Chair for Data Processing
TUM Department of Electrical and Computer Engineering
Technical University of Munich

Master's Thesis

Integration of machine learning and predictive analytics in agriculture to optimize plant disease detection and treatment in Egypt

Ahmed Tageldin

April 1, 2020

Examiner
Prof. Dr.-Ing. Klaus Diepold
Supervisor
Dr. Hassan Mostafa, Cairo University

Contents

1 Introduction

1.1 Motivation

The world's population is set to reach 9.2 billion people by the end of 2050, according to the UN Food and Agricultural Organization (FAO). To avoid a global food crisis, planting more and breeding more animals for food will not be enough. It will also be necessary to improve the efficiency within the current farming methods used worldwide.

Due to the small sizes of fields, farmers were previously able to treat fields individually, before the vast mechanization in agriculture. However, with the constant enlargement of fields and increased use of mechanization, as well as the complexity of future landscapes or difficulty of topographies, it will become more and more difficult to take consider within-field variability without a radical technological development.

In modern-day agriculture more and more farmers have access to sensors and mechanization that are constantly developed and conformed, enabling high automation and precision farming. These decision-support tools intend to be directed towards more effective and efficient design and delivery of monocultures, cultural practices, the use of insecticides (killing insects and other natural enemies), the introduction of pests in the environment, and the disruption of the natural equilibrium. Challenges faced daily can be reduced with the exposure to these technologies.

1.2 Research question

This research intends to present the impact of machine learning (ML) in precision agriculture (PA) to increase productivity and maximize the yields of crops by detecting diseases in plants before they spread irreversibly. By applying ML to sensor data, management systems are turning into artificial intelligence enabled programs providing recommendations and insights on the spot.

The scope of the project is targeting the eggs of the Egyptian cotton leafworm, as it is too resistant to be controlled by the common chemical insecticides and need to be targeted directly. A similiar behavior can be observed with the potato blight which could be used for analysis.

The goal is to construct a complete system for pest control which includes sensors supporting a machine learning software and a novel physical method for pest control.

Temperature, humidity sensors, and monitoring cameras will gather information and data for the detection of pests from inside a greenhouse and the system will update the farmers regularly.

Following questions will be answered during this research project:

- What is the precision for deploying the sensors to receive an accurate overview of the state of the field over the entire space?
- How frequent should the data be provided to receive an accurate overview of the state of the field over the entire time?
- Which machine learning algorithms are best suited for this application and can reach the best accuracy (focus will lie on Decision Tree Algorithms and conventional Regression methods)?
- Which additional features are required to increase accuracy and how can they be achieved (e.g. pH of the soil)?

Finally, testing will be conducted on a small scale (greenhouse) to ensure its durability and the existence of any drawbacks for future progress in this field.

1.3 Goals of the Research

This paper intends to present the impact of machine learning (ML) in precision agriculture (PA). PA intends to increase productivity and maximize the yields of crops. The entire crop cycle can be optimized through the administration of the correct amount of inputs (water, fertilizers, pesticides or fungicides) at the precise time and place, or by detecting diseases in plants before they spread irreversibly. By applying ML to sensor data, management systems are turning into artificial intelligence-enabled programs providing recommendations and insights on the spot.

The general scope of the project lies specifically in targeting the eggs of the Egyptian cotton leafworm (scientifically known as Spodoptera littoralis). This harmful pest is very resistant to be controlled by common chemical insecticides and need to be targeted directly. /cite Smart greenhouses and greenhouse automation system to denote the implementation of technology in traditional methods. The benefits of this application are multifold including improving the productivity of the farm. For effective pest control, smart greenhouse designs should include sensors for pest detection and systems for pest control and eradication. For effective pest control, smart greenhouse designs will include sensors for pest detection and system for pest control and eradication. The goal is to construct a complete system for pest control which includes sensors supporting a machine learning software and a novel physical method for pest control (e.g. magnetic fields).

1.4 Structure of the Research

Temperature and humidity sensors in addition to monitoring cameras will be the sensation system for the detection of pests inside the greenhouse. These sensors will provide information and data which will be fed into a machine learning software that will decide on operating the pest control device. The system will be able to send a warning signal to the operator with updating on the status of the greenhouse regularly. The pest control will be accomplished via a new clean and environment-friendly method, namely low frequency pulsed magnetic fields. This method has been tested on the targeted insects in the lab leading to promising preliminary results.

Before the implementation of the automatic system and validation of its operation, multiple experimental results will be secured. The effectiveness and validation of the sensation system for pest detection via the machine learning technique with sufficient accuracy will be conducted. On the other hand, the controlling power of the magnetic field on the targeted insects should be studied in the laboratory using different parameters with statistically accepted repetition of the results. Finally, testing the complete system on a small scale to ensure its durability and testing the existence of any drawbacks to solve.

Machine Learning problems can be broken down into two major parts; the datasets and the algorithms. The dataset will be provided by the sensors distributed in the greenhouse (temperature, humidity, and image). Thus, the sensors IoT-module will be set up.

The algorithm for the evaluation and classification of the data into safe or unsafe will be tested, while the focus will lie on regression analysis and decision tree analysis with linear and non-linear kernels. The output of this software will be connected to the automatic system used to control the pests. Data will be gathered concerning the relation between temperature and humidity and the insect population for system training. Additionally, new data will be gathered from the place of application for data verification.

2 Precision agriculture

This chapter will provide a general understanding of the topic of precision agriculture before laying out some of the previous work done in this area and the applicability to Egypt.

Over 25 percent of the Egyptian population are employed in the agriculture sector [9]. Actions taken to reduce losses, such as improved technologies, postharvest handling, processing activities, and better marketing channels, can boost growth across this crucial sector for the country. The pressure on this sector will keep increasing with the continuing growth of the population.

2.1 General Approach

The agriculture sector faces multiple challenges linked to diseases and pests as well as improper soil treatment and water systems and many others. Research is being conducted to address these issues. Artificial intelligence with its vast learning capabilities has become a major tool in the race to solve the agriculture related problems. [3]

Precision agriculture (PA), also called Precision farming, is defined as the scientific field that uses data intense approaches to especially drive agricultural productivity with regard to the environmental impact of the chemicals used [18]. It is the application of technologies and principles managing multiple aspects of agricultural production to improve crop performance and quality.[23] Reducing the number of chemicals used in plant protection products, will ultimately also reduce the levels of residuals found on our food [8].

The basic principles of PA can be described as a summary of good agricultural practices requiring the following[8]:

- Correct information (soil, previous crops, and treatment...)
- Correct observation
- Correct analysis
- Correct chemical/biological compound
- Correct place
- Correct time
- Correct dose
- Correct genotype
- Correct (climatic) conditions

- Correct equipment

This work will focus on the first seven points regarding the data gathering, analysis and the decision making process. To ensure the application of the correct dose of fertilizer at the correct moment the correlation with the soil and crop condition needs to be determined [8].

Difficulties arise, recording all steps and treatments carried out during the production of the crop. That is where automation and robotics can support in the decision-making process of the farmer to follow the correct treatment plans and to document all necessary data. [8]

2.2 Machine Learning

A lot of techniques were used to understand the rules and relationships from diverse data sets, to simplify the process of acquiring knowledge from empirical data. These techniques perform well on more or less artificial test data sets, the main goal is to make sense of real-world data [19].

Machine learning (ML) offers an alternative to the conventional engineering flow when problem are too complex to develop a solution with guarantees. On one hand the approach has the disadvantages of producing black-box-solutions that are not interpretable so they are only applicable to a limited set of problems. Following criteria should be fulfilled for problems for which machine learning methods could be useful [5] [25]:

- problem involves a function that maps defined inputs and outputs
- data exist or can be obtained containing pairs of inputs and corresponding outputs
- problem provides clear goals and metrics
- problem does not involve long chains of logic or reasoning that depend on diverse background knowledge or common sense
- problem does not need detailed explanations of why decisions were made
- problem has a tolerance for error and does not require precise or optimal solutions
- the function learned does not change over time

It is the application of artificial intelligence (AI) that provides the ability to automatically learn and improve from training sets without explicitly programmed instructions. The job of the modeling algorithm is to find the most applicable mapping function from input variables to output variables and aid in the discovery of rules and patterns in the data sets [19]. This paper reviews what ML can do in the agricultural sector, specifically in Egypt with the objective of developing a disease detection system that is robust and easy to adapt to different applications and crops, while following the criteria above [22].

2.2.1 Methods of Machine learning

To apply any sort of machine learning methods a dataset needs to be retrieved where each row of data represents an observation about something in the world. When working with data it is often not possible to have access to all possible observations. This could be due to the fact that it may difficult or expensive to make more observations. It may also be challenging to gather all observations if they are expected to be made in the future.

For all models, the dataset is divided into a train and a test set, both consisting of the features of analysis and the KPI. The train set is used to fit the model and define the relationship between the input features and the KPI, whereas the test set is used to measure how accurate the model is predicting the output given the test features.

Different types of algorithms and models can help achieve different goals, while in their core they are all ways of figuring out what drives the changes in the Key performance indicator (KPI) of the application. We distinguish between classification and regression problems [22].

Classification is about predicting a label. It is the method of approximating a mapping function from input to discrete output. The output is often called label or category. The function predicts the class for the given features or observations. For example, an email can be classified into one of two classes: “spam“ and “not spam“ [25].

Regression is about predicting a quantity. It is the method of approximating a mapping function from input to a continuous output, such as an integer or floating point value. These are often quantities, such as amounts and sizes. For example, an object may be predicted to cost a specific amount, given the circumstances (features/ observations) [25].

Table 2.1 – Machine learning methods

Method	Type	Description	Application
Logistic regression	Classification	Assumption of existing logistic relationship between KPI and features	Detection of Spam emails
Support vector machine (SVM)	Classification	Segregation of data points using non linear hyperplanes	Image classification and pattern detection
Linear regression	Regression	Finding linear relation between input and output while minimizing mean squared error	Time series problems like cost of product
Random Forest	Regression	Decision tree analysis	Grades of student based on all other grades and behavior of student
XGBoost	Regression & Classification	Different models are combined reducing the sum of errors of all models (Decision tree approach)	Assessment of vehicle driving behavior and risk predictions

Table 2.1 shows an overview of some commonly used ML algorithms and methods. The following subsections will describe the algorithms and their mathematical and statistical background further.

Logistic Regression

Logistic Regression is a classification method used when the dependent variable is categorical, like to predict whether an email is a spam, or whether a tumor is malignant or not.

The output of the model is the estimated probability. This is used to determine how confident the model is regarding its prediction given any input.

Logistic regression is based on the basic assumption of an existing logistic relationship between the dependent variable (KPI) and the independent features or observations. By fitting data to a logit function, it predicts the probability of occurrence of the event. The cost function of the Logistics regression is different from the cost function of the standard linear regression [10].

To explain this through an example, a person is given a task to solve with only two outcome scenarios (solved/not solved). The same person is then given a wide range of tasks in an attempt to understand which subjects they are good at. If they are given trigonometry based tenth grade problem, for example, they are 70% likely to solve it, while being 30% likely to solve a history question. In the Logistic Regression, the log odds of the outcome are modeled as a linear combination of the predictor variables.

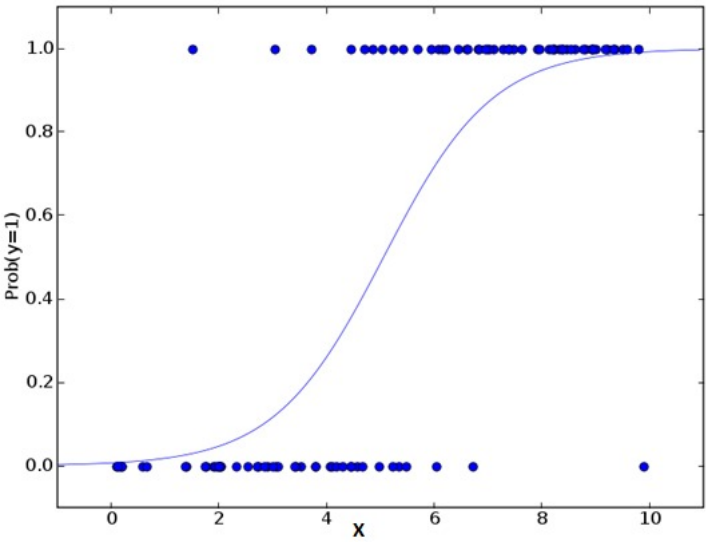


Figure 2.1 – Example of data points distributed according to their likelihood using logistic regression

This form of regression chooses parameters that maximize the likelihood of observing the sample values rather than that minimizing the sum of squared errors (like in ordinary regression). The log function is used as it mathematically replicates the step functions as can be seen in Figure 2.1.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised ML algorithm, mostly used in classification problems. Each data point is plotted in an n-dimensional space (n-features) with the value of each feature being the value of the particular coordinate. The classification is performed by finding the hyperplane/line that differentiates the categories and segregates the data points [30].

The hyperplane is selected, that segregates the two classes better while maximizing the distances between the nearest data point (either class). This distance is called the Margin. The hyperplane with a higher margin is selected due to its robustness. If the selected hyperplane has a low margin, there will be a high chance of miss-classification [30].

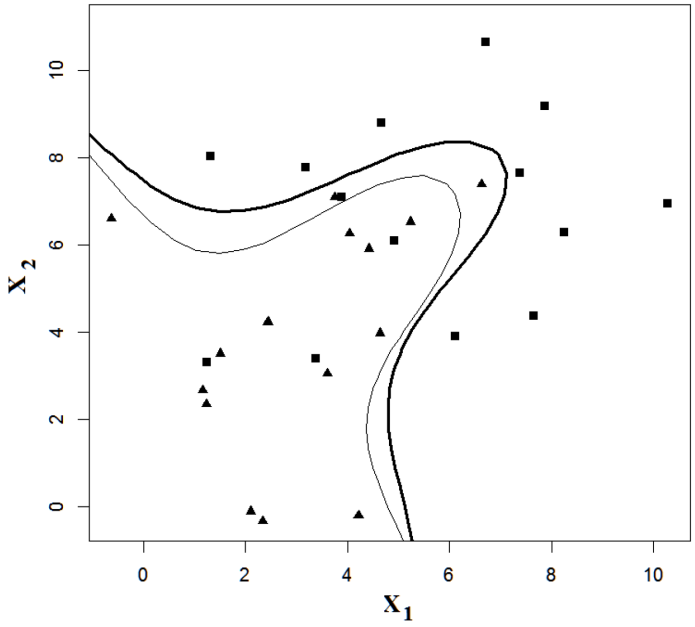


Figure 2.2 – Example of non-linear hyperplanes segregating data points in 2-dimensional space [30]

Figure 2.2 shows a distribution of data points in a 2-dimensional space with a non-linear hyperplane segregating the values. It solves the problems by introducing additional features to convert not separable problems to separable problems using so called kernels.

SVM also has the ability to ignore outliers and find the hyper-plane that has a maximum margin. It is mostly useful in non-linear separation problem using complex data transformations, then finding the function to separate the data based on the labels or outputs defined [30].

Linear Regression

The Linear Regression predicts values of a KPI as a linear combination of the independent observations/features. The linear coefficients are determined so as to optimize the error function (mean squared error) of the predictions. In summary, a set of independent features is used to predict one dependent KPI [1] [6].

For this and other ML-algorithms, the dataset is split into a training and a test set. During the training step, the linear coefficients of the model are calculated and used to predict the values of any unknown pattern, not provided in the training set [26]. The patterns of features in the test set are then used to predict the corresponding KPIs, with the error for the evaluation of the model being calculated as the mean squared error between the predicted values of the test set and the actual values.

The linear regression works with the basic assumption of the existence of a linear relationship between the KPI and the independent features measured. When applied with multiple independent variables, the linear regression is also referred to as multiple linear regression.

Random Forest

Random Forest (RF) is a ML algorithm making predictions based on multiple decision trees. To classify a new object based on attributes, each tree gives a classification and a rank for that class. The classification with the highest rank (over all the trees in the forest) is chosen and in case of regression, the average of outputs by different trees is used for predictions.

In Random Forest, each tree is set up as follows: N cases in the training set are defined. Samples of these cases are taken at random but with replacement. They will be the training set for growing the decision tree.

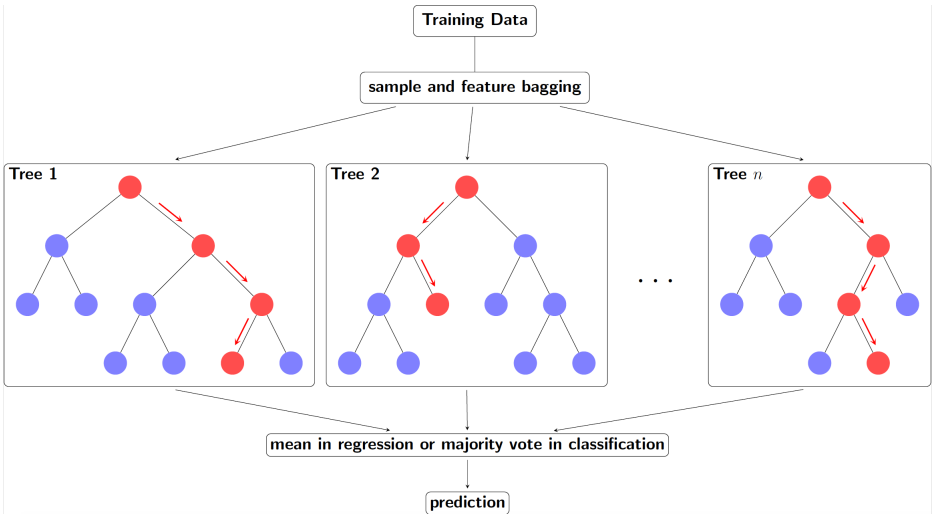


Figure 2.3 – Example of RF decision trees

Figure 2.3 shows an example of a RF. A number m smaller than M (Number of input variables,) is specified such that at each node, m variables are selected at random. The best split on these m is used to split the node, while the value of m is held constant. Each decision tree is grown to the largest extent possible without pruning. New data is being predicted by aggregating the predictions of the trees (average for regressions) [1].

eXtreme Gradient Boosting (XGBoost)

XGBoost is an implementation of gradient boosted decision trees (Similar to random forest in subsection 2.2.1) designed for more speed and performance [24].

The algorithm provides a system for use in different computing environments such as:

Parallelization of the tree constructions using the entire CPU cores during the training phase.

Distributed Computing to train large models using a cluster of machines.

Out-of-Core Computing for large datasets that would not fit into the memory.

Cache Optimization of data structures and algorithms making the most of the hardware.

XGBoost is mostly used for execution speed and model performance. It is an approach where new and different models are created that can predict the errors of the other models to make the final prediction. It uses a gradient descent algorithm to reduce the loss while constantly adding new models and it can be used for both regression and classification problems.

Among different Decision tree algorithms, boosting was considered to be one of the most important algorithms in ML over the last 20 years as it can turn an ensemble of weak classifiers into strong ones [13].

2.3 Previous work

Various papers describe different automation practices like Wireless Communications, IOT, ML, AI and Deep Learning. Nowadays, there is an urgent need to define the issues like the use of harmful pesticides, the effects of environment and others. In their paper Jha et al. describe the application of a range of machine learning algorithms to problems in agriculture [15], with different problem sets and outcomes.

In terms of disease detection, leaf wetness is one of the most important aspects involved in the development of fungal pathogens and other diseases. It affects the deposition of pollutants on the crops, making the measurement of leaf wetness an important indicator for disease detection in the field of agriculture [14].

However, leaf wetness was usually determined using empirical models, physical models, or statistical methods [6] [21]. The empirical and physical models present limitations as they are site-specific. Statistical methods expect a linear relationship between the KPI (the leaf wetness) and the measured features. Neural network and ML-technology for leaf wetness prediction were also used in some cases. The main advantage of these algorithms is the

construction of the regression surface without any assumption about the prediction model and its form [22].

Chtioui et al. used a generalized regression neural network (GRNN) as well as linear regression (LR), also referred to as multiple linear regression (MLR) in this case. Their applicability for leaf wetness prediction and forecasting several plant diseases was measured [6]. This paper is used to determine the optimal set of features for the different prediction models.

2.3.1 Features and datasets

To obtain a set of reliable data, continuous measurements of the necessary features are crucial [6]. In their paper, Chtioui et al. used data of spring wheat grown between 1993 and 1997 at the Agricultural Research Center of the North Dakota State University. Table 2.2 describes the meteorological features (temperature, relative humidity, wind speed, radiation and precipitation) used in the approach to measure leaf wetness to forecast diseases [6].

Table 2.2 – Meteorological features used for leaf wetness prediction [6]

Feature	Definition	Notation
Time	Regression	t
Temperature	Temperature on leaf	ttemp
First difference of temperature	Difference between temperature at time t and that 1 h earlier	dtemp
Relative humidity	Ratio of the quantity of vapor actually present to the greatest rh amount possible at the given temperature.	rh
First difference of relative humidity	Difference between relative humidity at time t and that 1 hour earlier	drh
Wind speed	Wind speed or turbulence	ws
First difference of wind speed	Difference between wind speed at time t and 1 h earlier	dws
Solar radiation	Total amount of solar radiation	sr
First difference of solar radiation	Difference between solar radiation at time t and that 1 hour earlier	dsr
Precipitation	Amount of precipitation	ppt
First difference of precipitation	Difference between the total amount of precipitation at time t and that 1 h one hour earlier:	lag1
Soil moisture index	/	lag72

To reduce the dataset size, the measurements of temperature, relative humidity and wind speed were summarized by the hour, while precipitation and solar radiation were summed up [6].

2.3.2 Results

The MLR and the GRNN were compared by Chtioui et al. for leaf wetness prediction. The GRNN is a neural network-based regression method taking into account the relevance of each measured feature, while the MLR, as described in Table2.1, is a standard statistical regression method.

The MLR resulted in an average absolute error of 13% at predicting leaf wetness for the training and 14.14% for the test set. In the meantime, the GRNN resulted in far better results of 4.91% and 8.94% respectively [6].

The features in Table 2.2 were assessed, based on their importance. The importance was determined by the effect they had on the mean absolute error of the prediction when removed from the calculation. Simulations were conducted showing that the six features calculated with the differencing-operation were decreased the error by 1.12% for the training and 0.23% for the test set.

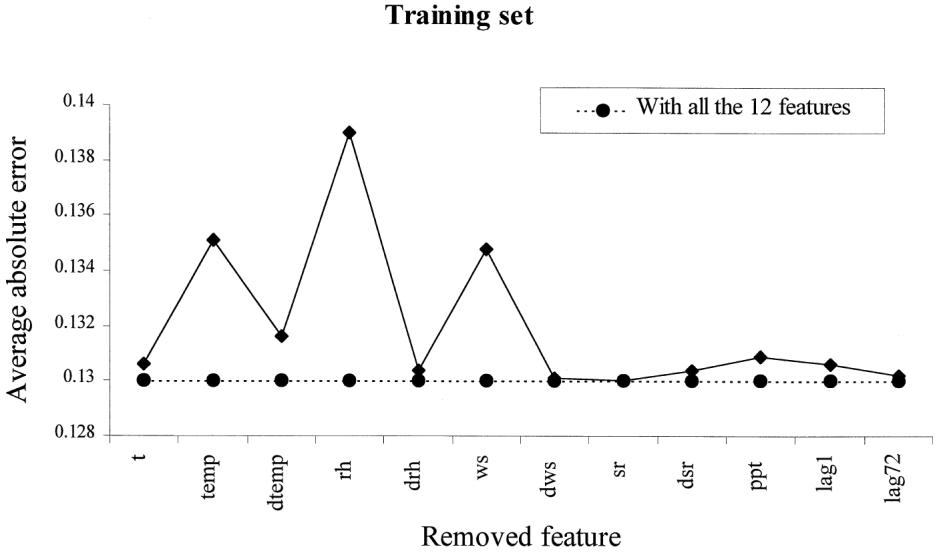


Figure 2.4 – Prediction accuracy obtained by LR. Results are summarized for the training set and for the removal of each individual feature [6]

Figure 2.4 shows the effect of each feature on the absolute error, when removed from the model in the simulation. Similar to the standard linear regression, the MLR works with the basic assumption of the existence of a linear relationship between the KPI or dependent variable and the independent features or observations [6].

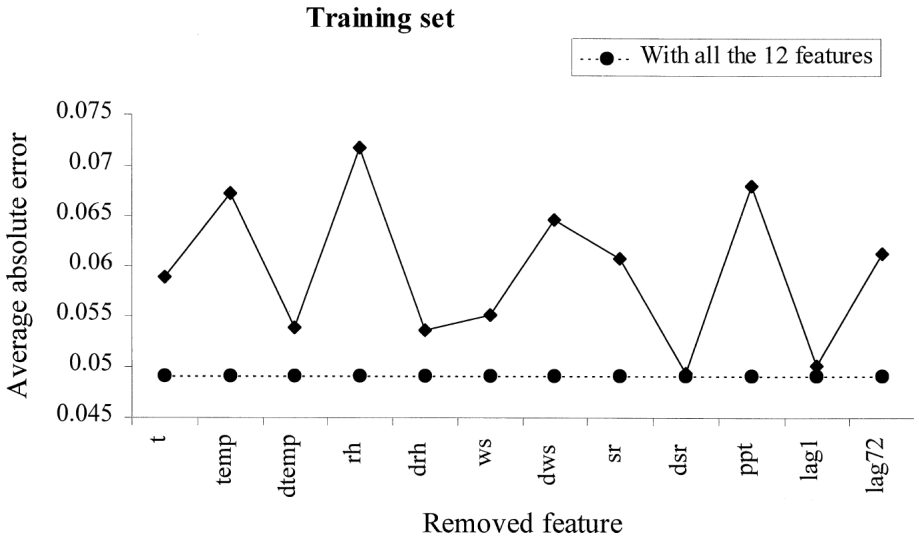


Figure 2.5 – Prediction accuracy of the GRNN for the training set when individual features were excluded [6]

On the other hand, the GRNN predicts the dependent variable (leaf wetness) without any assumption about the regression model and its form. The regression model is automatically generated with the information about the measured meteorological data [6]. This results in a different relation between the dependent and independent variables and thus a difference of effect or importance of each feature when removed from the simulation model as shown in Figure 2.5.

2.3.3 Conclusion

The results of Chtioui et al. (Figure 2.4 and 2.5) can be used to determine a realistic set of features for the analysis.

Due to its location at low latitude, the solar radiation in Egypt is relatively stable with a small variation where the daily range of solar radiation components are relatively small. during the year [20]. Therefore, these features will not be measured, as well as the soil moisture index.

The temperature, relative humidity as well as the precipitation and the wind speed should be measured for a reliable model in Egypt, alongside, their differential, calculated features. Table 2.3 shows a possible dataset for the analysis.

Table 2.3 – Proposed structure of database for precision agriculture analysis

Node	Date	Time	Temp	SM	RH	LF	LP	LW	DS
1	1.1.20	20:00	27.9 °C	62%	24.1%	x	x	x	x
1	1.1.20	21:00	27.3 °C	58%	24.3%	x	x	x	x
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

The network is proposed to transmit one new record for each individual node once per hour with the following features:

Temp: Temperature in °C
SM: Soil moisture in %
RH: Relative Humidity in %
LF: Days since last use of fertilizers
LP: Days since last time use of pesticides
LW: Hours since last watering
DS: Disease Severity.

The features LF, LP and LW will be entered manually into the front-end by the end user. They can also be generated automatically and integrated into the database if the farmer uses a schedule for watering, pesticides and fertilizers.

These features are expected to give a better understanding of the environment of the research as the measurements will be taken in a changing environment that adapts to waves of heat or dryness as well as the event of a plague. All external factors need to be accounted for in the data.

These main features can later be used in the analysis to derive further features for fine tuning like differential humidity or average humidities over past time windows. This will be further described in 4.

3 IoT System

This chapter will describe the implemented full IoT-system and each component contributing to the measurement and the analysis of the data.

Before applying any algorithms and prediction models, the required data needs to be obtained. For this purpose, an IoT system will be developed that connects to multiple sensors of each node to the main server where the collected data will be evaluated and decisions as well are warnings will be computed.

The micro-level data like temperature, humidity and soil moisture will be obtained at each node using the IoT, while other weather information (hours of sun, wind speed and rainfall) will be obtained from the closest weather station in the vicinity of the field [29] [2].

Similar Wireless sensor networks consisting of the battery-powered nodes and the sensors for monitoring agricultural/weather parameters have often been deployed and revealed weather-crop correlations that helped in generating a prediction model for several insects and diseases associated with carriers [29] [2].

3.1 Introduction

The proposed cloud-based IoT platform consists of three layers, the local node (perception layer), the gateway and the application layer [31] as shown in Figure 3.1. They work together as a controlled system transporting the signals transmitted at each node back to the main server where the analysis is performed. The signal is then cast back to each node with the specific command or action. The command can also be reported to the farmer responsible to inform him about upcoming and predicted events.

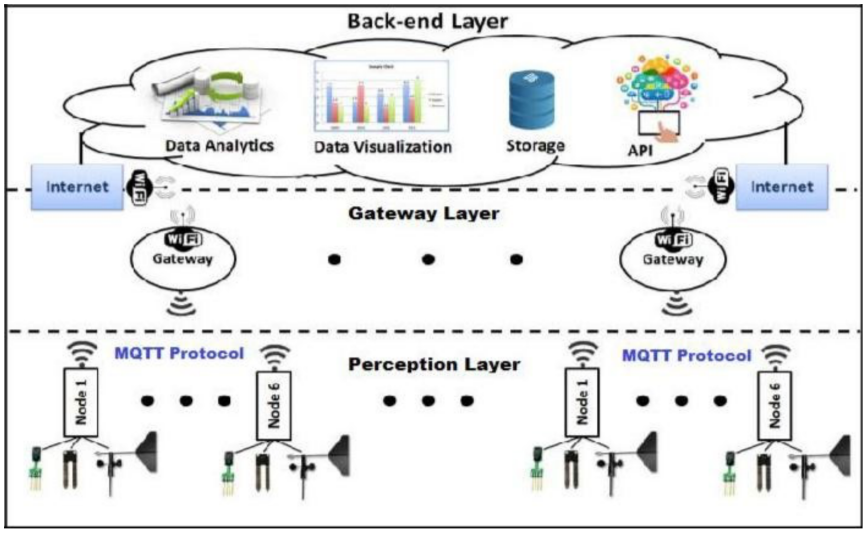


Figure 3.1 – Detailled description of the three layers of the IoT [2]

The implementation of these layers will be discussed in the following subsections.

3.1.1 Perception Layer

The first layer is the physical layer that contains the sensors of the nodes where the data is collected and transmitted to the next layer (gateway layer). It is made up of multiple nodes, each consisting of three components: sensors, microcontroller, and a communication module [17].

Sensors: Are used to measure the various environmental attributes needed for the given application. Air temperature sensors, air humidity sensors, and soil moisture sensors are used to measure yield conditions and environmental factors. The node then interfaces the collected data from the sensors with the microcontrollers.

Microcontroller: Is responsible for collecting the measured data of the sensors and can connect to the next layer (Gateway layer) using MQTT protocols.

Communication module and protocol: Wi-Fi modules and MQTT protocols are used to send the collected data to the gateway layer.

This protocol runs on TCP/IP connection and uses publish/subscribe communication pattern, sending data from sensors attached to NodeMCU continuously to gateway layer is defined as the publisher so that this node is defined as a publisher MQTT client. The MQTT client publishes the different data in a message-oriented where every message is published to a specific address called a topic.

To distinguish data, each sensor readings are published on a specific topic. The main distributor of the messages in the topics is a node that called an MQTT broker which is responsible for forwarding the messages between the sender and multiple receivers so that MQTT broker

forward the topic message to subscriber MQTT client which is the next layer, the gateway layer

3.1.2 Gateway Layer

The gateway layer serves as the bridge between the perception layer and the application layer. The different nodes deployed in the perception layer collect all sensors data and send it to the gateway layer. The gateway is implemented using Raspberry-Pi 3 microcontrollers (R-Pi 3). They provide the needed processing power and storage that make sure that all sensor data captured is forwarded to the database on the cloud server for analysis.

In this application, the R-Pi 3 serve as both the subscriber MQTT client as well as the broker. The R-Pi 3 is able to subscribe to the data from the same topics that the publisher MQTT client publishes in. In the gateway layer, the data is then analyzed after collection and stored on R-Pi 3. Depending on the proposed action determined by the ML algorithm, commands are then being sent from R-Pi 3 to the application layer.

3.1.3 Application Layer

Finally, the Application Layer visualizes the sensed data and the data analysis, thereby connecting the end-user with the dynamic application. The visual features can be used to code different attributes of data and change the commands [28]. The end user can see the effects and trends in his model without changing the original sensed data. the architecture is built via online database server to design the back-end layer with the previous specification as in Figure 3.2.

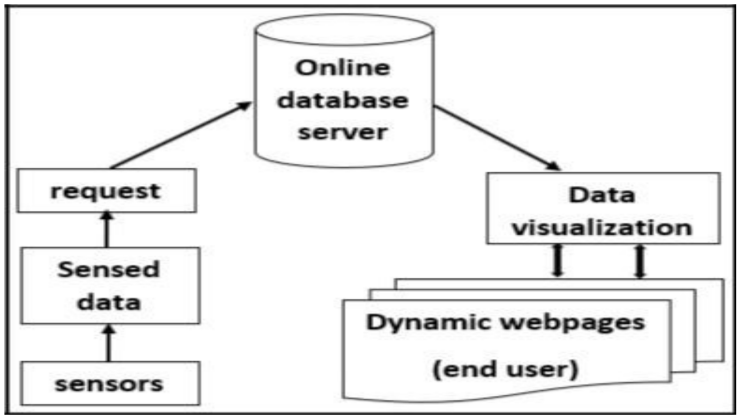


Figure 3.2 – Cloud server architecture [2]

The implemented back-end is based on an online MSSQL server through a free web hosting server which is receiving the data from the gateway by post method which receive the desired data with a specific key which has an agreement from transmitter and receiver to extract the

sensed data from the request and storing it to be accessed by the end user with a minimal downtime and without data corruption.

The website application consists of two separate web pages linked to the sensed data and the machine learning results gathered in one website to be viewed by the end user. The website is designed as a secure system by having a username and password for every client, linked to his data without accessing any other data.

3.2 Architecture

Section 3.1 described the theoretical structure of a general IoT System connecting the individual models to the front-end for the end user. This chapter describes the practical application of the IoT System in a given field.

3.2.1 Application

For this research a five hectare field in Cairo-Alexandria desert rode was used. Due to the shape of the field, the network will be set up using star topology (see Figure 3.3). This is one of the most common network setups. In this configuration, the nodes connect to a central hub or network device. The central device acts as a server whereas all peripheral devices serve as clients [16].

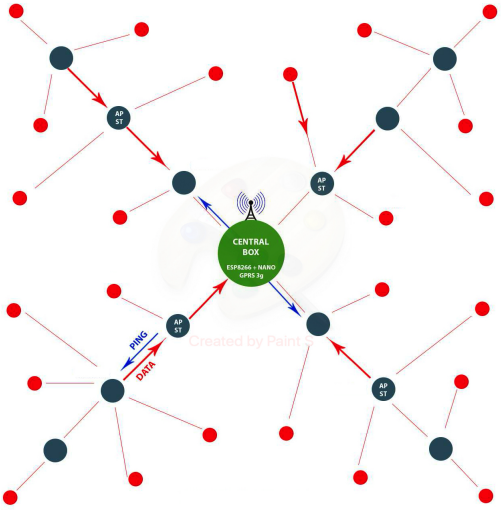


Figure 3.3 – Example of star topology (Red: Sensors; Blue: NodeMCUs; Green: Microcontroller)

The advantages of the star topology are the centralization of the network, the simplicity in adding another computer to the network as well as the security that if one node on the network fails, the rest of the network continues to function normally.

The range of the used sensors is sufficient to ensure the unproblematic transmission of information to the gateway without needing to use repeaters in the field. The locations of the

nodes are linked to the division of the field in sub-areas based on the available water pumps that will be controlled by the network.

The nodes will be battery powered, thereby limiting the lifetime to the battery life time. For this reason, the nodes will be transmitting information every hour to ensure a lifetime of 9-12 months.

3.3 Components

As described in Section 3.1, multiple devices will be used for sensing and analyzing the data. This chapter will describe in depth, which components were used in the implementation of the network and why.

3.3.1 Sensors

For the required measurements, several sensors will be used including temperature, humidity and soil moisture sensors. The sensors will be set up in and above the soil for accurate measurements. This chapter will focus on the used sensors.

It is important to keep in mind the possibility of oxidation and rusting of the sensors as a result of the exposure to water, therefore shielding is important. Several actions were taken to ensure adequate shielding to protect the network.

- Usage of sensors shielded polymer film that 'conforms' to the circuit board topology to protect electronic circuits from harsh environments that may contain high humidity, a range of airborne contaminants and varying temperatures
- Usage of sensors shielded with graphite as an antioxidant
- Usage of sensors immersed with gold as an antioxidant
- Limiting power consumption and thus slowing down the oxidation rate (Keeping the sensors in an idle state when not in use instead of leaving on). This will also increase battery lifetime

Temperature and Humidity Sensors

For the temperature and humidity sensors, three models were compared as in table 3.1. For simplicity, a device was used, that fulfilled both requirements in one.

Table 3.1 – Temperature and humidity sensors comparison

	DHT11	DHT22	HDC2080
Temperature Range	0 °C to 50 °C	-40°C to 80°C	–40°C to 125 °C
Humidity Range	20% to 90%	0% to 100%	0% to 100%
Accuracy	+/- 2 °C and +/- 5% RH	+/- 0.5 °C and +/- 2% RH	+/- 0.2 °C and +/- 2% RH
Sampling rate	1 Hz	0.5 Hz	1 Hz
Operating Voltage	3 V to 5V	3 V to 5V	1.62 V to 3.6 V
Operating current	2.5 mA	2.5mA	3 mA

Sensor DHT22 (See Figure 3.4) was chosen as a result of better pricing, the compatibility with the temperature needs as well as the availability of this sensor in Egypt.

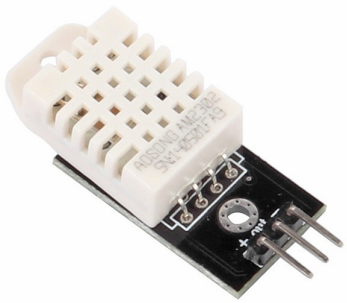


Figure 3.4 – DHT22 Digital Temperature and Humidity Sensor

Soil Moisture Sensors

As a soil moisture (precipitation) sensor, the YL-69 (see Figure 3.5) was used. The sensor consists of two pieces: the electronic board and the probe with two pads that detects the water content in the soil. This device is more common in commercial use because of its price and availability in Egypt. The operating voltage of this device is also compatible with the GPIOs of the used microcontroller at the node.

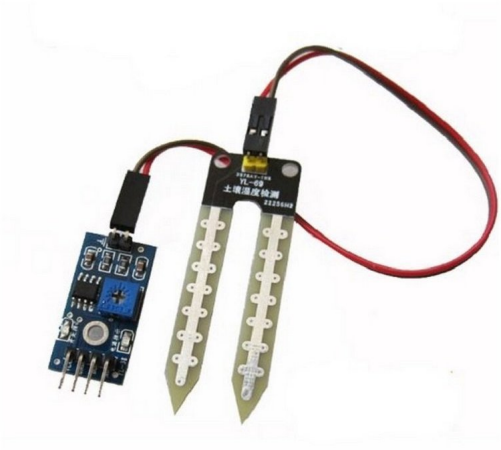


Figure 3.5 – Soil moisture sensor YL-69

3.3.2 Node Unit

As a node unit, the IoT platform NodeMCU - ESP8266 was used. It includes firmware running on the ESP8266, an open source IoT platform from Espressif Systems.

This Wifi module serves as both, a communication device and a microcontroller. This dual use technology proved more efficient in terms of power, availability and price.

Communication Module

The communication module is necessary for the communication between the sensors and the node module as well as the transmission to the gateway. For this purpose a Zigbee module XBee ZB Series S2C and Wifi module Esp 8266 were surveyed.

ZigBee technology is designed to transmit small amounts of data over a short distance, consuming very little power. However, the WiFi module uses a mesh networking standard, meaning each node in the network being connected to one another. Table 3.2 describes the technical differences between the modules.

Table 3.2 – Communication module comparison (Zigbee vs Wifi)

	Zigbee	Wi-fi
Network type	Wireless Personal Area Network (WPAN)	Wireless Local Area Network (WLAN)
Daily Power Consumption	0.39 watts	0.87 watts
Distance coverage	10 to 30 meters	30 to 300 meters
Data rate	250 Kbps	54 Mbps
Frequency Band	868/915 and 2.4GHz	2.4GHz and 5GHz

The specifications of both modules are sufficient for this use case but Zigbee modules are by far more expensive in Egypt. They are much more power efficient but they serve only as a communication module without any processing power, creating the need for an additional microcontroller.

Even though the Zigbee modules are easier in their handling because of simpler communication protocols, they also do not possess the necessary GPIOs provided in the used NodeMCU (see Figure 3.6).

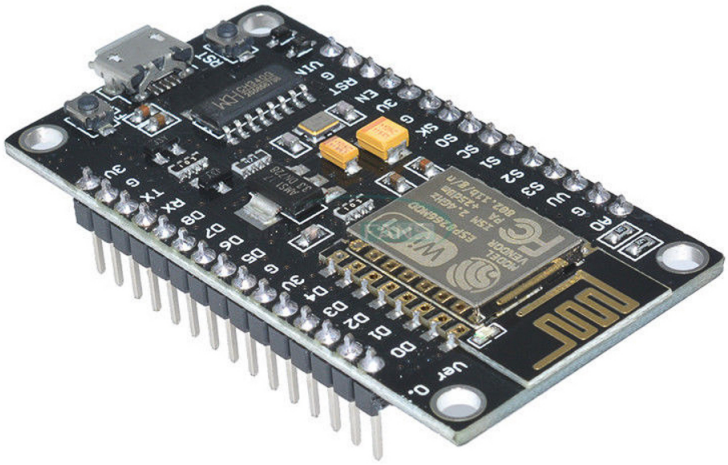


Figure 3.6 – NodeMCU - ESP8266

Microcontroller

As stated above the NodeMCU provides necessary processing power and the GPIOs to complete the application layer. The 12 GPIOs enable ad wide range of sensors and the possibility for future expansion of further sensors. This is especially important as more inputs will be used for the alarming and watering system. A separate microcontroller would also lead to additional costs and power consumption and increased complexity for end user.

3.3.3 Gateway

At the core of the network and the field itself stands the gateway layer represented by a processing unit and a GPRS module for transmitting a Wifi Signal.

GPRS

A GPRS module is used to establish communication between a mobile device and a GPRS system. The used module is the GPRS SIM900 (see Figure 3.7). It enables the NodeMCU to

transmit and receive the information and commands resulting from the ML algorithm. The module uses a sim card to start a hotspot for the Wifi modules. Possible complications could be network coverage in the field as most mobile providers don't have sufficient coverage in the area but were cleared with the SIM900 module and the Egyptian provider *Etisalat*.

Figure 3.7 – GPRS SIM900

Processing Unit

The used processing unit os the Raspberry Pi 3 (See Figure 3.8). It collects data using the same hotspot of the GPRS module and transmits to the web server. The build in Wifi sensor makes this product more attractive in terms of network connectivity.

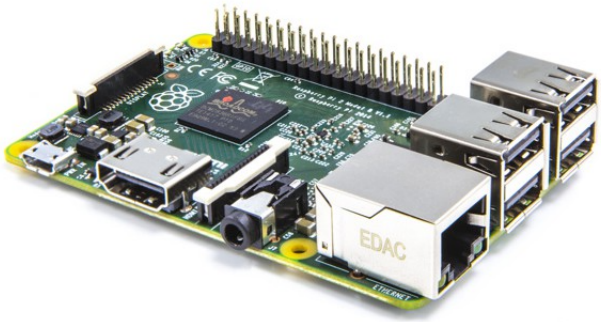


Figure 3.8 – Raspberry Pi 3 module

The Raspberry Pi is Python enabled and will later run the ML code to analyze all data and send complete reports back to the web-server.

The key here is to use the Raspberry Pi as a computing center for calculation and processing, while using an Arduino, as the executor for controls and collection. The Arduino has a built-in ADC, which is better for real time calculations. The Raspberry Pi is not fit for real-time operations, disabling the running of the ML algorithm as a continuous service as it's done on an MCU.

The Arduino also has many digital and analog I/O pins (GPIOs) that can be easily controlled. It will be used as a GPIO extender that can easily communicate with Arduino libraries/programs if chosen ver the current Raspberry Pi system.

3.3.4 Webserver Thinkspeak

The data collected will be stored in a database accessible through the webserver hosted on the main PC provided by the farm itself. It responds to client requests made over the Web (WWW). The software controls how a user accesses the database. The Web server process is an example of the client/server model. The Raspberry Pi will be able to access the database directly and fill in newly provided data, while the user can only access the database through the developed GUI.

The hosted database will be a Microsoft SQL Database. As a database server, it has the primary function of storing and retrieving data as requested by other software applications like the GUI and the Raspberry Pi 3.

3.4 Results of Field tests

The three sensors from subsection 3.3.1 were tested in the field and will be discussed in this section. he goal was to determine the optimal required amount of nodes and the distance between them to reduce the margin of error.

The sensors were tested multiple times in four different subareas in the field and the results for each subarea were averaged in table 3.3. It is clear that the temperature and the humidity measurements vary between different points in the ground. Therefore it is important to use multiple sensors in the same node but preferably on the same watering line to receive outputs representative of the entire subarea.

Table 3.3 – Sensor tests in field - Averaged results

Condition in soil	Relative Humidity	Temperature	Soil Moisture
Without water	26.2 % - 27.6 %	24 °C	764-813
With water and muddy soil	27.5 % - 28.1 %	24.1 °C - 24.2 °C	336-337
With water and sand soil	25.5 % - 25.9 %	23.2 °C	303-308
With water and grass soil	26.1 % - 27.5 %	22.2 °C - 22.5 °C	199-236

Measurements of the soil moisture depend on the watering time of the plant and are used as a threshold to determine the optimal watering schedule. The will be inside the ground and it is important to note that they will need to be put in a place representing to the whole area to avoid false results.

Each subarea is controlled by one water pump. As a result of the insights gathered is the field, five nodes will be put in each subarea and averaged out for analysis. They can not be regarded as five separate units as the subarea is controlled by a single pump and can not be divided any more.

4 Analysis

The previous chapters have described the goal of this research in chapter 1, they have given an overview of the previous work done in this field and the concluded insights to achieve the desired outcome in chapter 2 and the necessary components for the implementation of the full IoT system in chapter 3.

This chapter will now describe in detail the ML algorithms used and their outcomes and uncertainties over two obtained data sets. The insights from these tests can then be used to implement the backed layer of the IoT in the field.

As mentioned in section 2.2, Machine Learning is the acquisition of descriptions that make generalizations explicit and in a form that is straightforward in its interpretation. It embeds the knowledge in high- dimensional numerically parameterized unknown functions, thereby learning as a process of weight adjustments. It means learning from a training set of examples with known output patterns [19].

4.1 Datasets

A lot of research has previously been done on protected crops in greenhouses to control pests and diseases by biological means instead of the use of pesticides. These agrosystems are partly isolated from the environment and thereby highly controlled. This makes them good test areas for new and innovative methods in crop protection. These biophysical systems can be considered systems with inputs, outputs where the test variables work as control process loops [4].

This analysis will cover two separate data sets, one obtained within such a greenhouse and the other obtained in a plain field. As these analyses and models need a lot of data points which can take up to years to gather, the insights gained from the two available test sets will then be applied on the running system with a window for optimization.

One important problem domain here is the quality of the available data. Real data can be imperfect in the sense that it can be [19]:

- **incomplete** : missing values for some attributes and objects
- **irrelevant** : some attributes do not relate to the problem at hand but are mistakenly recorded
- **redundant** : involving unknown and unexpressed relations between attributes
- **noisy** : attributes can have measurement errors
- **erroneous** : transcribed incorrectly

ML algorithms need to be stable enough to deal with imperfect data and to discover correlations that are useful for the problem at hand [19].

4.1.1 Analysis of Potato Blight dataset

The first dataset used for this analysis was collected by Dr. Mohamed Fahim from the department of Plant Pathology at Cairo University. The set contains data of weather conditions within potato areas that were collected during four consecutive seasons, i.e. 2002/2003, 2003/2004, 2004/2005 and 2005/2006. The weather data were recorded manually in Badrashin region. The measured disease was the potato blight (see Figure 4.1)

Figure 4.1 – Potato blight on a leaf

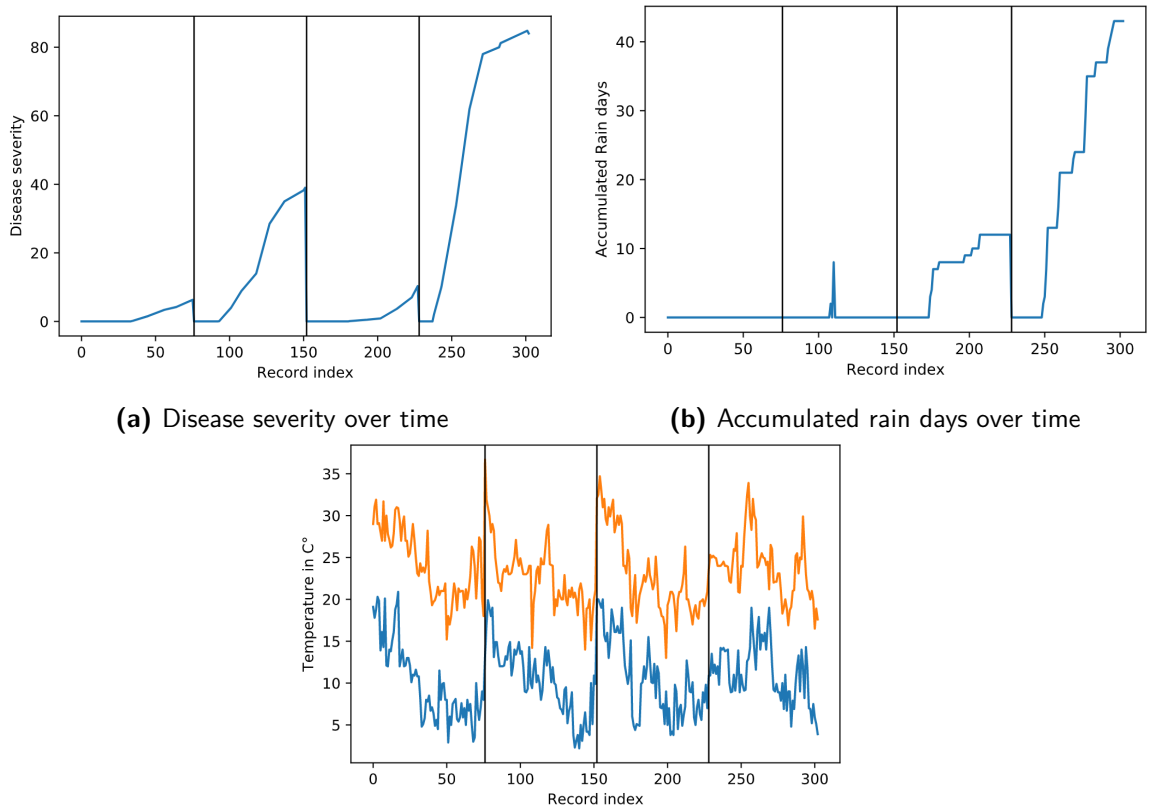
Initial State

The dataset concludes 303 records in four seasons from 2002 until 2006. Each season lasted around four months between October and February. The measurements were not started until mid November with around 40 day after planting when the crop is already on the surface and lasted until 115 days after planting, when the crop was harvested. Each record represents one day of measurement.

Table 4.1 shows the columns in the first dataset provided, as well as information on how the data was recorded. These features are then visualized in Figure 4.2. They are plotted over time which is represented by the index of each record. As stated above, the records are all measured between the 40th day and the 115th day after planting.

Table 4.1 – Original features in potato blight dataset

Column	Type of feature	Description
Date	Observed	Date of observation
DAP	Observed	Days after planting
T_{min}	Observed	Lowest Temperature recorded this day
T_{max}	Observed	Highest Temperature recorded this day
Rain days>0.1mm	Observed	The accumulated number of rain days with rain more than 0.1mm
Season	Observed	The current season of the measurements
Daily blight obsrv.	Observed	Observation of disease severity on the current day



(a) Disease severity over time

(b) Accumulated rain days over time

(c) Highest and lowest temperature over time

Figure 4.2 – Features for prediction model over the four season in the winters from 2002 - 2006

Figure (a) show the disease severity (DS) over time in each season. It is clearly visible that the DS in season two (peak at 39%) was higher than in season one (peak at 6.3%) and three (peak at 10.3%). The DS peaked it season four at 84%. Pesticides were not used during

these experiments which lead to a steady increase in the DS over time due to the lack of treatment.

It is important to note that the infections always started between DAP 50-70 of each season. In the first season, it started on Day 73, in the second season on day 56, in the third season on day 68 and in the fourth season on day 49, which explains the differences in the peaks of the DS.

The second and third plot show the amount of rain during the season as well as the highest and lowest temperature during the four seasons. It is notable that the rain in season four was much higher compared to the other seasons which could indicate a connection between the DS and the rain in future analysis.

The highest and lowest temperature of the day is fluctuating throughout the season but a negative trend is visible indicating that it is getting colder towards January which matches reality

Additional features for Analysis

The original features are used to calculate additional weather related features to help boost the ML models. The additional features are described in table 4.2.

Table 4.2 – Additional calculated features for Potato Blight dataset

Record	Type of record	Description
T_{mean}	Calculated	Daily average temperature
GDD	Calculated	Growing day degree
Accu. GDD	Calculated	Accumulated growing day degree
IP result	Calculated	Length of infection period
lastDS	Calculated	Disease severity on previous day
meanlast3DS	Calculated	Average of disease severity of last three days

GDD calculation: In 2007 Hannukkala et al. determined the thresholds of temperature and rainfall. In this analysis, only those days where the temperature is between 8 and 19 °C and rainfall was above 0.1 mm, were involved in the description of environmental favorable conditions. Correlations between disease severity and accumulative day-degree were estimated and was used and tested to predict the appearance of the first late blight lesions on potato foliage in Egypt. The GDD is calculated as $GDD = T_{mean} - 10$ [12].

Infection period calculation: Infection periods (IPs) were calculated from daily temperature and rainfall obtained from the weather stations. The Hannukkala method requires five consecutive days with minimum temperature 8 °C or above, maximum temperature below than 25 °C and ten days with rain $>$ 0.1mm [12]. The duration of this infection period is accumulated and used for analysis within each season and can be seen in Figure 4.3. The strong fluctuation of the temperature as well as the increased rain in season four results in a much higher infection period than in previous seasons.

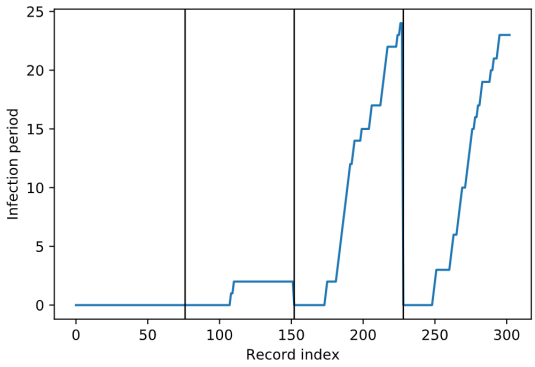


Figure 4.3 – Infection period over time in days

Previous disease severities: In Dr. Fahims experiments, no pesticides were used, leading to steady growth of the blight in the field depending on the environmental factors. To capture this increase, two features were introduced giving information about the situation of the potato blight over the past three days and specifically on the previous day.

In comparison to table 2.2 and the Conclusion in subsection 2.3.3, additional measurements of wind speed and soil moisture can be valuable for the prediction model and should be added in future data collections but will not be taken into consideration in the current example.

4.1.2 Analysis of Cotton Leaf Worm dataset

The second dataset used for this analysis was collected by Dr. Haitham Sharaf from the department of electrical engineering at Cairo University. The set contains data of weather conditions inside a controlled greenhouse system where Egyptian Cotton is planted, among others. The weather data has been collected manually for the past two years and is planned to be recorded for another year. The recorded disease was the number of cotton leaf worms (see Figure 4.4) in the greenhouse.

Figure 4.4 – Cotton leaf worm on a leaf

Initial State

The dataset concludes 130 records between 09/2017 and 02/2020 with each record representing one week of measurement. The Cotton plantation works as a continuous process with new plantations and harvests every day. It takes 35 to 40 days for each individual plant to from seedling to harvest. This leads to a constant state during the entire time of measurement where there are always young and old plants living simultaneously in the greenhouse as well as other plants which will not be analyzed during this research.

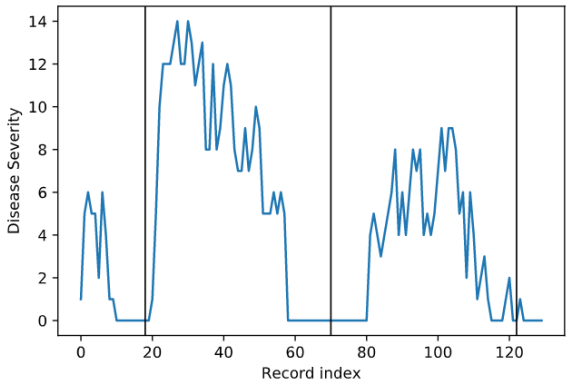
Table 4.3 shows the columns in the second dataset, as well as information on how the data was recorded. These features are then visualized in Figure 4.5. They are plotted over time which is represented by the index of each record. As stated above, each record represents a whole week of measurement.

Table 4.3 – Additional calculated features for Cotton Leaf Worm dataset by

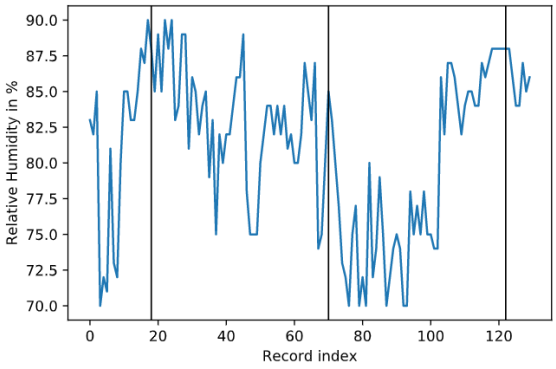
Record	Type of record	Description
No. S. littoralis	Observed	Disease severity
H Temp.	Observed	Highest Temperature recorded this week
L Temp.	Observed	Lowest Temperature recorded this week
RH	Observed	Relative humidity
Bt	Observed	Biological control protocol
Thrips	Observed	Number of Thrips found in greenhouse
Fert	Observed	Indicator for the use of fertilizers during this record

As this greenhouse is used for different plants and the insects related to one plants could also be affecting the cotton, this feature is taken into consideration. The column **Thrips** shows the number of insects captured during the week.

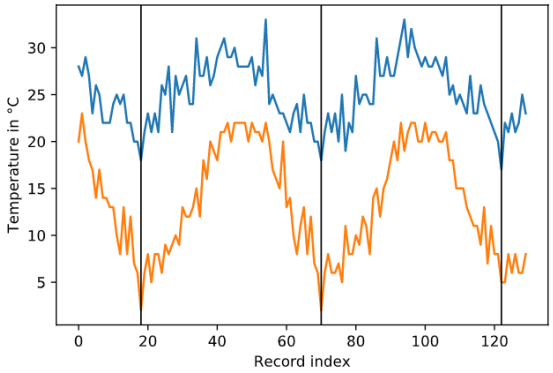
The column **Fert** is as an indicator to know when fertilizers were used in the greenhouse to measure the effects of these fertilizers on the Disease Severity.



(a) Disease severity over time



(b) Relative Humidity over time



(c) Highest and lowest temperature over time

Figure 4.5 – Features for prediction model over three years in 2017-2020

Figure (a) shows the disease severity (DS) over time in each season. In contrary to the potato blight dataset, the disease severity is not linearly increasing. This is due to the controlled environment of the greenhouse and the biological control protocol. 8 traps were set inside the greenhouse and the total number of captured leaf worms per week was recorded in this dataset.

As the plantation of the cotton was a constant process with the same amount of plants in every stage at all times, the DAP was not recorded and the variation in the DS is expected to be tracked back to the environmental factors. As this was a practical experiment, a biological control protocol was used to fight the leaf worms. An insect is admitted into the greenhouse at certain points that tackles the harmful worms without affecting the plants.

Peaks in the disease severity can be observed in 2018 as the control protocol was not yet tested effectively. It is clearly visible that the protocol had a positive effect on the plants in 2019. As the test is planned to continue until 2021, promising insights can be expected that will be useful to tune a ML algorithm for a more effective schedule for the treatment method.

The second and third plot shows the relative humidity as well as the highest and lowest temperature for three years. It is notable that the relative humidity in 2018 was much higher

than in 2019 which could have led to the increased disease severity in 2018. Further analysis will be conducted in the next section.

The highest and lowest temperature of the day is fluctuating throughout the weeks but indicate a similar pattern like in the potato blight dataset. This plot was created with the weekly recordings from the second dataset.

Additional Features for Analysis

The original features are used to calculate additional weather related features to help boost the ML models. The additional features are described in table 4.4.

Table 4.4 – Additional calculated features for the Cotton Leaf Worm dataset

Record	Type of record	Description
T_{mean}	Calculated	Weekly average temperature
GDD	Calculated	Growing day degree
lastBt	Calculated	Bt on previous week
lastDS	Calculated	Disease severity on previous day
meanlast3DS	Calculated	Average of disease severity of last three weeks
diffTHigh	Calculated	Fluctuation in highest temperatures in the current week
diffTLow	Calculated	Fluctuation in lowest temperatures in the current week

GDD calculation: The GDD calculation is described in depth for the potato blight dataset in subsection 4.1.1. As the greenhouse is operated continuously without planting seasons, the Accumulated GDD will not be used as the accumulation can not be reset.

Bt calculations: Bt is the name of the insect used for the biological control protocol. The previous Bt calculation simply shows whether the protocol has been used in teh previous week.

Previous disease severities: The previous disease severities were calculated with the method as in subsection 4.1.1. The importance of this feature is expected to be different than in the potato blight analysis, as the measurements were taken weekly and the previous record thereby represents the last week. As the biological control protocol can be activated during that time, the feature importance could be affected.

Temperature fluctuation calculations: Each record represents a whole week, therefore allowing only one temperature measurement per week. An additional feature has been introduced to the dataset to represent the fluctuation in the temperature during the measurement period and tune the ML algorithm. For these features, the temperature data of the entire period was collected. The highest and lowest temperatures in every week were compared and the difference between the highest THigh per day and the lower THigh per day was recorded in the fluctuation feature. The same method was used for the lowest temperatures.

Similar to the potato blight dataset, the features proposed in subsection 2.3.3 like the wind speed and soil moisture are still to be added to achieve the best possible results with the ML algorithms.

4.2 Results

The datasets described in section 4.1 will be analyzed using the ML models as described in section 2.2 to determine the accuracy of each model and rank their performance. The best performing ML models will then be used for hyper-parameter tuning to achieve better results.

After achieving the initial analysis results, the features of the ML models can further be adjusted to reach a better understanding of the relations between the environment and the spreading of certain diseases in the field of agriculture.

To rank the performance of the ML models an error function needs to be introduced. This error function compares the predicted values with actual data points provided from the test set. The most popular error functions are the RRMSE and the MAPE [18].

RRMSE: The Relative Root Mean Square Error is the relative standard deviation of the residuals. These prediction errors measure how distant from the actual values the predicted values are. It is also a measure for how spread out they are. It simply explains how concentrated the data points are around the line of best fit and is also commonly used in climatology, forecasting, and regression analysis to verify experimental results. With d_i being the actual value and f_i being the forecast value, the formula is given by:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(d_i - f_i)^2}$$

The RMSE is then divided by the average of the actual values to get the RRMSE for analysis.

MAPE: The mean absolute percentage error is a measure for how accurate a forecast system is. With d_i being the actual value and f_i being the forecast value, the formula is given by:

$$MAPE = \frac{1}{n}\Sigma_{i=1}^{n}\frac{d_i - f_i}{d_i}$$

As described in subsection 2.2.1, the dataset is split into a training and a test set. In the following analysis, the split is 80:20 as is typical in most ML analysis. The sampling will differ in both analyses.

4.2.1 Results on Potato Blight dataset

This subsection describes the ML analysis performed on the first dataset provided about Potato Blight from subsection 4.1.1.

Six different algorithms were tested, Random Forest, Extra Tree Regression, Linear Regression, Xtreme Gradient Boosting, State vector Machines and Logistic Regression and the results are collected in Table 4.5.

After fitting and training the model with the data points and the DS for each record, the model is applied to the features of the test set. The predicted values for these combinations of environmental data are then compared with the actual DS of these records using the error functions. The Table shows the RRMSE and the MAPE achieved for the predictions using each of the above algorithms.

In the first part of this analysis, all the observed features from Table 4.1 were used and the model was enhanced by the additional environmental features like the GDD and the IP. Information on previous DS was not introduced at this point.

The five different sampling methods were used for analysis, splitting the dataset in five different ways. For the first four predictions, the set was split for each of the four seasons, meaning that for the first prediction, the training set consisted of all the records from season two-four while the data from season one was predicted using that fitted model. The DS of season one was then used for evaluation.

A fifth and more common method was then used, the random sampling. Here the dataset is split randomly using the above mentioned 80:20 rule using records from all seasons in the training as well as in the test set. This method ensures that other differences between the seasons are captured as well.

Table 4.5 – ML analysis on Potato Blight dataset without information on previous disease severity

Sampling	RF		ExtraTrees		Linear		XGB		SVM		Logistic	
	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE
S1	1392%	1044%	1231%	964%	1461%	1049%	541%	314%	370%	348%	1640%	1214%
S2	125%	101%	120%	95%	129%	98%	124%	95.15%	122%	87.27%	121%	90.5%
S3	2138%	1669%	2973%	2404%	1749%	1448%	2336%	169 8%	2506%	2053%	1074%	550%
S4	111%	94.36%	111%	94.74%	172%	143%	114%	97.25%	118%	99.84%	118%	100%
Random	18.18%	8.75%	17.97%	7.89%	78.75%	54.9%	89.09%	44.36%	182%	95%	125%	51.1%

The same predictions were then performed an additional time adding the information on the previous DS to each record. Again the predictions were made using all six ML models and the five different sampling techniques for comparison of the two error metrics. The results are listed in table 4.6.

The prediction errors can be seen to have improved after introducing the previous disease information while the classification algorithms still underperform with regards to the other algorithms.

Table 4.6 – ML analysis on Potato Blight dataset with information on previous disease severity

Sampling	RF		ExtraTrees		Linear		XGB		SVM		Logistic	
	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE
S1	29,07%	15,04%	28,85%	18,74%	43,54%	38,85%	270,62%	55,56%	369,10%	347%	898%	558%
S2	42,92%	23,07%	32,89%	24,60%	5,85%	4,94%	106,61%	72,28%	128,70%	94,70%	55,77%	40,03%
S3	49,04%	30,09%	462%	372%	232%	175%	411%	95,33%	329%	309%	477%	340%
S4	67,7%	27,94%	65,93%	27,16%	2,87%	2,25%	66,64%	53,09%	116,50%	98,40%	68,78%	57,42%
Random	2,82%	1,57%	2,35%	1,23%	3,7%	2,93%	4,45%	2,16%	181,40%	95,20%	5,92%	2,88%

The results of these analyses will be discussed in chapter 5 at length. The differences between the algorithms will be explained as well as the effect of the sampling and the introduction of the windows.

The errors from tables 4.5 and 4.6 may vary when measured again as the random sampling uses different train test splits each measurement.

4.2.2 Results on Cotton Leaf Worm dataset

This subsection describes the ML analysis performed on the second dataset provided about the Cotton Leaf Worm from subsection 4.1.2.

Similarly to subsection 4.2.1, five different algorithms were tested and the RRMSE and the MAPE recorded for analysis. The SVM was not tested as it proved to be inefficient for these use cases. Table 4.7 shows the results achieved for the predictions of the DS using each algorithm.

In the first part of this analysis, all the observed features from Table 4.3 were used and the model was enhanced by the GDD features. Information on previous disease severity and the temperature fluctuations was not introduced at this point.

For the second approach, the data of the previous DS as well as the averaged DS over the last three weeks was introduced into the dataset. This led to slight improvements in the prediction model.

A similar behavior was recorded when the temperature relating data was introduced in the third approach. The fluctuation of the temperatures within each week seems to have a minor effect on the prediction algorithm.

Lastly the information about the Thrips was excluded from the model to test the indirect impact of these insects on the Cotton Leaf Worm. This resulted in better performance of the XGB and poorer performance in the Logistic Regression leaving the other three algorithm performances unchanged. This approach will need to be reevaluated further in the discussion.

Random sampling was chosen in all approaches as the biological control protocol was adapted with time and the records are all equal in terms of plant age and count. Due to the limited number of records, the test set had a length of 22 records.

Table 4.7 – ML analysis on Cotton Leaf Worm dataset with random sampling

Sampling	RF		ExtraTrees		Linear		XGB		Logistic	
	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE
Basic	27,86%	21,17%	24,51%	19,66%	36,57%	30,27%	33,84%	25,15%	41,5%	29,45%
Prev DS	27,12%	20,59%	26,08%	18,96%	24,17%	19,37%	24,43%	16,56%	35,25%	20,25%
Temp	27,03%	20,86%	21,92%	18,02%	36,42%	30,20%	33,84%	25,15%	45,44%	31,29%
Thrips	27,87%	21,67%	24,26%	19,90%	35,91%	29,46%	28,67%	19,63%	49,76%	34,97%
Total	27,41%	21,04%	25,01%	18,13%	24,31%	19,35%	25,61%	17,79%	31,44%	21,47%

The last row shows the results achieved when all features were combined. This includes the previous DS, the number of Thrips as well as the temperature fluctuation and leads to stable results throughout the model

The results of this analysis will be discussed in Chapter 5 in depth and further modifications will be made to achieve better results.

5 Discussion

In this chapter, the results of the initial analysis in chapter 4 will be discussed and the different algorithms from subsection 2.2.1 are compared with regard to the prediction errors. The differences between the algorithms are explained and the feature importance for further analysis is presented.

5.1 Potato Blight

Looking back at the results from table 4.5 (Prediction errors without information about previous disease severity) and 4.6 (Prediction errors with information about previous disease severity) one can see strong discrepancies when previous time windows were used as well as which sampling methods and algorithm were used.

5.1.1 Classification versus Regression

Both tables show the largest errors when using the classification method SVM, where error rates of far more than 100% are reached.

SVM is normally used in binary classification problems, where interests are the probability of an outcome occurring. Probability ranges from 0 and 1, where the probability of something certain is 1, and 0 is something very unlikely to happen. In this example, an absolute number needs to be predicted, which can range outside 0 and 1.

It is possible to limit any value greater than 1 to be 1 or to normalize the values in the dataset (scaling the numbers to have values between 0 and 1) to be able to use the same underlying method of SVM. This method is called Linear SVR (Support vector regression). Still, SVR proves to be much less effective than other regression algorithms as can be seen in figure 5.1. This is due to the fact that simple regression focuses on minimizing the error rate, while SVR tries to fit the error within a certain threshold. This technology can not be applied for this application as the dataset does not contain multiple recurring information with slight variations as is the case with classification problems.

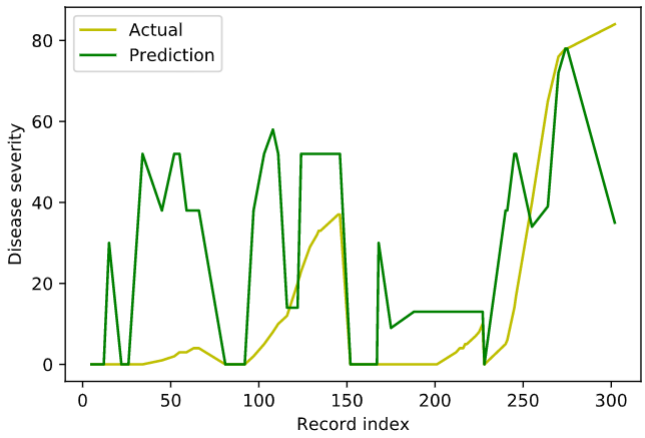


Figure 5.1 – SVM with random sampling on test set with information on previous disease severity

In any case, regression algorithms are more suitable for predicting continuous outputs, such as predicting the price of a property or in this case, disease severities. Its prediction output can be any real number, range from negative infinity to infinity as the regression line is a continuous function.

In further discussion, the classification method SVM will not be further reviewed. This work will focus on regression algorithms in the further discussion.

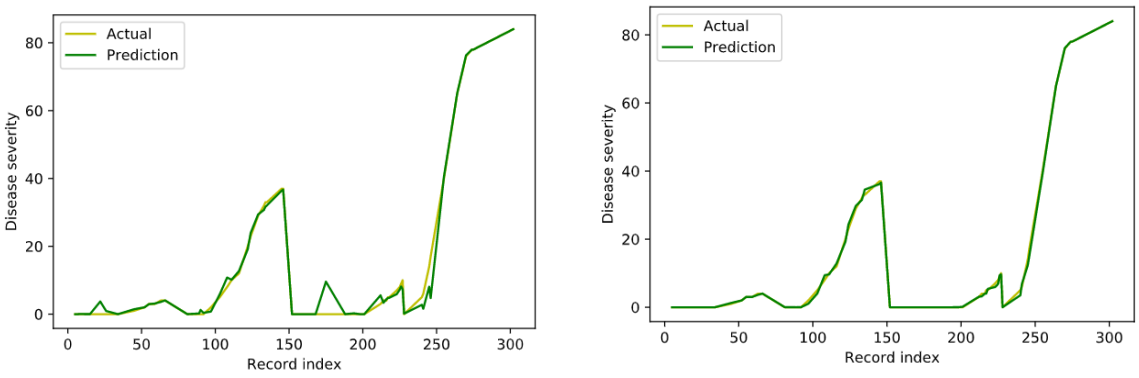
5.1.2 Information on Previous Disease Severity

This subsection will focus on the effect of introducing information on previous disease severity to each observation on the prediction models. For simplicity, the Extra Tree regression algorithm is used for comparison on a test set with random sampling.

The introduction of the two additional features about the previous disease severity has shown a positive effect on all prediction models. This is due to the fact that every observation is exposed to the DS of the observation before it and the gradient seems flat.

As this model is using the DAP (Days after planting) and a random sampling strategy can be used the additional features could be seen as redundant and could lead to overfitting the model, meaning a model that models the training data too well.

Overfitting happens when a model learns the details and fluctuations in the training data to the extent where it could negatively impact the performance on the new test data. The problem is that these fluctuations might not apply to the test set and could thereby negatively impact the model's ability to generalize and predict.



(a) Without previous DS information

(b) With previous DS information

Figure 5.2 – Extra Tree Regresson with random sampling on test set with and without information on previous disease severity

In this example, the model is not being overfitted and the introduction of the windows leads to an improvement in the results as can be seen in Figure 5.2. This could be due to the fact that the gradient of the DS is not as flat as assumed in the hypothesis above. The random sampling could also lead to removing three or more consequent observations at once in the test set, leaving the model clueless on how the disease is behaving.

As a result of this comparison, the use of information on previous disease severity is suggested with caution. It will be introduced in further discussion.

5.1.3 Sampling

This section will compare the five different sampling techniques with one another. The four sampling techniques where three seasons were used for training and the fourth used for testing as well as the random sampling method.

The sampling can highly influence the outcome of the predictions as the training set is used to generalize properties from the training set to the test set and later for future predictions. Two main ways to introduce errors into your training set include selection bias and sampling error:

- **Selection Bias** : Caused when the method of drawing observations skews the sample
- **Sampling Error** : Caused due to the random nature of drawing observations skewing the training set

Looking back at both tables 4.5 and 4.6 it is clear that using seasons one and three performed worse than seasons two and four. However, applying the random sampling method outperformed seasonal sampling in all cases.

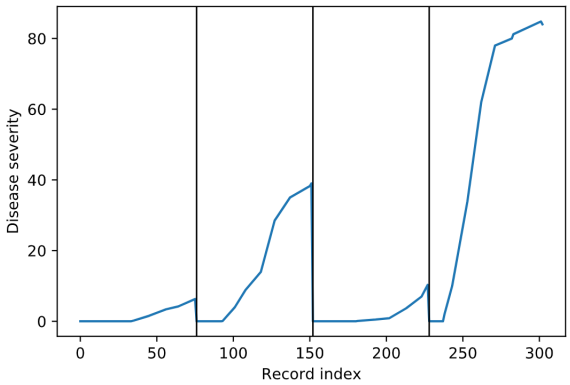
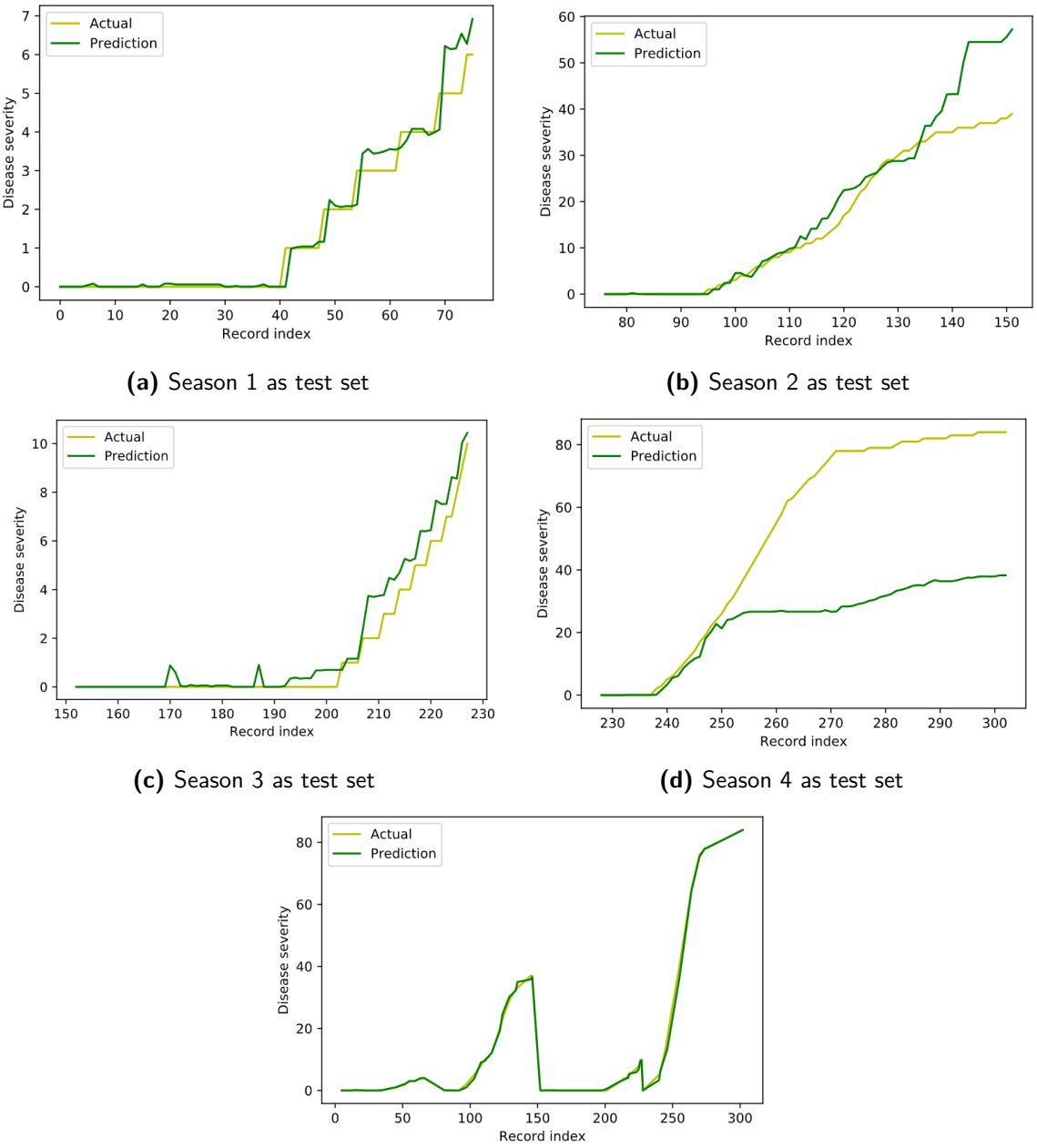


Figure 5.3 – Disease severity of Potato Blight over time

Figure 5.3 shows once again the disease severity in the four observed seasons as described in section 4.2.1. It is visible that the disease severites vary drastically between the seasons. Using three of the given seasons to predict the fourth season in this case clearly leads to Sampling bias errors.



(a) Season 1 as test set

(b) Season 2 as test set

(c) Season 3 as test set

(d) Season 4 as test set

(e) Random Sampling

Figure 5.4 – Random Forest on five sampling strategies for test set with information on previous disease severity

Figure 5.4 shows the five predictions for each of the test cases (Random Forest as an example) while the green line represents the predicted values and the yellow line represents the actual values. It is clearly visible that testing on samples one and three lead to very strong over predictions while testing on season four leads to a strong under prediction as the other three seasons had way lower disease severities. Only the fifth test case shows acceptable results as information from all five cases are used to fit the model.

The errors are much higher when overpredicting the disease in a season with lower severity than when underpredicting the disease in a season with higher severity. As an example, season three and four are compared. The average absolute error in Season three is 0.42 and the average actual value is 1.41 . This leads to a relative error of 30%. On the other hand, when predicting season four the absolute error is a similar 27.94 but when compared with the average actual error of 51.4, the absolute error results in a relative error of 54.35%. Due to the nature of the individual seasons, the increase in the absolute error from 0.42 to 27.94 only resulted in an increase in the relative error from 30% to 54%.

As a result, it is clear that in order to get a more complete picture on the behavior of the disease in this test environment it is recommendable to use random sampling for al further analysis. Additional tests should be conducted to ensure more clarity on the environment and ruling out sampling bias.

5.1.4 Simple Regression versus Decision Trees

The analysis from the three subsections above has concluded that better results can be achieved using regression models, introducing information on previous disease severity, and using random sampling to fit the prediction models.

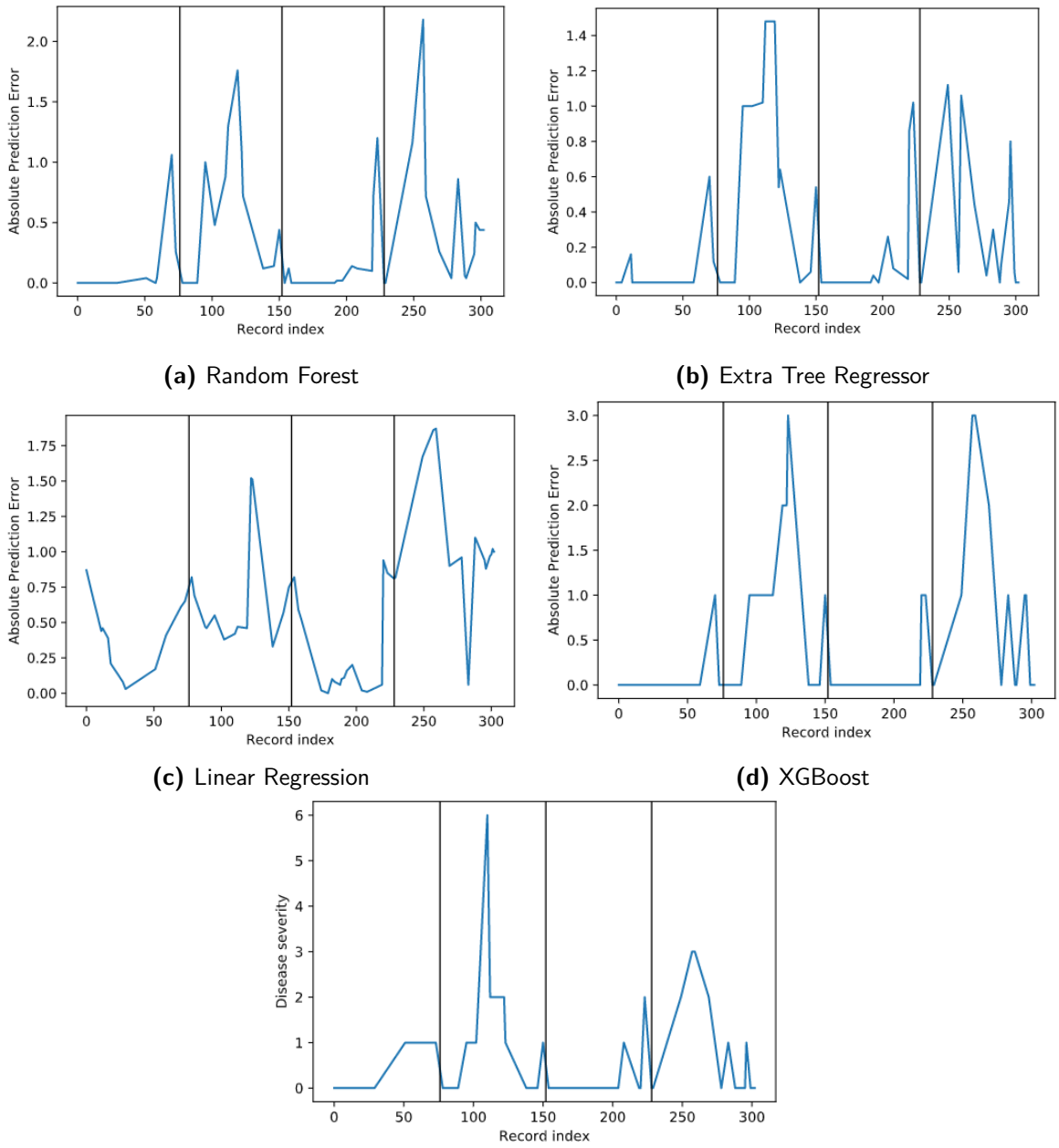
In this subsection, the Linear and Logistic Regressions will be compared as well as the Decision Tree algorithms Random Forest and XGBoost to explain differences in the results as well as the reasons for these discrepancies.

Looking back at table 4.6, it is clear that the algorithms have similar performances when it comes to the prediction errors. Table 5.1 shows the errors from table 4.6 in a compressed version.

Table 5.1 – Prediction Errors of all algorithms with random sampling on the test set with information on previous disease severity

RF		ExtraTrees		Linear		XGB		Logistic	
RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE	RRMSE	MAPE
2,82%	1,57%	2,35%	1,23%	3,7%	2,93%	4,45%	2,16%	5,92%	2,88%

As can be seen in Figure 5.2 the errors on the predicted values of the Extra Tree regressor are difficult to spot. To be able to further analyze the differences between the predictions, only the absolute errors in the predicted values over time are shown in Figure 5.5.



(a) Random Forest

(b) Extra Tree Regressor

(c) Linear Regression

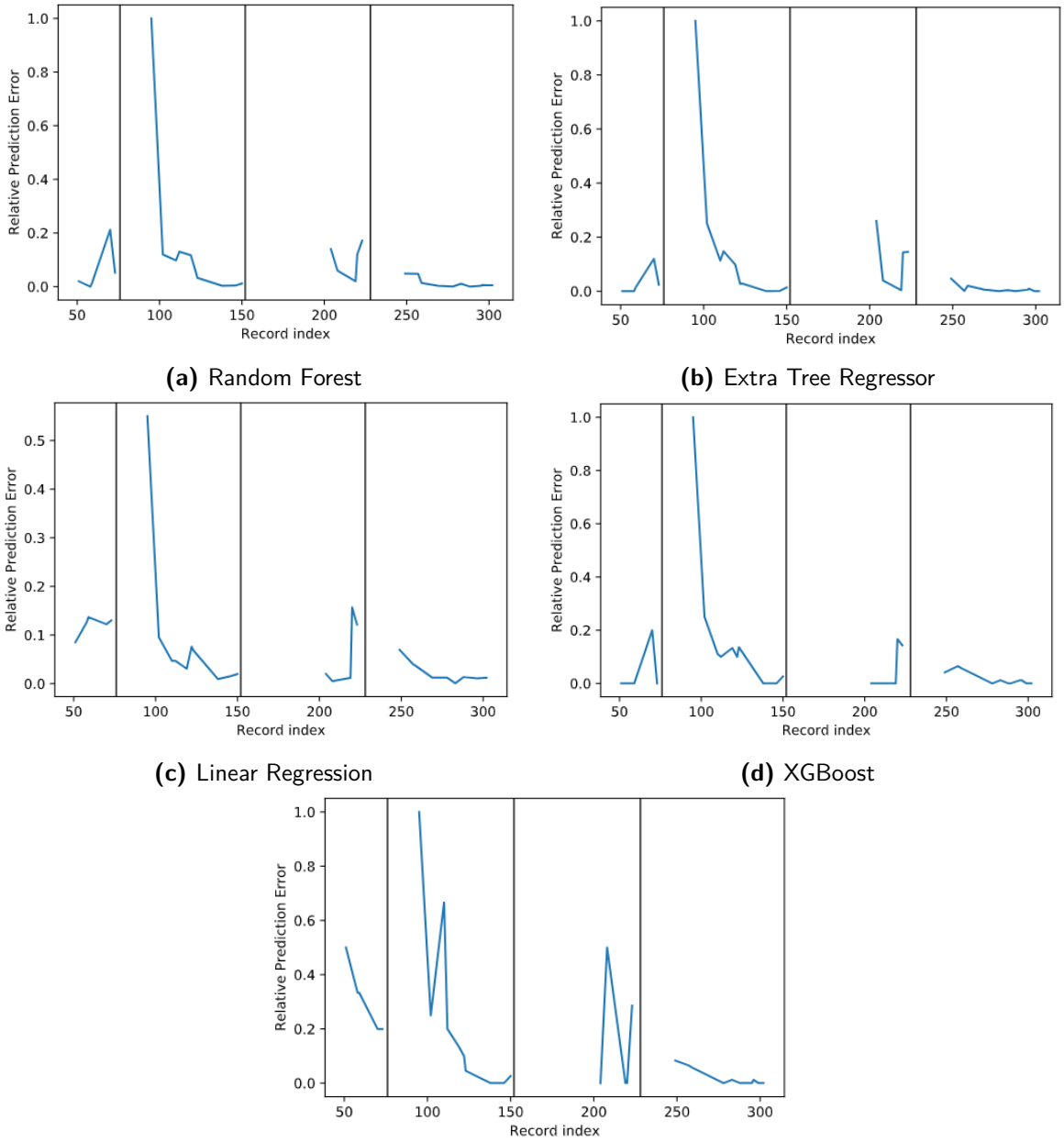
(d) XGBoost

(e) Logistic Regression

Figure 5.5 – Absolute Errors of all algorithms with random sampling on test set with information on previous disease severity

All algorithms perform better in Season one and three where the actual values were expected to be small. However, this visualization skews the results as the same relative error in Seasons three and four, for example, would lead to much higher absolute errors in Season four.

To avoid these skewed results the relative error is shown in Figure 5.6 where the relative error is the absolute error divided by the actual expected values. The missing values in between, represent data points where the expected value was 0 (As there can be no dividing by 0).



(a) Random Forest

(b) Extra Tree Regressor

(c) Linear Regression

(d) XGBoost

(e) Logistic Regression

Figure 5.6 – Relative Errors of all algorithms with random sampling on test set with information on previous disease severity

Figure 5.6 gives a better overview of the prediction error of all algorithms. Compared to the visualization before, Season four is actually performing much better than the other Seasons as the actual values are expected to be very high thereby leading to low relative errors.

In the following part, the differences in the algorithms will be explained.

It is clear that the Logistic regression performs less than the three decision tree algorithms use in this research. This could be due to the fact that decision tree algorithms emphasize

feature selection. They weigh certain features as more important than others. They also do not assume that models have linear relationships, like regression models do. A random forest, for example, takes random samples, creates many decision trees, and then averages out the nodes to get a clearer model.

Decision trees, in general, are transparent and easily interpretable algorithms and the main reason to use them is to analyze quantitative and qualitative patterns in the dataset to find hidden correlations as is the case in the current analysis. They are an ensemble approach that combines many base models to create predictions. While building the trees, a number of small trees are grown such that every successive tree focuses entirely on the attributes of the training set that have been missed in the preceding one [7].

This makes them superior to logistic and linear regression in this case. Looking back at the relative errors, one can also see that the regression predictions have a very low accuracy as almost every datapoint shows an error larger than zero. As all algorithms fail to achieve accurate results in Season two, the logistic regression performs poorly in the other seasons as well.

Given that all three remaining technologies perform similarly the feature importances are compared as well in table 5.2. Given that the RF and the ET almost solely rely on the information on previous disease severity, for this part of the analysis, the feature importance is also plotted for the case without information on previous disease severity.

Table 5.2 – Feature importance on potato blight analysis with Decision Tree algorithms

	With lastDS			Without lastDS		
Feature	RF	ET	XGB	RF	ET	XGB
DAP	0,3%	0,2 %	9,3%	2,2%	3,3%	14,4%
TMIN	0,0%	0,0 %	5,4%	0,3%	0,7%	9,1%
TMAX	0,0%	0,0 %	4,6%	0,2%	1,4%	8,7%
TMEAN	0,0%	0,0 %	6,5%	0,2%	1,7%	9,9%
RainDays	2,7%	13,1 %	16,1%	82,3%	77,2%	29,6%
GDD	0,0%	0,0%	0,0%	0,3%	2,5%	0,0%
Accu. GDD	0,5%	0,2%	7,8%	3,9%	2,8%	17,6%
IP Result	0,7%	0,6%	6,6%	10,6%	10,5%	10,8%
lastDS	46,5%	41,6%	25,9%	/	/	/
meanlast3DS	49,1%	44,2%	17,8%	/	/	/

The difference between the random forest and the extra tree algorithms lies mainly in the fact that, instead of calculating the optimal feature combination locally (random forest), in the extra tree algorithm a random value is selected for the split. This adds up to to a more diversified tree with less combinations to evaluate when fitting an extra tree model.

However, the extra tree model performs slightly better. This can be traced to the different feature importances from table 5.2. While the random forest allocates 95% of its importance to the information on previous disease severity, the Extra tree, allocates only 85% to the last disease severity and 13% instead of only 2% to the number of Rain days.

This is a clear indication of the importance of the feature Rain days when fitting the model. This is the strongest weather related feature and can describe the strong differences in the disease severity in Season two and season four. Even comparing this feature with the other models when information on previous DS is not used the Rain days are still the strongest feature.

The question that still needs to be answered is why the XGB performs differently from the other two Decision Trees (DT) algorithms. To decrease the prediction error the best tradeoff between the bias and variance in the DT needs to be found. A shallow DT results in a high bias and low variance, whereas a too deep DT has a low bias but high variance.

The random forest is very a bagging algorithm meaning that it generates random samples from the dataset to reduce the variance of the model. Therefore, using the random forest leads to deep trees as they have a low variance, but increases the bias [13] [27].

Boosting reduces variance and bias as it uses multiple models (bagging). Thereby, it trains the subsequent model by telling it what errors the previous models made using the difference between the predicted and actual values. The base learner must be weak. If the data is overfitted, there won't be any errors for the subsequent models to build upon [11] [24].

The effect of the bias and the variance can also be seen in table 5.2 with regard to the feature importance. The focus of the RF and the ET on fewer features is grounded by the higher bias, whereas the XGB is giving more importance to all features and only 44% to the information on DS.

By definition, this more equally distributed feature importance reduces the bias but could lead to overfitting. Even without regarding the previous DS the XGB still has a much more equal distribution of feature importance than the other two algorithms who perform very similarly. However, the XGB performed poorly when not provided information about previous DS.

This could mean that using the XGB, the variance was too high, detracting from the actual important features for fitting this model.

5.1.5 Result

Based on the discussion above, the best results can be achieved for this dataset, when random sampling is used for dividing the dataset and a Decision Tree algorithm is used to predict further disease severities.

The Random Forest and the Extra Tree algorithm outperformed the other algorithms reaching error rates of only 7,98% without information on previous disease severity and only 1,23% when given information about previous disease severity.

In every model, the RainDays is the most important feature with regard to prediction errors, followed by the Infection period (IP), the Days after planting and lastly the temperature features. This is consistent with the assumptions in Chapter 2. It is expected that results could be further improved with more data to get a better picture of the strong variations between the seasons. The humidity could be expected to also further improve the results.

5.2 Cotton Leaf Worm

A similar analysis procedure to the one from section 5.1 will be applied to the second dataset in this chapter. The accuracy of the data on the Cotton Leaf Worm is expected to be lower as only two years of data are provided and the measured data points are on a weekly basis.

Following the insights from chapter 5.1 random sampling will be applied for the prediction models and the classification algorithm SVM will not be considered. The short amount of time does not allow seasonal sampling as a whole cycle needs to be used as the test set and with only two years provided this would leave only 50% of the datapoints for training. The Classification algorithm will not be considered again as the results are expected to be similar to the results in section 5.1.1 due to the nature of this research.

Rather, the difference between the features will be compared and their impact on the model as well as a section about the difference between the Regression and the Decision Tree approaches. Additionally the importance between the RRMSE and the MAPE will be explaied further.

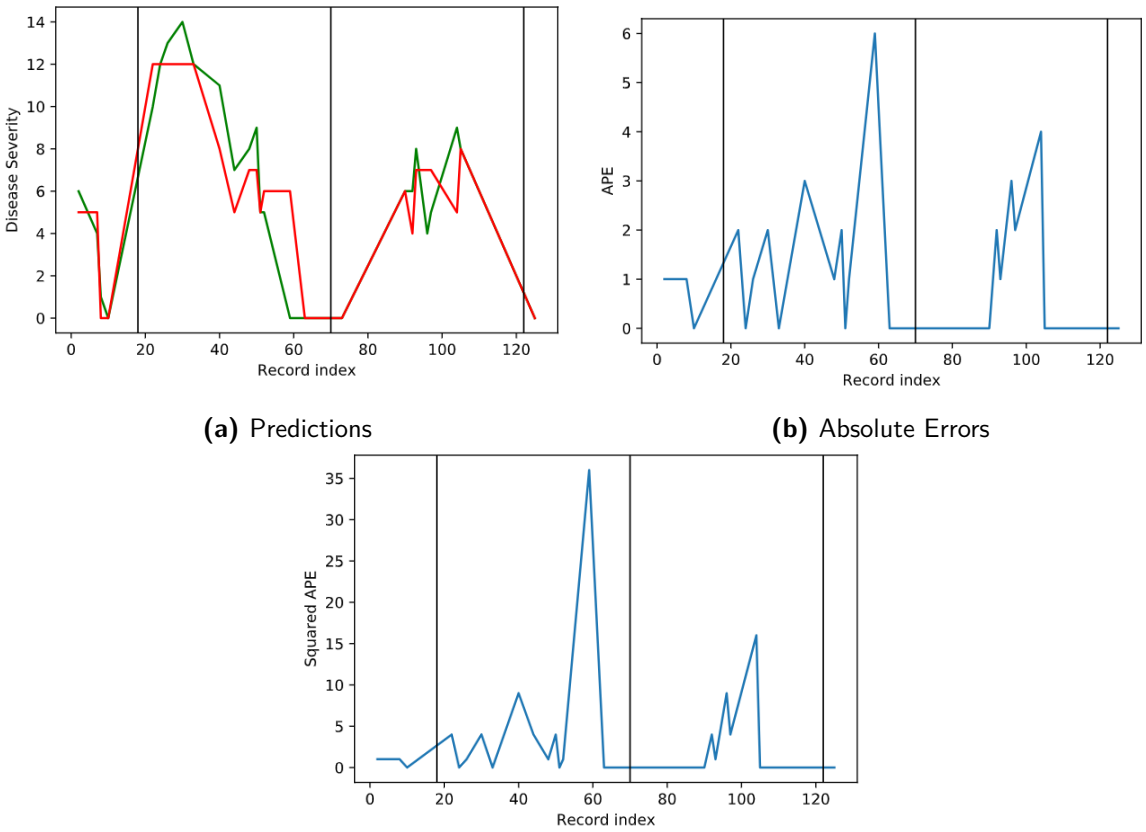
5.2.1 RRMSE versus MAPE

In table 4.7, the prediction error is captured in two ways, the Relative Route Mean Squared Error (RRMSE) and the Mean Absolute Percentage Error (MAPE) as described in section 4.2.

In contrast to table 4.6, in this experiment the RRMSE is higher than the MAPE by at least 5%. By squaring before averaging the errors in the RRMSE more weight is put on the records with larger errors.

In the first dataset, the difference between the two values was 1-2%, whereas in the second dataset the difference is up to 5-12%. This indicates that the predictions in the models have multiple outliers.

Figure 5.7 shows the prediction result when using Logistic regression on the second dataset with all features included, where (b) and (c) are the absolute error and the squared absolute error of the prediction.



(a) Predictions

(b) Absolute Errors

(c) Squared Absolute Errors

Figure 5.7 – Logistic Regression on Cotton Leaf Worm dataset with all features

When averaging the errors from figure (c) it is expected that the errors at record 59 and 103 have a higher impact on the overall error, leading to an RRMSE of 31,44% instead of 21,47% with the MAPE. The outliers have a much bigger impact on the overall error calculation.

This could be due to the small number of records provided in the second dataset. After the completion of the third year of the testphase, the results are expected improve as the size of the dataset will likely increase by 50%.

For further analysis, only the MAPE will be regarded in the further discussion of he results. Table 5.3 shows a compressed version of table 4.7 without the RRMSE.

Table 5.3 – MAPE on second dataset from table 4.7

	RF	ExtraTrees	Linear	XGB	Logistic
Basic	21,17%	19,66%	30,27%	25,15%	29,45%
Prev DS	20,59%	18,96%	19,37%	16,56%	20,25%
Temp	20,86%	18,02%	30,20%	25,15%	31,29%
Thrips	21,67%	19,90%	29,46%	19,63%	34,97%
Total	21,04%	18,13%	19,35%	17,79%	21,47%

5.2.2 Additional features

First the baseline was established by calculating the predicted error with the basic observed features from table 4.3. This resulten in errors between 19,66% with the Extra Tree Regressor and 30,27% with the Linear Regression.

In the second approach, information about the previous Disease Severity was introduced to the dataset before fitting the model. This resulted in

5.2.3 Simple Regression versus Decision Trees

Similar to the results from the first dataset in section 5.1.4, the regression algorithms perform less than the decision tree algorithms, where the Linear Regression performs slightly better than the Logistic Regression.

As stated above, this could be due to ability of Decision Trees to weigh certain features as more important than others, rather than assuming that features have linear relationships [7].

This is especially the case when information on previous Disease Severities were not provided in the model where prediction errors of 29,45-34,97% were calculated. These features seem to correlate strongly to the current Disease Severity and the lack of these information leave the model with a gap of information. The Decision Tree algorithms are able to tackle this lack of information by increasing the importance of the other features in this case.

xx xx xxx Feature Importance

5.2.4 Result

Based on the discussion above, the optimum results can be achieved for this dataset, when improving the model by adding calculated features about the temperature and the previous DS.

6 Summary and outlook

This final chapter will summarize the results of this research paper. It will also give an overview of the answered research questions and an outlook to future work that can be done in this area.

6.1 Summary

We have done our analysis to answer the research question to present the impact of machine learning (ML) in precision agriculture (PA) to increase productivity and maximize the yields of crops by detecting diseases in plants before they spread irreversibly.

A cloud-based IoT platform consisting of three layers was implemented at a xx (size of farm and place) farm in xx. The layers work together as a controlled system transporting the signals transmitted at each node back to the main server where the analysis is performed. The signal is then cast back to each node with the specific command or action. The command can also be reported to the farmer responsible to inform them about upcoming and predictions.

It is clear that the temperature and the humidity measurements vary between different points in the ground. Therefore multiple sensors will be used in the same node (on the same watering line) representative of one of the four subareas.

It was found that Decision Tree algorithms would lead to the best results given the nature of the analyzed datasets. To achieve optimum results both the Extra Tree/Random Forest as well as an XGB should be used and compared for each further use case to find the best tradeoff between bias and variance in the specific dataset.

When analyzing the available datasets on the Potato Blight and the Cotton Leaf Worm, multiple insights were gathered for future precision agriculture researches. The measurements should be conducted at least on a daily, as weekly observations could skew the data.

The dataset should also include all necessary information about the environment, especially when the experiment is not being conducted in a controlled area. This includes information about the use of fertilizers and pesticides as well as the measurements of diseases from other plants within the test field.

Required features to increase accuracy are the temperature related features relative humidity and temperature as well as further features derived from these measurements as can be seen in Chapter 5. Additional features like pH or solar radiation don't seem to be necessary at this point but can be added in the future.

Prediction errors as small as 1,23% can be expected for future analysis. However, for this to be possible, enough data needs to be captured from the environment. After discussing

the results of the analysis it can be assumed that the future expansion of the project will be beneficial for farmers in Egypt and will help reduce the amount of pesticides used for disease prevention can be reduced.

Pesticides or other methods for prevention can be utilized in more targeted manners, which would lead to a reduction of the costs of the crop and an increase of the amount produced. The division of the test field into subareas will help capture the impacts of these areas on one another.

6.2 Outlook

As this project is expected to run for multiple years, a lot of additional work can be done to improve results and ensure their validity of the measurements while keeping the IoT system maintainable.

6.2.1 Longer Measurement Time

In contrary to many research fields, precision agriculture does not just require dedication and commitment but also a lot of time. After implementing the necessary measurement system, the data needs to be gathered over time. This includes multiple full cycles in terms of years or seasons as to be able to recognize patterns and correlations between the features and the disease.

The more records are available for fitting the ML model, the more accurate the predictions will be, as more patterns and correlations repeat themselves over time.

6.2.2 Parameter Tuning

Hyperparameters of the prediction models can be tunes. These are variables that control the training process itself. In case of the decision tree algorithms, the hyperparameters decide how many trees are used. These variables are not directly linked to the training data, but configure the model and are usually constant during a job.

6.2.3 Add Features

Additional to the features observed and calculated in this research, other features can be included to improve the ML models.

The measurements of the wind speed and solar radiation can be included in the IoT on the test field. Furthermore, other environment related information can be included like drought or plagues like the plague of locusts in africa in 2020.

6.2.4 Validate on Different Crops

To ensure the validity of the precision agriculture approach used in this research, the system can be applied on multiple crops measuring different diseases simultaneously.

Additionally to the Potato Blight and the Cotton Leaf Worm, other insects can be captured such as thrips which have a direct impact on the health of other crops

Bibliography

[1] Yile Ao et al. “The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling”. In: *Journal of Petroleum Science and Engineering* 174 (2019), pp. 776–789. ISSN: 09204105. DOI: 10.1016/j.petrol.2018.11.067. URL: https://doi.org/10.1016/j.petrol.2018.11.067.

[2] Alaa Adel Araby et al. “Smart IoT Monitoring System for Agriculture with Predictive Analysis”. In: *2019 8th International Conference on Modern Circuits and Systems Technologies, MOCAST 2019* (2019), pp. 3–6. DOI: 10.1109/MOCAST.2019.8741794.

[3] Gouravmoy Bannerjee et al. “Artificial Intelligence in Agriculture : A Literature Survey”. In: 7.3 (2018).

[4] Paul Boissard, Vincent Martin, and Sabine Moisan. “A cognitive vision approach to early pest detection in greenhouse crops”. In: *Computers and Electronics in Agriculture* 62.2 (2008), pp. 81–93. ISSN: 01681699. DOI: 10.1016/j.compag.2007.11.009.

[5] Erik Brynjolfsson, Tom Mitchell, and P Olicy Forum. “What can machine learning do? Workforce implications”. In: *Science* 358.6370 (2017), pp. 1530–1534. ISSN: 0036-8075. DOI: 10.1126/science.aap8062. URL: http://www.sciencemag.org/lookup/doi/10.1126/science.aap8062.

[6] Younes Chtioui, Suranjan Panigrahi, and Leonard Francl. “A generalized regression neural network and its application for leaf wetness prediction to forecast plant disease”. In: *Chemometrics and Intelligent Laboratory Systems* 48.1 (1999), pp. 47–58. ISSN: 01697439. DOI: 10.1016/S0169-7439(99)00006-4.

[7] Fatemeh Davoudi Kakhki, Steven A. Freeman, and Gretchen A. Mosher. “Evaluating machine learning performance in predicting injury severity in agribusiness industries”. In: *Safety Science* 117.April (2019), pp. 257–262. ISSN: 18791042. DOI: 10.1016/j.ssci.2019.04.026. URL: https://doi.org/10.1016/j.ssci.2019.04.026.

[8] Josse De Baerdemaeker. *Precision agriculture technology and robotics for good agricultural practices*. Vol. 1. PART 1. IFAC, 2013, pp. 1–4. ISBN: 9783902823304. DOI: 10.3182/20130327-3-jp-3017.00003. URL: http://dx.doi.org/10.3182/20130327-3-JP-3017.00003.

[9] FAO. “Food Loss and Waste in Egypt”. In: *Food and Agricultural Organization of the United Nations* (2019). URL: https://www.sciencedirect.com/science/article/pii/S0306919217302440.

[10] Kiarash Ghazvini et al. “Predictors of tuberculosis: Application of a logistic regression model”. In: *Gene Reports* 17 (2019), p. 100527. ISSN: 24520144. DOI: 10.1016/j.genrep.2019.100527. URL: https://doi.org/10.1016/j.genrep.2019.100527.

[11] Tirthankar Goon. *Evolution of Machine learning from Random forest to Gradient Boosting method*. 2018. URL: https://www.linkedin.com/pulse/difference-between-random-forest-gradient-boosting-algo-goon/.

[12] A. O. Hannukkala et al. "Late-blight epidemics on potato in Finland, 1933-2002; increased and earlier occurrence of epidemics associated with climate change and lack of rotation". In: *Plant Pathology* 56.1 (2007), pp. 167–176. ISSN: 00320862. DOI: 10.1111/j.1365-3059.2006.01451.x.

[13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer New York, 2009. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7. URL: http://link.springer.com/10.1007/978-0-387-84858-7.

[14] L Huber and T J Gillespie. "Modeling Leaf Wetness in Relation to Plant Disease Epidemiology". In: *Annual Review of Phytopathology* 30.1 (1992), pp. 553–577. ISSN: 0066-4286. DOI: 10.1146/annurev.py.30.090192.003005. URL: http://www.annualreviews.org/doi/10.1146/annurev.py.30.090192.003005.

[15] Kirtan Jha et al. "A comprehensive review on automation in agriculture using artificial intelligence". In: *Artificial Intelligence in Agriculture* 2 (2019), pp. 1–12. ISSN: 25897217. DOI: 10.1016/j.aiia.2019.05.004. URL: https://doi.org/10.1016/j.aiia.2019.05.004.

[16] Jekishan K. and Ankit Desai. "IoT: Networking Technologies and Research Challenges". In: *International Journal of Computer Applications* 154.7 (2016), pp. 1–6. DOI: 10.5120/ijca2016912181.

[17] Ahmed Khattab, Ahmed Abdelgawad, and Kumar Yelmarthi. "Design and implementation of a cloud-based IoT scheme for precision agriculture". In: *Proceedings of the International Conference on Microelectronics, ICM* 0 (2016), pp. 201–204. DOI: 10.1109/ICM.2016.7847850.

[18] Konstantinos G. Liakos et al. "Machine learning in agriculture: A review". In: *Sensors (Switzerland)* 18.8 (2018), pp. 1–29. ISSN: 14248220. DOI: 10.3390/s18082674.

[19] Robert J. McQueen et al. "Applying machine learning to agricultural data". In: *Computers and Electronics in Agriculture* 12.4 (1995), pp. 275–293. ISSN: 01681699. DOI: 10.1016/0168-1699(95)98601-9.

[20] M. A. Omran. "Analysis of solar radiation over Egypt". In: *Theoretical and Applied Climatology* 67.3-4 (2000), pp. 225–240. ISSN: 0177798X. DOI: 10.1007/s007040070011.

[21] M. J. Pedro and T. J. Gillespie. "Estimating dew duration. II. Utilizing standard weather station data". In: *Agricultural Meteorology* 25.C (1981), pp. 297–310. ISSN: 00021571. DOI: 10.1016/0002-1571(81)90082-0.

[22] Odysseas Pentakalos. "Introduction to machine learning". In: *Cmg Impact 2019* February 2010 (2019). ISSN: 09521976. DOI: 10.4018/978-1-7998-0414-7.ch003.

[23] Francis J. Pierce and Peter Nowak. "Aspects of Precision Agriculture". In: *Advances in Agronomy* 67.C (1999), pp. 1–85. ISSN: 00652113. DOI: 10.1016/S0065-2113(08)60513-1.

[24] Xiupeng Shi et al. “A feature learning approach based on XGBoost for driving assessment and risk prediction”. In: *Accident Analysis and Prevention* 129.November 2018 (2019), pp. 170–179. ISSN: 00014575. DOI: 10.1016/j.aap.2019.05.005. URL: https://doi.org/10.1016/j.aap.2019.05.005.

[25] Osvaldo Simeone. “A brief introduction to machine learning for engineers”. In: *Foundations and Trends in Signal Processing* 12.3-4 (2018), pp. 200–431. ISSN: 19328354. DOI: 10.1561/2000000102.

[26] L.T. Skovgaard. “Applied regression analysis. 3rd edn. N. R. Draper and H. Smith, Wiley, New York, 1998. No. of pages: xvii+706. Price: £45. ISBN 0-471-17082-8”. In: *Statistics in Medicine* 19.22 (2000), pp. 3136–3139. ISSN: 0277-6715. DOI: 10.1002/1097-0258(20001130)19:22<3136::AID-SIM607>3.0.CO;2-Q. URL: http://doi.wiley.com/10.1002/1097-0258{\%}2820001130{\%}2919{\%}3A22{\%}3C3136{\%}3A{\%}3AAID-SIM607{\%}3E3.0.CO{\%}3B2-Q.

[27] M. S. Suchithra and Maya L. Pai. “Improving the prediction accuracy of soil nutrient classification by optimizing extreme learning machine parameters”. In: *Information Processing in Agriculture* xxxx (2019). ISSN: 22143173. DOI: 10.1016/j.inpa.2019.05.003. URL: https://doi.org/10.1016/j.inpa.2019.05.003.

[28] I. M. Tolentino et al. “Design, development, and evaluation of a simple wireless sensor network for indoor microclimate monitoring”. In: *IEEE Region 10 Annual International Conference, Proceedings/TENCON* (2010), pp. 2018–2023. DOI: 10.1109/TENCON.2010.5686565.

[29] A. K. Tripathy et al. “Data mining and wireless sensor network for agriculture pest/disease predictions”. In: *Proceedings of the 2011 World Congress on Information and Communication Technologies, WICT 2011* (2011), pp. 1229–1234. DOI: 10.1109/WICT.2011.6141424.

[30] Lev V. Utkin. “An imprecise extension of SVM-based machine learning models”. In: *Neurocomputing* 331 (2019), pp. 18–32. ISSN: 18728286. DOI: 10.1016/j.neucom.2018.11.053. URL: https://doi.org/10.1016/j.neucom.2018.11.053.

[31] Kumar Yelmarthi, Ahmed Abdelgawad, and Ahmed Khattab. “An architectural framework for low-power IoT applications”. In: *2016 28th International Conference on Microelectronics (ICM)*. Vol. 0. IEEE, 2016, pp. 373–376. ISBN: 978-1-5090-5721-4. DOI: 10.1109/ICM.2016.7847893. URL: http://ieeexplore.ieee.org/document/7847893/.

List of Figures