilitated a comparative analysis with its predecessor, which was designed by Fukai. Our model, while reflecting the methods and technologies of the Fukai model, demonstrated significant advancements. Notably, one of these enhancements lies in the effective integration of soil nutrients and the adaptive application of fertilizers based on the specific requirements of the soil and the plant. Therefore, it can be inferred that our model not only serves as a valuable benchmark compared to its predecessors but also introduces improvements, particularly in the realm of fertilizer application.

Additionally, within this study, the development and comparison of four machine learning models were undertaken. This comparative analysis involved assessing the performance of random forest, K-nearest neighbors (KNN), and two distinct constructions of Neural Networks. The outcomes revealed that under irrigation conditions, the random forest model exhibited superior performance, while in drought scenarios, the Neural Network constructed with Keras showed the most suitable predictive capability. However, these conclusions change depending on the environmental variable to be tested. Notably, the most influential environmental factors were identified as evaporation and radiation, which significantly impacted the model's predictive accuracy across varying conditions.

In future endeavors, it would be beneficial to explore a comparison involving a Kalman filter. ML is generally used for making predictions or decisions based on data patterns, while the Kalman filter is used for state estimation in dynamic systems. ML algorithms make fewer assumptions about the underlying dynamics of the system and can handle complex, nonlinear relationships. The Kalman filter assumes linear dynamics and Gaussian noise, which means it's most effective for systems that can be reasonably approximated by linear models with additive Gaussian noise.

Coupling the Kalman filter with machine learning techniques can be beneficial in scenarios where you have a dynamic system with complex, nonlinear behavior, and you want to improve state estimation using both historical data and real-time measurements. This can be achieved in different ways:

- Use machine learning algorithms to learn the relationship between system states and sensor measurements from historical data.
- Use machine learning techniques to adaptively adjust the process noise covariance matrix and measurement noise covariance matrix based on real-time sensor data.
- Use the Kalman filter to provide state estimates and uncertainty estimates as inputs to a machine-learning model.
- Incorporate online learning techniques to continuously update the machine learning models used in conjunction with the Kalman filter.
- Combine the outputs of the Kalman filter and machine learning models using fusion techniques such as ensemble methods or Bayesian model averaging.

By coupling the Kalman filter with machine learning techniques, you can leverage the strengths of each approach to improve state estimation in dynamic systems, especially in scenarios where traditional Kalman filtering assumptions may not hold or where complex nonlinear relationships exist between system states and measurements like in our case. Following this trajectory, the development of a combination of an advanced machine learning model and Kalman filter capable of weekly predictions, accurately assessing plant requirements, and determining optimal nutrient dosage and irrigation needs every week would represent a significant step forward. Such a comprehensive model could be complemented by disease detection functionalities, further enriching its applicability and utility within agricultural contexts.

References

[1] Yaganteeswarudu Akkem, Saroj Kumar Biswas, and Aruna Varanasi. Smart farming using artificial intelligence: A review. *Engineering Applications of Artificial Intelligence*, 120:105899, 2023.

[2] M Amanullah, C Kailasam, A Mohamed Safiullah, S Selvam, K Sivakumar, et al. Crop simulation growth model in cassava. *Research Journal of Agriculture and Biological Sciences*, 3(4):255–259, 2007.

[3] Er Vikram Puri Anand Nayyar. Iot based smart sensors agriculture stick for live temperature and moisture monitoring using arduino, cloud computing & solar technology. *May 2015*, 2015.

[4] Luís Barreto and António Amaral. Smart farming: Cyber security challenges. In *2018 International Conference on Intelligent Systems (IS)*, pages 870–876. IEEE, 2018.

[5] Adeshina Oyedele Bello. Modeling cassava yield: A response surface approach. *arXiv preprint arXiv:1408.0251*, 2014.

[6] Sadok Ben Toumia, Christian Berger, and Hans P Reiser. An evaluation of blockchain application requirements and their satisfaction in hyperledger fabric: A practical experience report. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 3–20. Springer, 2022.

[7] Carolina Bento. Principal component analysis algorithm in real-life: Discovering patterns in a real-estate dataset, 2020.

[8] BWJ Boerboom. A model of dry matter distribution in cassava (manihot esculenta crantz). *Netherlands Journal of Agricultural Science*, 26(3):267–277, 1978.

[9] TF Burks, SA Shearer, RS Gates, and KD Donohue. Backpropagation neural network design and evaluation for classifying weed species using color image texture. *Transactions of the ASAE*, 43(4):1029–1037, 2000.

[10] Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. An overview on edge computing research. *IEEE access*, 8:85714–85728, 2020.

[11] Aniket Chande, Ayush Kumar Chaudhary, Govind Gurme, and Mrunalini Bhandarkar. Iot based smart farming system. *Technology (IRJET)*, 3(3):1, 2014.

[12] Eugene Charniak. *Introduction au deep learning*. Dunod, 2021.

[13] Surabhi Chouhan, Divakar Singh, and Anju Singh. An improved feature selection and classification using decision tree for crop datasets. *International Journal of Computer Applications*, 142(13), 2016.

[14] James H Cock, D Franklin, G Sandoval, and P5 Juri. The ideal cassava plant for maximum yield 1. *Crop Science*, 19(2):271–279, 1979.

[15] DJ Connor and JH Cock. Response of cassava to water shortage ii. canopy dynamics. *Field Crops Research*, 4:285–296, 1981.

[16] DJ Connor, JH Cock, and GE Parra. Response of cassava to water shortage i. growth and yield. *Field Crops Research*, 4:181–200, 1981.

[17] Rahul Dagar, Subhranil Som, and Sunil Kumar Khatri. Smart farming–iot in agriculture. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1052–1056. IEEE, 2018.

[18] Konstantinos Demestichas, Nikolaos Peppes, Theodoros Alexakis, and Evgenia Adamopoulou. Blockchain in agriculture traceability systems: A review. *Applied Sciences*, 10(12):4113, 2020.

[19] KS Ezui, PA Leffelaar, AC Franke, A Mando, and KE Giller. Simulating drought impact and mitigation in cassava using the lintul model. *Field Crops Research*, 219:256–272, 2018.

[20] Muhammad Shoaib Farooq, Shamyla Riaz, Adnan Abid, Kamran Abid, and Muhammad Azhar Naeem. A survey on the role of iot in agriculture for the implementation of smart farming. *Ieee Access*, 7:156237–156271, 2019.

[21] Fabrice Nolack Fote, Saïd Mahmoudi, Amine Roukh, and Sidi Ahmed Mahmoudi. Big data storage and analysis for smart farming. In *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)*, pages 1–8. IEEE, 2020.

[22] S Fukai and GL Hammer. A simulation model of the growth of the cassava crop and its use to estimate cassava productivity in northern australia. *Agricultural Systems*, 23(4):237–257, 1987.

[23] Luana Fernandes Gabriel, Nereu Augusto Streck, Debora Regina Roberti, Zeferino Genésio Chielle, Lilian Osmari Uhlmann, Michel Rocha da Silva, and Stefanía Dalmolin da Silva. Simulating cassava growth and yield under potential conditions in southern brazil. *Agronomy Journal*, 106(4):1119–1137, 2014.

[24] I Ghosh and RK Samanta. Teapest: An expert system for insect pest management in tea. *Applied Engineering in Agriculture*, 19(5):619, 2003.

[25] H Gijzen, HJ Veltkamp, J Goudriaan, and GH De Bruijn. Simulation of dry matter production and distribution in cassava (manihot esculenta crantz). *Netherlands Journal of Agricultural Science*, 38(2):159–173, 1990.

[26] Queensland Government. Open data portal, 2023.

[27] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer, 2003.

[28] J Guruprakash and Srinivas Koppu. Ec-elgamal and genetic algorithm-based enhancement for lightweight scalable blockchain in iot domain. *IEEE Access*, 8:141269–141281, 2020.

[29] AP Gutierrez, B Wermelinger, F Schulthess, JU Baumgaertner, HR Herren, CK Ellis, and John S Yaninek. Analysis of biological control of cassava pests in africa. i. simulation of carbon, nitrogen and water dynamics in cassava. *Journal of Applied Ecology*, pages 901–920, 1988.

[30] Gerrit Hoogenboom, Cheryl H Porter, Kenneth J Boote, Vakhtang Shelia, Paul W Wilkens, Upendra Singh, Jeffrey W White, Senthold Asseng, Jon I Lizaso, L Patricia

Moreno, et al. The dssat crop modeling ecosystem. In *Advances in crop modelling for a sustainable agriculture*, pages 173–216. Burleigh Dodds Science Publishing, 2019.

[31] Jian Hou, Huijun Gao, and Xuelong Li. Dsets-dbscan: A parameter-free clustering algorithm. *IEEE Transactions on Image Processing*, 25(7):3182–3193, 2016.

[32] Reinhardt Howeler, NeBambi Lutaladio, and Graeme Thomas. *Save and grow: cassava. A guide to sustainable production intensification*. Fao, 2013.

[33] Kuo-Yi Huang. Application of artificial neural network for detecting phalaenopsis seedling diseases using color and texture features. *Computers and Electronics in agriculture*, 57(1):3–11, 2007.

[34] Godwin Idoje, Tasos Dagiuklas, and Muddesar Iqbal. Survey for smart farming technologies: Challenges and issues. *Computers & Electrical Engineering*, 92:107104, 2021.

[35] Bert H Janssen, FCT Guiking, Dirk van der Eijk, Eric MA Smaling, Joost Wolf, and Henk van Reuler. A system for quantitative evaluation of the fertility of tropical soils (quefts). *Geoderma*, 46(4):299–318, 1990.

[36] James W Jones, Gerrit Hoogenboom, Cheryl H Porter, Ken J Boote, William D Batchelor, LA Hunt, Paul W Wilkens, Upendra Singh, Arjan J Gijsman, and Joe T Ritchie. The dssat cropping system model. *European journal of agronomy*, 18(3-4):235–265, 2003.

[37] Sasan Karamizadeh, Shahidan M Abdullah, Azizah A Manaf, Mazdak Zamani, and Alireza Hooman. An overview of principal component analysis. *Journal of Signal and Information Processing*, 4(3B):173, 2013.

[38] BA Keating, J_P Evenson, and S Fukai. Environmental effects on growth and development of cassava (manihot esculenta crantz.) i. crop development. *Field Crops Research*, 5:271–281, 1982.

[39] BA Keating, J_P Evenson, and S Fukai. Environmental effects on growth and development of cassava (manihot esculenta crantz.) ii. crop growth rate and biomass yield. *Field Crops Research*, 5:283–292, 1982.

[40] BA Keating, JP Evenson, and S Fukai. Environmental effects on growth and development of cassava (manihot esculenta crantz.) iii. assimilate distribution and storage organ yield. *Field Crops Research*, 5:293–303, 1982.

[41] Saeed Khaki and Lizhi Wang. Crop yield prediction using deep neural networks. *Frontiers in plant science*, 10:621, 2019.

[42] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.

[43] Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39:261–283, 2013.

[44] Akshay Krishnan, Shashank Swarna, et al. Robotics, iot, and ai in the automation of agricultural industry: a review. In *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*, pages 1–6. IEEE, 2020.

[45] LB Krithika et al. Survey on the applications of blockchain in agriculture. *Agriculture*, 12(9):1–38, 2022.

[46] Marek Kulbacki, Jakub Segen, Wojciech Knieć, Ryszard Klempous, Konrad Kluwak, Jan Nikodem, Julita Kulbacka, and Andrea Serester. Survey of drones for agriculture automation from planting to harvest. In *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, pages 000353–000358. IEEE, 2018.

[47] Rakesh Kumar, MP Singh, Prabhat Kumar, and JP Singh. Crop selection method to maximize crop yield rate using machine learning technique. In *2015 international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM)*, pages 138–145. IEEE, 2015.

[48] Takio Kurita. Principal component analysis (pca). *Computer Vision: A Reference Guide*, pages 1–4, 2019.

[49] Krithika LB. Survey on the applications of blockchain in agriculture. *Agriculture*, 12(9):1333, 2022.

[50] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[51] Shubo Li, Yanyan Cui, Yuan Zhou, Zhiting Luo, Jidong Liu, and Mouming Zhao. The industrial applications of cassava: current status, opportunities and prospects. *Journal of the Science of Food and Agriculture*, 97(8):2282–2290, 2017.

[52] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[53] Jun Liu. Design and implementation of an intelligent environmental-control system: Perception, network, and application with fused data collected from multiple sensors in a greenhouse at jiangsu, china. *International Journal of Distributed Sensor Networks*, 12(7):5056460, 2016.

[54] Yanli Liu, Yourong Wang, and Jian Zhang. New machine learning algorithm: Random forest. In *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3*, pages 246–252. Springer, 2012.

[55] Philipp Lottes, Raghav Khanna, Johannes Pfeifer, Roland Siegwart, and Cyrill Stachniss. Uav-based crop and weed classification for smart farming. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3024–3031. IEEE, 2017.

[56] Fabian Löw, U Michel, Stefan Dech, and Christopher Conrad. Impact of feature selection on the accuracy and spatial uncertainty of per-field crop classification using support vector machines. *ISPRS journal of photogrammetry and remote sensing*, 85:102–119, 2013.

[57] James Lowenberg-DeBoer, Iona Yuelu Huang, Vasileios Grigoriadis, and Simon Blackmore. Economics of robots and automation in field crop production. *Precision Agriculture*, 21:278–299, 2020.

[58] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993.

[59] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9(1):381–386, 2020.

[60] Mohd Saiful Azimi Mahmud, Mohamad Shukri Zainal Abidin, Abioye Abiodun Emmanuel, and Hameedah Sahib Hasan. Robotics and automation in agriculture: present and future applications. *Applications of Modelling and Simulation*, 4:130–140, 2020.

[61] Elizabeth H Mahood, Lars H Kruse, and Gaurav D Moghe. Machine learning: a powerful tool for gene function prediction in plants. *Applications in Plant Sciences*, 8(7):e11376, 2020.

[62] Aishwarya Himanshu Manek and Parikshit Kishor Singh. Comparative study of neural network architectures for rainfall prediction. In *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, pages 171–174. IEEE, 2016.

[63] Luis A Manrique. Growth and yield performance of cassava grown at three elevations in hawaii. *Communications in soil science and plant analysis*, 23(1-2):129–141, 1992.

[64] RB Matthews and LA Hunt. Gumcas: a model describing the growth of cassava (manihot esculenta l. crantz). *Field Crops Research*, 36(1):69–84, 1994.

[65] Roger Matthews. Criminal statistics. 1997.

[66] Tin Maung Aye and Reinhardt H Howeler. Integrated crop management for cassava cultivation in asia. 2017.

[67] Diana Elena Micle, Florina Deiac, Alexandru Olar, Raul Florentin Drența, Cristian Florean, Ionuț Grigore Coman, and Felix Horațiu Arion. Research on innovative business plan. smart cattle farming using artificial intelligent robotic process automation. *Agriculture*, 11(5):430, 2021.

[68] Velayudhan Santhakumari Santosh Mithra, AR Seena Radhakrishnan, and Divya K Lekshmanan. Computer simulation of cassava growth. In *Cassava*. IntechOpen, 2018.

[69] VS Santhosh Mithra, J Sreekumar, and CS Ravindran. Computer simulation of cassava growth: a tool for realizing the potential yield. *Archives of Agronomy and Soil Science*, 59(4):603–623, 2013.

[70] Patricia Moreno-Cadena, Gerrit Hoogenboom, James H Cock, Julian Ramirez-Villegas, Pieter Pypers, Christine Kreye, Meklit Tariku, Kodjovi Senam Ezui, Luis Augusto Becerra Lopez-Lavalle, and Senthold Asseng. Modeling growth, development and yield of cassava: A review. *Field Crops Research*, 267:108140, 2021.

[71] Pete Mutschler. Threats to precision agriculture. 2018.

[72] Anamai Na-udom and Jaratsri Rungrattanaubol. Data mining techniques for predicting cassava yields in lower northern thailand. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(2-4):95–99, 2017.

[73] Vladimir Nasteski. An overview of the supervised machine learning methods. *Horizons. b*, 4:51–62, 2017.

[74] Emerson Navarro, Nuno Costa, and António Pereira. A systematic review of iot solutions for smart farming. *Sensors*, 20(15):4231, 2020.

[75] Felix I Nweke. The cassava transformation in africa. In *Proceedings of the validation forum on the global cassava development strategy*, volume 2, pages 15–61, 2005.

[76] Mićo Oljača, Kosta Gligorević, Miloš Pajić, Ivan Zlatanović, Milan Dražić, Dušan Radojičić, Dragan Marković, Vojislav Simonović, Ivana Marković, Milorad okić, et al. Application of drone in agriculture. In *18th Scientific conference Current problems and tendecies in agricultural engineering ", Proceedings*, pages 89–101. University of Belgrade, Faculty of Agriculture, The Institute for ..., 2016.

[77] Pariwat Ongsulee. Artificial intelligence, machine learning and deep learning. In *2017 15th international conference on ICT and knowledge engineering (ICT&KE)*, pages 1–6. IEEE, 2017.

[78] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. A review on random forest: An ensemble classifier. In *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, pages 758–763. Springer, 2019.

[79] Heather Pasley, Hamish Brown, Dean Holzworth, Jeremy Whish, Lindsay Bell, and Neil Huth. How to build a crop model. a review. *Agronomy for Sustainable Development*, 43(1):2, 2023.

[80] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.

[81] Phanupong Phoncharoen, Poramate Banterng, Nimitr Vorasoot, Sanun Jogloy, Piyada Theerakulpisut, and Gerrit Hoogenboom. Growth rates and yields of cassava at different planting dates in a tropical savanna climate. *Scientia Agricola*, 76:376–388, 2019.

[82] Meghna Raj, Shashank Gupta, Vinay Chamola, Anubhav Elhence, Tanya Garg, Mohammed Atiquzzaman, and Dusit Niyato. A survey on the role of internet of things for adopting and promoting agriculture 4.0. *Journal of Network and Computer Applications*, 187:103107, 2021.

[83] J Ranjani, VKG Kalaiselvi, A Sheela, G Janaki, et al. Crop yield prediction using machine learning algorithm. In *2021 4th international conference on computing and communications technologies (ICCCT)*, pages 611–616. IEEE, 2021.

[84] Rapeepat Ratasuk, Nitin Mangalvedhe, Yanji Zhang, Michel Robert, and Jussi-Pekka Koskinen. Overview of narrowband iot in lte rel-13. In *2016 IEEE conference on standards for communications and networking (CSCN)*, pages 1–7. IEEE, 2016.

[85] Kazeem O Rauff, Rasaq Bello, et al. A review of crop growth simulation models as tools for agricultural meteorology. *Agricultural Sciences*, 6(09):1098, 2015.

[86] Abderahman Rejeb, Alireza Abdollahi, Karim Rejeb, and Horst Treiblmaier. Drones in agriculture: A review and bibliometric analysis. *Computers and Electronics in Agriculture*, 198:107017, 2022.

[87] Michael J Roberts, Noah O Braun, Thomas R Sinclair, David B Lobell, and Wolfram Schlenker. Comparing and combining process-based crop models and statistical models with some implications for climate change. *Environmental Research Letters*, 12(9):095010, 2017.

[88] Amine Roukh, Fabrice Nolack Fote, Sidi Ahmed Mahmoudi, and Saïd Mahmoudi. Big data processing architecture for smart farming. *Procedia Computer Science*, 177:78–85, 2020.

[89] Peter Norvig Russell. Artificial intelligence: a modern approach by stuart. *Russell and Peter Norvig contributing writers, Ernest Davis...[et al.]*, 2010.

[90] Minwoo Ryu, Jaeseok Yun, Ting Miao, Il-Yeup Ahn, Sung-Chan Choi, and Jaeho Kim. Design and implementation of a connected farm for smart farming system. In *2015 IEEE SENSORS*, pages 1–4. IEEE, 2015.

[91] H Sabireen and VJIE Neelanarayanan. A review on fog computing: Architecture, fog with iot, algorithms and research challenges. *Ict Express*, 7(2):162–176, 2021.

[92] Max v Schönfeld, Reinhard Heil, and Laura Bittner. Big data on a farm—smart farming. *Big data in context*, pages 109–120, 2018.

[93] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[94] Wenjiao Shi, Fulu Tao, and Zhao Zhang. A review on statistical models for identifying climate contributions to crop yields. *Journal of geographical sciences*, 23:567–576, 2013.

[95] Pramila P Shinde and Seema Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[96] Shraddha Shukla and S Naganna. A review on k-means data clustering approach. *International Journal of Information & Computation Technology*, 4(17):1847–1860, 2014.

[97] M Monica Subashini, Sreethul Das, Soumil Heble, Utkarsh Raj, and R Karthik. Internet of things based wireless plant sensor for smart farming. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(2):456–468, 2018.

[98] Michael E Sykuta. Big data in agriculture: property rights, privacy and competition in ag data services. *International Food and Agribusiness Management Review*, 19(1030-2016-83141):57–74, 2016.

[99] Sebastian Terence and Geethanjali Purushothaman. Systematic review of internet of things in smart farming. *Transactions on Emerging Telecommunications Technologies*, 31(6):e3958, 2020.

[100] Susan Thapa, Gaetano Piras, Sudesh Thapa, Pravesh Rimal, Aradhya Thapa, and Kushal Adhikari. Blockchain-based secured traceability system for the agriculture supply chain of ginger in nepal: A case study. *Arch. Agric. Environ. Sci*, 6:391–396, 2021.

[101] Feng Tian. An agri-food supply chain traceability system for china based on rfid & blockchain technology. In *2016 13th international conference on service systems and service management (ICSSSM)*, pages 1–6. IEEE, 2016.

[102] Luana F Tironi, Nereu A Streck, Paulo I Gubiani, Rômulo P Benedetti, and Charles P de O Freitas. Simanihot: A process-based model for simulating growth, development and productivity of cassava. *Engenharia Agrícola*, 37:471–483, 2017.

[103] Paolo Tripicchio, Massimo Satler, Giacomo Dabisias, Emanuele Ruffaldi, and Carlo Alberto Avizzano. Towards smart farming and sustainable agriculture with drones. In *2015 international conference on intelligent environments*, pages 140–143. IEEE, 2015.

[104] Antonis Tzounis, Nikolaos Katsoulas, Thomas Bartzanas, and Constantinos Kittas. Internet of things in agriculture, recent advances and future challenges. *Biosystems engineering*, 164:31–48, 2017.

[105] Frank Veroustraete. The rise of the drones in agriculture. *EC agriculture*, 2(2):325–327, 2015.

[106] JIŘÍ VOHRADSKY. Neural network model of gene expression. *the FASEB journal*, 15(3):846–854, 2001.

[107] Tran Thien Vu and Hue Hoang Hong Trinh. Blockchain technology for sustainable supply chains of agri-food in vietnam: a swot analysis. *VNUHCM Journal of Economics, Business and Law*, 5(1):1278–1289, 2021.

[108] Sun-Chong Wang and Sun-Chong Wang. Artificial neural network. *Interdisciplinary computing in java programming*, pages 81–100, 2003.

[109] Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jeroen Bogaardt. Big data in smart farming–a review. *Agricultural systems*, 153:69–80, 2017.

[110] Jinyuan Xu, Baoxing Gu, and Guangzhao Tian. Review of agricultural iot technology. *Artificial Intelligence in Agriculture*, 2022.

[111] Jun Xu, Junmei Yao, Lu Wang, Zhong Ming, Kaishun Wu, and Lei Chen. Narrowband internet of things: Evolutions, technologies, and open issues. *IEEE Internet of Things Journal*, 5(3):1449–1462, 2017.