Spatial prediction of the infestation risks of winter wheat by the pathogens *Blumeria graminis* f. sp. *tritici* (Powdery mildew) and *Puccinia triticina* (Brown rust) in Schleswig-Holstein using machine learning techniques

Dissertation

zur Erlangung des Doktorgrades
an der Mathematisch-Naturwissenschaftlichen-Fakultät
der Christian-Albrechts-Universität zu Kiel

vorgelegt von

Wolfgang Berengar Hamer

Kiel, 2018

Erster Gutachter: Prof. Dr. Rainer Duttmann

Zweiter Gutachter: Prof. Dr. Joseph-Alexander Verreet

Tag der mündlichen Prüfung: 30. Mai 2018

Zum Druck genehmigt: 30. Mai 2018

gez. Prof. Dr. Natascha Oppelt, Dekanin

Acknowledgements

First of all, I would like to thank my supervisor Prof. Dr. Rainer Duttmann for the years of support and guidance. I would also like to thank Prof. Dr. Joseph-Alexander Verreet for introducing me to the world of phytopathology and for supporting me in it.

The same applies to the many members of his department, such as Dr. Christian Engel, Dr. Tim Birr, Dr. Holger Klink and the many doctoral students who have always supported me.

Likewise I would like to thank the Stiftung Schleswig-Holsteinische Landschaft for its kind funding.

I would also like to thank my friends and colleagues of the Geographical Institute, like all three Michaels, the two Katjas, Jan, Thea, Nicole, Nicolaus, Tim, Daniel, Philipp, Kilian and all the others for their always friendly and kind encouragement.

Last but not least, I would like to thank my siblings Ansgar and Magdalena and my brother-in-law Jesper as well as my parents Karoline and Hubert for their support.

Abstract

Wheat is one of the most important cereals in the world. Phytopathogens such as powdery mildew or brown rust can considerably reduce wheat yields. By treatment with fungicides, infections with these pathogens can be contained. It is of decisive importance to be informed about upcoming dangerous infestation events in real time to be able to respond to them. To define which infestation events are to be classified as yield-relevant, this work uses the damage threshold concept according to Klink (1997). This concept assumes that the exceedance of a 70 % threshold value for powdery mildew and of a 30 % threshold value for brown rust of infected plants in a field would threaten the yield of the complete stock of winter wheat and, thus, suggests the application of fungicides.

The main objective of this thesis is the spatial prediction of the probability of exceedance above this damage threshold. In order to achieve this goal, a concept was developed that regionalises the hourly weather data on a daily basis and subsequently uses these data as input parameters for predicting the pathogen-specific behaviour. Besides, the modelling concept uses supervised machine learning techniques to generate models based on these aggregated weather data, regionalised climate data and manually collected infestation data. The following learning methods are used to predict the occurrence of infestation spatially: *k*-Nearest Neighbor, Decision Trees, Boosted Decision Trees and Random Forests. The concept was examined iteratively using various evaluation methods, and thus the pathogen-specific performance of the models was tested concerning the prediction of the probability of infestation.

The combination of interpolation and machine learning techniques to predict the two pathogens presented first time within this thesis proved to be successful, as evidenced by an accuracy between 61 % and 84 % for powdery mildew and between 72 % and 84 % for brown rust. The values of the *Receiver Operating Characteristic Area Under the Curve* (ROC AUC) used as a quality measure for the successfully predicted exceedances, range from 0.64 to 0.86 (Powdery mildew) and from 0.67 to 0.91 (Brown rust) also prove this. Distinct differences in dependence of the used machine learning methods and the investigated phytopathogens were found. Thus, the *k*-Nearest Neighbor procedure proved to be the least promising in this context. The Random Forest method achieved lower overall accuracy, especially in the prediction of powdery mildew, but it achieved the highest sensitivity in each test. The Decision Tree approaches were most often able to achieve the highest values for the statistical measures such as accuracy, specificity, precision and ROC AUC, but at the expense of the sensitivity of the models produced. In the process, the difference in overall accuracy between the approaches used to predict brown rust infestations was smaller and these events were always more accurately predicted than yield-endangering powdery mildew events.

Zusammenfassung

Weizen ist eines der bedeutendsten Getreide der Welt. Phytopathogene wie der Echte Mehltau oder der Braunrost können den Weizenertrag deutlich reduzieren. Durch Behandlung mit Fungiziden können Befälle mit diesen Erregern eingedämmt werden. Von entsprechender Bedeutung ist es, bevorstehende gefährliche Befallsereignisse vorherzusagen, um auf diese reagieren zu können. Um festzulegen, welche Befallsereignisse als ertragsrelevant zu klassifizieren sind, verwendet diese Arbeit das Schadschwellenkonzept nach Klink (1997). Entsprechend dieses Konzeptes, gehen von mehr als 70 % mit dem Echten Mehltau oder von mehr als 30 % mit dem Braunrost befallenen Pflanzen eines Bestandes eine Gefährdung des Weizenertrages aus, welche mit Fungiziden abgewandt werden sollte.

Das Hauptziel dieser Arbeit ist die flächenhafte Vorhersage der Wahrscheinlichkeit einer Überschreitung dieser Schadschwelle. Um dieses Ziel zu erreichen, wurde ein Konzept entwickelt, welches die stündlichen Wetterdaten tagesaktuell interpoliert und diese mit dem erregerspezifischen Verhalten in Beziehung setzt. Darüber hinaus verwendet das Konzept überwachte maschinelle Lernverfahren, um Modelle zu generieren, die auf diesen oben genannten Wetterdaten, den regionalisierten langfristigen Klimadaten und manuell erfassten Befallsdaten basieren. Zur räumlichen Vorhersage der Befallsereignisse werden dabei folgende Lernverfahren eingesetzt: *k*-Nearest Neighbor, Decision Trees, Boosted Decision Trees und Random Forests. Das Konzept wurde unter Verwendung verschiedener Evaluierungsverfahren iterativ überprüft und somit die erregerspezifische Performanz der Modelle hinsichtlich der Vorhersage der Befallswahrscheinlichkeit getestet.

Diese erstmalig mit dieser Arbeit vorgestellte Kombination aus Interpolations- und maschineller Lernverfahren zur Vorhersage dieser Erreger erwies sich als erfolgreich, wie die Genauigkeiten zwischen 61 % und 84 % für den Echten Mehltau und zwischen 72 % und 84 % für den Braunrost belegen. Dies zeigen auch als Gütemaß für die erfolgreich vorhergesagten Überschreitungen die Werte der *Receiver Operating Characteristic Area Under the Curve* (ROC AUC) zwischen 0,64 und 0,86 (Echter Mehltau) und zwischen 0,67 und 0,91 (Braunrost). Allerdings ergeben sich im Vergleich der verwendeten maschinellen Lernverfahren deutliche Unterschiede. So erwies sich das *k*-Nearest Neighbor Verfahren in dem hier untersuchten Kontext als am wenigsten vielversprechend. Das Random Forest Verfahren erzielte zwar bei der Vorhersage der Wahrscheinlichkeit eines ertragsgefährdenden Ereignisses durch den Echten Mehltau eine geringere Gesamtgenauigkeit, aber dafür erzielte das Verfahren bei jedem Test die höchste Sensitivität. Die Decision Tree Ansätze konnten bei den verwendeten statistischen Kennzahlen Genauigkeit, Spezifizität, Präzision und ROC AUC zwar am häufigsten die höchsten Werte erreichen, dies allerdings auf Kosten der Sensitivität der erzeugten Modelle. Dabei war die Differenz in der Gesamtgenauigkeit zwischen den Verfahren bei der Vorhersage bevorstehender Braunrostereignisse geringer und die Vorhersage war immer genauer als die der ertragsgefährdenden Mehltauereignisse.

x

Contents

List of Figures

Glossary

accuracy number of true predictions divided by the total number of predictions (see chapter 2.6). 36, 41, 69–71, 77, 90–93, 98, 111, 113, 114, 117, 119, 123

ANN Artificial Neural Network. 40

BDT Boosted Decision Tree. 36, 38, 59–61, 69–75, 77, 80, 84, 86, 88, 90–96, 98, 102, 105, 108, 111, 113–115, 117, 118, 123

CDC Climate Data Center. 48, 49, 51, 55, 161

CTU Cumulative Thermal Unit. 58, 60, 64, 74, 79, 80, 97, 98, 100, 101, 111, 115, 118, 120

disease incidence percentage of diseased plants. 9–11, 14, 16, 18, 39, 50, 52–54, 56–60, 63, 64, 77, 81, 86, 88, 90, 94, 98, 102, 108, 111, 114, 116, 117, 146–151

disease severity percentage of infected plant tissue. 9–12, 14, 16–19, 40, 50, 116, 119

DT Decision Tree. 31, 33–36, 39, 59–61, 69–75, 77–80, 83, 86, 88, 90–96, 98–102, 104, 111, 113–115, 117–120, 122, 123, 125, 126

DTU Daily Thermal Unit. 57, 58, 60, 64, 74, 79, 115, 126

DWD German Meteorological Service. 47–49, 64, 77, 161

IDW Inverse distance weighting. 22–24, 29, 30, 53, 54, 56, 64

IPS Integrated plant protection system. 49, 50, 67, 145

k-NN *k*-Nearest Neighbor. 31, 32, 58–61, 64, 65, 69–75, 77, 80, 82, 86, 88, 90–96, 98, 102, 103, 108, 111, 113, 115, 117, 118, 120, 123

KED Kriging with External Drift. 22, 24, 27–30, 54–56, 64

LOOCV leave-one-out cross-validation. 29, 30, 62–64, 70–73, 92–95, 113, 114, 117

NG Northern Germany. 48, 49, 161

NRMSE root-mean-squared-error normalized by the standard deviation. 29, 56

OK Ordinary Kriging. 22, 24–27, 29, 30, 54, 56

OOB out-of-bag. 35, 61

precision proportion of predicted threshold exceedances classified correctly(see chapter 2.6). 41, 42, 69–71, 77, 90–93, 98, 111, 113, 117

RF Random Forest. 35–37, 59–61, 67, 69–77, 79–81, 85, 86, 88, 90–98, 101, 102, 106, 108, 111, 113–115, 117, 118, 120, 122, 123, 125

RK Regression Kriging. 24, 27, 30

RMSE root-mean-squared-error. 19, 29, 30, 56

ROC Receiver Operating Characteristic. 42, 70, 71, 92–94, 117, 123

ROC AUC ROC Area Under the Curve. 19, 41, 42, 59, 61, 69–75, 77, 91–96, 98, 111, 113–115, 117, 118, 120

sensitivity proportion of observed threshold exceedances classified correctly(see chapter 2.6). 41, 42, 69–71, 77, 91–93, 98, 111, 113, 114, 117, 123

SH Schleswig-Holstein. 43–45, 47–50, 52–54, 56, 58, 148–151

specificity proportion of observed threshold underruns classified correctly(see chapter 2.6). 41, 69–71, 77, 90–93, 98, 111, 113, 117, 123

UK Universal Kriging. 22, 24, 27, 29, 30, 54, 56

Chapter 1

Introduction

The world craves for wheat. This is shown by the Food and Agriculture Organization of the United Nations (2017c) statistics, which reveal that in 2014, one-third of the land used worldwide for growing cereals was utilized to produce wheat. In the same year, almost 730 million tonnes of wheat were produced worldwide. These numbers are reflected on a smaller scale in Schleswig-Holstein. In 2017, almost one-third of Schleswig-Holsteins farmland was used for the cultivation of winter wheat (Statistik-Nord, 2017a), resulting in a yield of 1,642,900 tonnes of winter wheat (Statistik-Nord, 2017b). It is not surprising that such extensive cultivation of cereals is so essential for the daily diet. After all, wheat is one of the oldest crops and thus belongs to our most important food suppliers for carbohydrates (Diepenbrock et al., 2005). Wheat was already cultivated in the 9th millennium BC in the Near East and it is still an integral part of many crop rotation cycles throughout the world today. It is mainly used for brewery, distillery, starch, forage and as bakery wheat (Rimbach et al., 2015). As vital as wheat is to humankind, it can be disastrous if the crop yield is reduced. In 1963, for example, the Soviet Union had to make a third of its gold reserves available after a disastrously poor harvest to purchase large quantities of cereals abroad (Gajdar and Paqué, 2015). In addition to bad harvests caused by extreme weather conditions, such as drought and fires in 2010, which caused a harvest collapse of 20 percent of Russian wheat production (Schmidt, 2010), various insects, weeds and phytopathogens can also endanger the harvest. Safeguarding the wheat harvest is of corresponding importance. Protection against phytopathogens, i. e. organisms that can cause plant diseases like nematodes, bacteria, viruses or fungi, plays a decisive role in this process. Fungi, in particular, can endanger crop yield. Curtis et al. (2002), for example, referred to powdery mildew, one of the fungi studied in this work, as *"one of the most important foliar diseases of wheat worldwide"*. According to Schlüter et al. (2009), the potential

damage from powdery mildew is usually more severe if no mix of varieties is used in wheat cultivation, as is currently common practice. In this way, the fungus could trigger entire disease epidemics within a concise period of time. Fungi such as powdery mildew or brown rust are distributed via air and infect the leaves of the plants. This simple dispersal mechanism makes it possible for the fungi to disseminate all over the world (Schlüter et al., 2009). If they can successfully infect the host plants and spread in the following days, this can lead to substantial yield losses as the productivity of infected plants decreases. However, as implied before, these problems are not limited to Schleswig-Holstein, Germany or Europe. As the Food and Agriculture Organization of the United Nations (2017d) commented on wheat rusts, these are a *"threat to wheat production worldwide"*. It is therefore not surprising that, according to the Food and Agriculture Organization of the United Nations (2017b), the average use of fungicides and bactericides between 2010 and 2015 was around 470,000 tonnes worldwide. Normalised over the agrarian area of the countries (Food and Agriculture Organization of the United Nations, 2017a), Germany ranks twenty-fifth with 0.64 tons of fungicides and bactericides per 1,000 hectares. These substances are often applied regularly to avoid infestations with an impact on earnings from the outset. Such a regular treatment can lead to the application of fungicides, although no increase in infestations was to be expected, for example, because the weather conditions would be unfavourable for the pathogen, which would, in turn, lead to higher costs for the farmer. Likewise, avoidable discharges of fungicides into the environment would occur. Informing farmers in time about upcoming or non-occurring infestation events could reduce these discharges as well as losses due to missing treatments.

Objectives The main objective of this work is to investigate whether and to which extent innovative methods can be used to support early warning of dangerous infestations of various pathogens in winter wheat. For this purpose the following sub-goals have been defined:

(a) Identification of the parameters influencing infestation and infestation risk

(b) Spatial interpolation of parameters derived from (a)

(c) Identification of machine learning methods suitable for the prediction of infestations measured in the field by Verreet et al. (2000)

(d) Development of a modelling concept combining the pathogens behaviour, the parameters created in (b) and the machine learning methods selected in (c)

(e) Spatio-temporal prediction of the infestation risk for powdery mildew and brown rust using the modelling concept (d)

(f) Assessment of the forecast quality of (e) using customary statistical measures

(g) Development of an on-demand web-based prediction system

Outline of the thesis Chapter 2 gives an overview of the "State of art". The current findings on powdery mildew, the known environmental influences on its development, its known effects on wheat as well as countermeasures and established models for its development are presented here. The following section gives the same overview of the brown rust. This is followed by a brief overview of the winter wheat host plant and the models describing its life cycle. Subsequently, the use of frequently applied machine learning methods and the methodology on which they are based are explained, as well as the most common deterministic and stochastic spatial interpolation methods.

In chapter 3 the study area is described in terms of its climatic conditions and its natural habitat structure. Then the data basis of this thesis is described including climate and weather data as well as observed infestations with various pathogens. This is followed by a description of the parts from which the approach developed here for generating a prediction model is composed and how these parts work in detail and the approach as a whole. It also explains how the different machine learning methods compared here are calibrated and how they are evaluated. The website, which was created in the context of this work to make the results of the prediction widely accessible, is also briefly described.

In chapter 4 the results of this thesis are presented. This chapter is divided into two parts. First, the results of the different methods used to predict the powdery mildew are shown. Subsequently, the presentation of the prediction of the brown rust follows. The results illustrate the spatial and temporal aspects of the prediction in a differentiated way.

Chapter 5 discusses the results presented before. As with the results, the pathogens are initially considered separately. Also, links are drawn between the state of research and the results. The chapter concludes with a synthesis of the variable and machine learning methods and the entire modelling approach itself.

Chapter 6 closes with the conclusion of the thesis. An overall view is given and possibilities are discussed as to how work can be continued with the approach.

Chapter 2

State of art

2.1 Life cycle of the powdery mildew

Powdery mildew is an obligate biotroph ectoparasite, that means it spreads at the surface of the plant and requires living plant tissue. Regarding taxonomy powdery mildew belongs to the family Erysiphacea of the order Erysiphales of the phylum Ascomycota (Schlüter et al., 2009). The phytopathogen considered in this thesis is tailored to wheat and triticale. Thus it is the subspecies (formae speciales) *tritici* of the species *Blumeria graminis*, which represents the powdery mildew adjusted to cereals (Hallmann et al., 2009). Subspecies as *B. graminis* f. sp. *hordei*, are adapted to barley, while *B. graminis* f. sp. *secalis* coevolved with rye and *B. graminis* f. sp. *avenae* with oats (Hallmann et al., 2009). The powdery mildew does not only attack cereals. More species have emerged adapted to different plant species like the *Beta vulgaris* (*Erysiphe betae*), the apple (*Podosphaera leucotricha*), the cucumber (*Sphaerotheca fuliginea*), the gooseberry (*Sphaerotheca mors-uvae*) and the grape vine (*Uncinula necator*) (Schlüter et al., 2009).

The polycyclic life cycle of *Blumeria graminis* f. sp. *tritici* (figure 2.1) is adjusted to the lifespan (figure 2.3) of its host the winter wheat *Triticum aestivum* (Hau and de Vallavieille-Pope, 2006). The infection occurs through sexual (asci) or asexual (conidia) spores (Schlüter et al., 2009). The wind carries these onto the leaf surface where the process of infection starts with the germination of the conidium (Hau and de Vallavieille-Pope, 2006). The spores stick to the surface and penetrate the cuticula using an enzyme which dissolves the cell wall (Schlüter et al., 2009). At this point, the ectoparasite develops a haustorium by which it is provided with nutrients from the host plant. If the parasite is metabolic dependent of the plant, the infection is completed (Hau and de Vallavieille-Pope, 2006). The following phase is called incubation, which

is the time between the infection and the appearance of the first symptoms, or latency period, describing the time between the infection and the start of the sporulation (Hau and de Vallavieille-Pope, 2006). The powdery mildew spreads on the leaf surface, penetrating more cells and developing additional haustoria (Schlüter et al., 2009). Following conidial chains grow from the parasite at the leaf surface. Such a chain consists of several connected conidia, which are thin-walled, colourless and single-celled spores (Yarwood, 1957). During the infectious phase, these spores are pinched off and spread by the wind (Schlüter et al., 2009; Hau and de Vallavieille-Pope, 2006). Most of the new infections caused by the conidia on the upper leaf levels of the plant stock originate from the lower levels of the same plant stock due to turbulent winds. Though a long-distant transport of the spores by wind is possible, this only accounts for a small proportion of the conidia (Audsley et al., 2005).

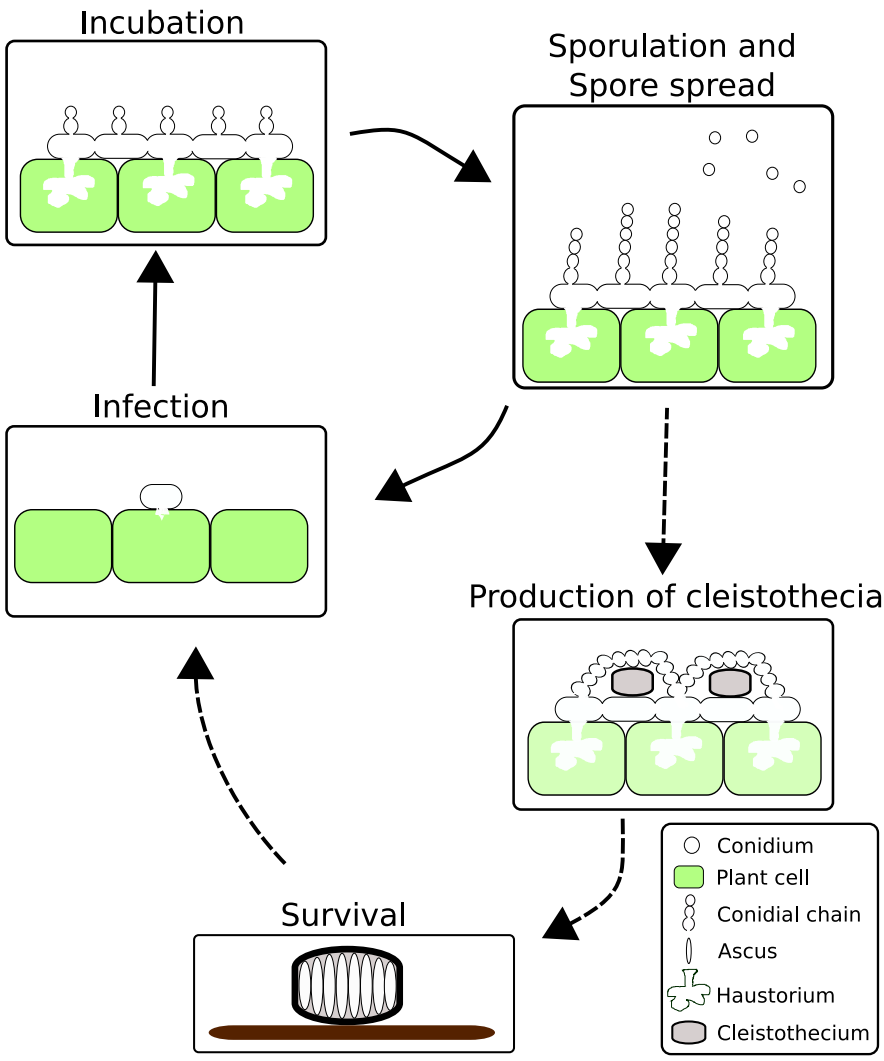


Figure 2.1: Life cycle of powdery mildew

Under adverse environmental or poor nutritional conditions, the powdery mildew will produce fruiting bodies (cleistothecia) which develop multiple sexual spores, referred to as asci (Agrios, 2005; Hau and de Vallavieille-Pope, 2006). This process can start from mid-May if in the year a hot and dry spring was dominant (Hau and de Vallavieille-Pope, 2006). Those cleistothecia help to survive the period with a lack of vegetation after the harvest of the wheat. Under favourable conditions, the powdery mildews asexual conidia can infect volunteer plants, which are self-sown and not planted by the farmer. The pathogen outlasts during additional cycles until it infects the lowest leaf sheath of the new sowed winter wheat (Hau and de Vallavieille-Pope, 2006). That way the powdery mildew hibernates as mycelium or as cleistothecium (Schlüter et al., 2009). Thereby the population of the mycelium grows until the lower cardinal temperature is reached (Hau and de Vallavieille-Pope, 2006).

2.1.1 Environmental influences on the life cycle of powdery mildew

The previously mentioned interactions of the powdery mildew are affected by environmental influences. The following paragraphs will show those effects during the different periods of the powdery mildews life cycle (see figure 2.1). It must be kept in mind, that the microclimate of the plant stock can buffer temperatures and modify wind movements depending on its development stage (Yarwood, 1957; Goudriaan and Van Laar, 1994). Another important influence is the genetic resistance of the host plant. Since it differs with the wheat variety and the pathogen's evolutionary stage of development, this aspect is omitted in the description of known environmental influences (Miedaner and Flath, 2007).

Environmental influences on the sporulation and spread The spread of the powdery mildew in a plant stock and between several plant stocks depends on the wind conditions. Strong gusts at the surface of the stock can vacuum air from the stock and cause turbulent upwardly airflows, by which the conidia are blown from lower to upper leaf levels of the plant stock (Cao et al., 2012). Cao et al. (2012) specifies a necessary wind speed of 0.6 to $2\,\mathrm{m\,s^{-1}}$ and notes that wind speeds above $0.5\,\mathrm{m\,s^{-1}}$ are seldom reached. Furthermore, Adams et al. (1986) observed enhanced spore discharge at a sudden sink of relative humidity at a constant wind speed of $0.5\,\mathrm{m\,s^{-1}}$ whereas a constant high relative humidity lowered spore release. Following the change in humidity, the spore spread reveals a diurnal periodicity (Adams et al., 1986). Just as humidity varies by time of day, so does the temperature for which

Hammett and Manners (1971) detected an influence on the spore dispersal. This periodicity is less noticeable on cloudy days because the cloud cover smothers the daily variations of the temperature, the relative humidity and the wind action in the stock by the dimmed solar radiation (Adams et al., 1986). Audsley et al. (2005) and Friedrich (1995a) found an influence of the temperature on the sporulation as well. Besides Sreeramulu (1964) and Last (1955) noticed a decrease in spore concentration per volume air during a precipitation event and the following 3 to 5 days. There is no information about the distance that conidia of powdery mildew can be transported. Hau and de Vallavieille-Pope (2006), however, describe an empirical exponential model to calculate a disperse gradient. The average distance traversed by the spores is less than 10 m. Also, several articles prove that the conidia carried by wind over a long distance only accounts for a small proportion of all transported conidia (Audsley et al., 2005; Cao et al., 2012).

Environmental influences on the production of cleistothecia According to Hau and de Vallavieille-Pope (2006) increase in temperature and starting senescence of the host plant trigger the powdery mildews production of cleistothecia. Knowledge about the influence of the environment on the production of cleistothecia is widely missing. Agrios (2005) and Sinha et al. (2004) only mentioned the production of cleistothecia starts under unfavourable environmental or nutritional conditions.

Environmental influences on the survival of cleistothecia Under dry conditions, the cleistothecia can survive without a host plant over a longer period, but the production of ascospores by the cleistothecia is reduced in comparison to moister conditions, which reduces the lifespan of the conidia (Liu et al., 2012). Thereby the survival probability is higher if the conidia are attached to plant residuals or straw and lower, if they are situated on the bare ground and even lower if they are located inside the soil (Liu et al., 2012).

Environmental influences on the infection Temperature mainly influences infection. Although assessments of the optimum temperature differ, they are close to each another. Paulus (1990) found an increased infection rate at the temperature range between 15 and 25 °C and an optimum at 20 °C. Beest et al. (2008) detected an optimum range between 10 and 22 °C and a decrease at temperatures above 25 °C. Eckhardt et al. (1984) however made up a possible range from 0 to 30 °C for infection and an optimum range between 12 and 24 °C with the germination efficiency in form of a parable. Also, the relative humidity has a considerable influence on the infection.

A humidity value close to 100 % supports the pathogen and results in a quick infection (Beest et al., 2008; Hau and de Vallavieille-Pope, 2006; Paulus, 1990). According to Paulus (1990) and Schlüter et al. (2009) at least a humidity of 80 % is necessary for the infection, while Hau and de Vallavieille-Pope (2006) and Yarwood (1957) stated that infection can happen close to 0 %. Yarwood (1957) explains this with the high water content of the conidias surface which is unique among air-borne fungi. According to Merchán and Kranz (1986) the influence of precipitation on the infection is negligible, except for heavy rain, which can wash off the conidia from the plant surfaces. Moreover, Eckhardt et al. (1984) found out, the density of the spores can affect the host. The higher the density of the conidia, the lower is the proportion of successful infections. Eckhardt et al. (1984) also detected a lower efficiency of infection on days with a high amount of incoming solar radiation in comparison to cloudier days. This corresponds to the foregone studies of Yarwood (1957) who saw the benefit of a lower temperature in a shaded area.

Environmental influences on the incubation After infection, powdery mildew is metabolic dependent on the host plant and therefore passively dependent on the host's environment. Nevertheless, the temperature still has a major influence on the development of the pathogen. Correspondingly the time of incubation varies depending on the temperature between 6 and 14 days (Friedrich, 1995b). Friedrich (1995b) found out, that the incubation could happen at temperatures between -4 and 26.5 °C whereas the studies of Kocourek and Vǎchet (1984) resulted in a temperature range between 0 and 24.5 °C. Hence their developed equations, displayed in figure 2.2 assuming a constant temperature, differ from one another. Both of them used empirical data but for different varieties of wheat. Friedrich (1995b) employs the wheat variety 'Kanzler' and Kocourek and Vǎchet (1984) used 'Blue boy', which could explain the mixed results.

2.1.2 Influence on yield and countermeasures

Infestation of powdery mildew on winter wheat reduces photosynthesis due to the destruction of photosynthetic plant tissue and the coverage of the leaf area (Schlüter et al., 2009). According to Käsbohrer et al. (1988) already a percentage of more than 1 to 2 % of infected plant tissue, can be relevant to the yield. Like Käsbohrer et al. (1988), this proportion of infected leaf surface is referred to as disease severity in this work. The number of infected plants in the stock is used as a further characteristic of the infestation. It is referred to as disease incidence. These two terms, as defined by

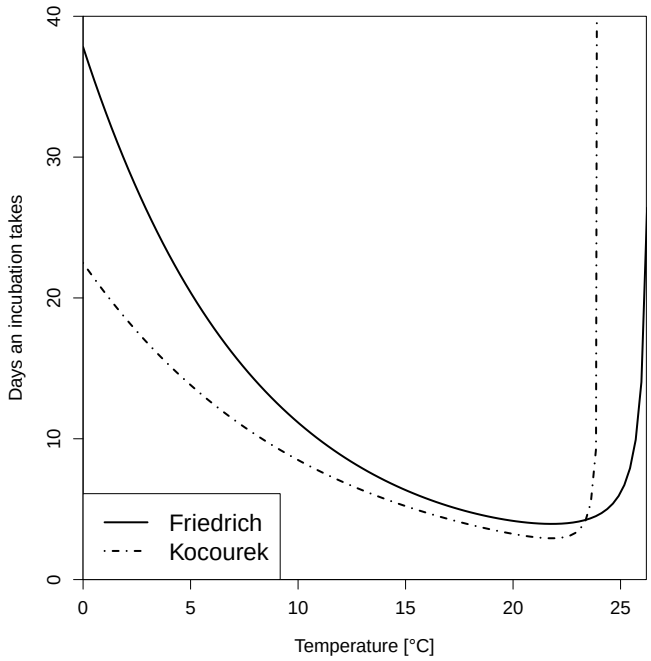


Figure 2.2: Incubation days depending on the temperature according to Friedrich (1995b) and Kocourek and Vǎchet (1984)

Madden and Hughes (1999), are among the most common in the description of infestation events. The disease severity detected by Käsbohrer et al. (1988) corresponds to a disease incidence of 60 to 80 %. Depending on the plant variety Käsbohrer et al. (1988) found a yield improvement of 5.8 to 11 % if the treatment with the fungicides Bayleton and Bayfidan was coordinated with the disease severity threshold in comparison with an unattended variant. Klink (1997) also detected a disease incidence control threshold of 70 %. However, Rabbinge et al. (1985) detected a yield loss of 10 % at a disease severity of 4 % due to the plants reduced assimilation and transpiration rates.

As Käsbohrer et al. (1988) implied, the fungicidal treatment of the plants is a common way to reduce the infestation with powdery mildew. Vechet et al. (2009) worked out that both the synthetic and biological resistance inducers protect the wheat. Aside from fungicide treatment the cultivation of resistant plant varieties and the optimal incorporation of crop residue into soil help to reduce the infestation of powdery mildew (Schlüter et al., 2009). Liu et al. (2012) detected the lowest survival probability of conidia found inside the soil.

The high adaptability of the pathogen which overcomes the plant's resistance makes it difficult to develop fungicides and resistant plant varieties (Miedaner and Flath, 2007; Schlüter et al., 2009; Agrios, 2005). The development of susceptibility to powdery mildew of the wheat varieties most recommended by LKSH (2015) for Schleswig-

Holstein in the last years (table 2.1) illustrates the pathogen's adaptability. More than one-third of these varieties show an increase of the susceptibility to powdery mildew whereas only one variety shows a decrease. In addition, of the 259 wheat varieties examined by the Bundessortenamt (2017) since 2002 28 % show an increase and 2 % show a decrease in susceptibility.

Table 2.1: Susceptibility of certain wheat varieties to powdery mildew. High values represent a high susceptibility. Based on Bundessortenamt (2017)

Wheat varieties	2008	2009	2010	2011	2012	2013	2014	2015	2016
Julius	3	3	3	3	4	4	4	4	4
Smaragd	-	5	5	4	4	4	4	4	4
Primus	-	1	1	1	1	1	1	1	2
Tobak	-	-	2	2	2	2	2	2	2
Inspiration	3	3	3	3	3	3	3	3	3
Desamo	-	-	-	-	3	3	4	4	4
Elixer	-	-	-	2	2	2	2	2	3
Lear	-	2	2	2	2	2	2	2	2
Anapolis	-	-	-	-	-	1	2	2	2
JB Asano	3	3	3	3	3	3	3	3	3
Tuareg	1	1	2	2	2	2	2	2	-
Potenzial	2	2	2	2	3	3	3	3	3
Tabasco	1	1	1	1	1	1	1	1	1
Orcas	-	3	3	3	3	3	3	3	3

2.1.3 Existing models to simulate powdery mildew infestations

There are already existing some tools to model one or several steps of the powdery mildews life cycle to reduce the impact of infestations with *Blumeria graminis* f. sp. *tritici*. A simple approach has already been introduced with the disease severity threshold of Käsbohrer et al. (1988). According to this concept, after an empirically determined number of infected plants, it can be assumed that the additional expenditure for treatment with fungicides pays off. Klink (1997) identified a disease incidence threshold of 70 % to avoid yield relevant damage. In other words, if 70 % of the plants are infested, treatment with fungicides is justified by the otherwise occurring loss of yield. Similar to the threshold method but more complex are systems like PC-Plant Protection (Secher et al., 1995). Following a short description of further developed models, simulating the temporal but not the spatial behaviour of powdery mildew is provided. Table 2.2 provides an overview of these and the models used to calculate brown rust and wheat growth.

Modell according to Hau (1985) Hau (1985) developed the local model GEMETA for the simulation of the barley powdery mildew (*Blumeria graminis* f. sp. *hordei*). It is based on the EPIGRAM model of Aust et al. (1983). The model includes the development of the barley by leaf area development. The model requires hourly temperature values and daily precipitation as input. For the case, that only daily maximum and minimum temperature data are available Hau (1985) suggested calculating the hourly data with a sine function. Furthermore Hau (1985) converted these air temperature values to leaf temperature based on formula 2.1.

$$T_{leaf} = T_{air} * 1.267 - 4.0 \tag{2.1}$$

The model follows the infection chain of the pathogen. It starts with the infection, goes on with the incubation, then calculates the growth of the lesion and finishes with the sporulation (Hau, 1985). The infections efficiency calculated in the model depends on the temperature and limited by heavy precipitation. Also, the barleys age-related resistance against the powdery mildew and the conidias reduced germination capacity due to high temperatures during the conidia production and the age of the producing conidia limit the infection process (Hau, 1985). The model's incubations duration is only dependent on the temperature similar to the functions displayed in figure 2.2. The growth of the pathogens lesion on the plant is calculated by a fifth-degree polynomial depending on the temperature which results in the area covered by *B. graminis* (Hau, 1985). Also, the growth can be reduced due to the amount of precipitation. Moreover, the barleys age-related resistance has been taken into account in this step. Therefore the higher leaf levels have lower growth rates (Hau, 1985). The intensity and duration of the sporulation depend on the temperature in the GEMETA model. The age-related resistance is included the same way it has been in the growth rate (Hau, 1985). Because Hau (1985) had no wind data, it was assumed, that there was always enough wind to release the spores. In the model, the liberation of spores was only reduced by precipitation. The model quality was not specified by Hau (1985), due to unreliable disease severity measures.

Modell according to Friedrich (1994) Friedrich (1994) developed a local model which describes the infection chain of *Blumeria graminis* f. sp. *tritici* at the wheat variety 'Kanzler'. The model also follows the infection chain of the pathogen. It consists of sporulation, flight and landing of the conidia, infection and incubation. The conidia production in the model follows a function of temperature and relative humidity. Besides, the release of conidia also depends on the wind speed (Friedrich,

1995a). The infection follows the landing of the conidia and is calculated with two terms depending on the temperature (Friedrich, 1994, 1995c). With these equations, an hourly development value of the infection is computed, which adds up for every hour until an exceedance of the value "1" indicates a successful infection. This development can be disrupted in the model if the survival probability is too low (Friedrich, 1994). Such an abortion occurs at too low or too high temperature, too low relative humidity, too much rainfall and too high wind speed. Following the incubation is calculated similarly to the infection (Friedrich, 1994, 1995b). Only the incubation cannot be aborted by the survival probability since the pathogen is already parasitically dependent on the host. Friedrich (1994) validated only separate processes of the model. There is no validation of the models prognosis of infection probability.

Model according to Jensen and Jensen (1996) Jensen and Jensen (1996) developed the decision support system MIDAS for the management of powdery mildew in winter wheat. Deviating from the former mentioned models MIDAS is not an analytic or deterministic model (according to Hau (1985)) but a decision model. The system of Jensen and Jensen (1996) calculates possible effects of the host's treatment, while the development of the pathogen is computed with a growth rate, affected by crop density, the weather conditions and the protection against the infection. Temperature, humidity and wind are specified as weather conditions in the MIDAS model. In the subsystems, Jensen and Jensen (1996) use deterministic model. The model uses thermal weeks to show the temporal progression during a cultivation period. Each thermal week, which results from the sums of daily mean air temperature, represents one timestep. The calculations of the host's treatment and the development of the pathogen are made for every time step. The result is a disease-level where prediction comes close to the observation. Jensen and Jensen (1996) also planned to implement diseased plants as an output variable, but this has not yet been implemented.

Model according to Bruns (1996) Bruns (1996) developed the forecasting model MEVA-PLUS on the basis of the model MEVA. MEVA-PLUS aims at the prognosis of the damage due to powdery mildew on winter wheat. To include the influence of the weather conditions Bruns (1996) referred to the GEMETA model of Hau (1985). Correspondingly only the daily maximum and minimum of the temperature and the precipitation sum represent the weather in the MEVA-PLUS model. Although the GEMETA model was created to analyse the impact of powdery mildew on barley Bruns (1996) adopt the assumptions of the host's age-related resistance to MEVA-PLUS. On the basis of necessary monitoring of the infestation at the beginning of a model run,

the implemented GEMETA model calculates a prognosis of the infestation (Bruns, 1996). With this prognosis, the MEVA-PLUS model calculates the possible damage due to the infestation. For this purpose Bruns (1996) developed different crop-loss-functions. It is the responsibility of the user to estimate, which loss is acceptable. The validation of MEVA-PLUS is carried out by a comparison of the observed and predicted disease severity values in the different plant development stages. When applying this model approach to predict the disease severity using monitoring at the beginning of the season, (Bruns, 1996) achieved coefficients of determination (R^2) between 0.00002 and 0.796. The coefficients of determination decrease with a longer time distance to the initial monitoring.

Model according to Rossi and Giosuè (2003) Rossi and Giosuè (2003) developed a model to calculate the disease incidence of powdery mildew on winter wheat. As most of the models mentioned before, the model of Rossi and Giosuè (2003) follows the infection chain. Like the model of Friedrich (1994) the model requires temperature, vapour pressure deficit, rainfall and wind data. The whole model follows a logistic function to describe the disease progress expressed in the following formula:

$$PLA_{Lj} = PLA_{L(j-1)} + (DPLA_{Lj}) * CV * LL_L \tag{2.2}$$

where PLA_{Lj} is the infected leaf area, $PLA_{L(j-1)}$ is the infected leaf area of the foregone timestep, $DPLA_{Lj}$ is the daily increase of infection, CV is a parameter for the wheat variety and LL_L is the leaf layer (Rossi and Giosuè, 2003). The daily increase of infection is calculated by a submodel, which follows the infection chain, including the growth of the fungal colonies on the leaves, sporulation and new infections. In doing so, the model of Friedrich (1994) is implemented as well as the WHEat GROwth SIMulation (WHEGROSIM) of Rossi et al. (1997). In the validation, (Rossi and Giosuè, 2003) concluded the model would produce a satisfactory simulation with a R^2 of 0.89 in the comparison of observed and simulated powdery mildew disease incidences.

Model according to Willocquet et al. (2008) Willocquet et al. (2008) developed the mechanistic simulation model WHEATPEST to calculate the effects of weeds, aphids, viruses and different fungal infections on winter wheat. WHEATPEST requires daily temperature and radiation as well as drivers for production situation and for injury profiles (Willocquet et al., 2008). An injury profile is generated from the combined effects of various pests such as pathogens, insects and weeds. Like the model InfoCrop of Aggarwal et al. (2006), such crop yield assessment simulation models often require

the actual infestation events to calculate the potential crop damage. The model's outputs are the development stage of the crop, the dry biomass, the leaf area index and the expected yield. Willocquet et al. (2008) did not validate the simulated yields by comparing them with observed values.

2.2 Life cycle of the brown rust

The wheat brown rust (*Puccinia triticina* formerly known as *Puccinia recondita* f.sp. *triticina*) is a biotroph ectoparasite of the family Pucciniaceae of the order Uredinales of the phylum Basidiomycota (Hallmann et al., 2009). Other subspecies of brown rust are adapted to rye (*Puccinia recondita* f.sp. *recondita*) and barley (*Puccinia hordei*). In addition, there are numerous other types of rust, such as the wheat yellow rust (*Puccinia striiformis* f.sp. *tritici*) and the oat crown rust (*Puccinia coronata*) (Schlüter et al., 2009). Like the powdery mildew, the brown rust requires living plant material to fulfill its polycyclic life cycle (Hau and de Vallavieille-Pope, 2006). Other than the powdery mildew rusts can develop five different types of spores including urediniospores, teliospores and basidiospores on the wheat and spermatia (formally known as pycniospores) and aeciospores for volunteer plants. With five different types of spores, brown rust is referred to as macrocyclic rust. Most diseases are caused by urediniospores (Hau and de Vallavieille-Pope, 2006; Schlüter et al., 2009).

The aeciospores are released from the volunteer plants by wind and land on the host plant wheat (Schlüter et al., 2009) where the spore germinates and infects the plant through the leaf stomata (Hallmann et al., 2009). Different from the powdery mildew the brown rusts mycelium is not formed exophytic on the outside of the leaf but endophytic intercellular within the leaf (Hau and de Vallavieille-Pope, 2006). After successful infection and incubation of the host, the brown rust produces urediniospores which can infect more wheat plants. Similar to the cleistothecia of powdery mildew brown rust produces teliospores in the plant's leaf at the end of the host's life cycle (Hau and de Vallavieille-Pope, 2006). These spores are released from the bottom of the leaf and serve as a sexual overwintering stage of the development of the pathogen. After winter the teliospores form the basidium which again releases the basidiospores (Agrios, 2005). These spores can infect the rusts intermediate host the "Common meadow-rue" (*Thalictrum flavum*) and form spermatogonia which produce spermatia (Schlüter et al., 2009). The spermatia can fertilise receptive hyphae of other infected meadow-rue. Following on the bottom of the leaves aeciospores are produced by the brown rust (Schlüter et al., 2009).

Table 2.2: Overview of existing models to simulate fungal infestations and wheat development

Model name	Prediction of	Input parameters	Author
GEMETA	Disease severity of Barley powdery mildew	Hourly temperature Daily precipitation Start disease severity	Hau (1985)
–	Development of Wheat powdery mildew	Hourly temperature Hourly humidity Hourly precipitation Hourly windspeed	Friedrich (1994)
MIDAS	Value of yield Cost of treatment Yield loss induced by Wheat powdery mildew	Hourly temperature Hourly humidity Hourly windspeed Start disease incidence	Jensen and Jensen (1996)
MEVA-PLUS	Disease severity of Wheat powdery mildew	Daily max. temperature Daily min. temperature Daily precipitation Start disease severity	Bruns (1996)
–	Disease incidence of Wheat powdery mildew	Hourly temperature Hourly vapour pressure deficit Hourly precipitation Hourly wind data	Rossi and Giosuè (2003)
RUSTDEP	Disease severity of Wheat brown rust	Hourly temperature Hourly precipitation Hourly humidity Start disease severity	Rossi et al. (1997)
–	Disease severity of Wheat brown rust	Daily temperature Daily precipitation Daily radiation Daily evapotranspiration	Gouache et al. (2015)
WHEATPEST	Dev. stage of wheat Dry biomass LAI Expected yield	Daily temperature Daily radiation "Production parameters" "Injury profiles"	Willocquet et al. (2008)
CERES	Growth of winter wheat	Daily temperature Day of year Latitude	Ritchie et al. (1988)
Onto	Development of wheat, barley and rape	Daily temperature Daily humidity Relative soil moisture Day length	Wernecke and Claus (1996)
SIMONTO	BBCH of winter crops	CERES inputs Onto inputs	Roßberg et al. (2005)
AFRCWHEAT1 AFRCWHEAT2	Development of leaves Photosynthetic processes Dry matter production	Faily temperature Daily solar radiation Sowing date Latitude	Porter (1984) Weir et al. (1984) Porter (1993)
APSIM	Yield estimation Influence on Soil processes	Environment data (weather and soil) Management data Plants genotype	McCown et al. (1996)

2.2.1 Environmental influences on the life cycle of brown rust

Like the life cycle of the powdery mildew, the life cycle of the brown rust is weather dependent as well. The wind speed mainly influences the sporulation and spread of spores. Following Geagea et al. (1997) a wind speed of at least 1.3 to 1.8 $\mathrm{m\,s^{-1}}$ is needed to release the pathogens spores. Sache (2000) additionally mentioned the influence of rain on the spore release. Either by rain-splash which carries the spores within the water or by dry-dispersal which moves the leaves by raindrops without contact to the spores. According to Hau and de Vallavieille-Pope (2006), rusts are not able to survive a dry period during the germination and penetration since the spores of the brown rust need to be hydrated for the infection of new plants. This is supported by Simkin and Wheeler (1974) who found out that no spores of *Puccinia hordei* germinated at humidity below 100 %. They also detected a temperature range of 15 to 25 °C for successful germination. The duration of infection itself also depends on the temperature. Roelfs et al. (1992) found successful infections if dew is available for 3 h for temperatures around 20 °C and durations of 12 h for temperatures around 10 °C. Additional Roelfs et al. (1992) defined a incubation time of 7 to 10 d.

2.2.2 Influence on yield and countermeasures

Similar to the infection with powdery mildew, a brown rust infestation can reduce the crop yield if no countermeasures are taken. Hallmann et al. (2009) simply described a yield loss of 10 to 20 % for a severe infestation. However, the susceptibility of the host to brown rust strongly depends on the wheat variety observed. Herrera-Foessel et al. (2006) studied 30 different durum wheat genotypes and compared plots with fungicidal treatment with untreated plots. They detected an average loss in grain yield by 4.8 to 51.1 % for the untreated plots depending on the host's genotype. The protected plots were treated with the fungicide tebuconazole three times to avoid infection and spread of brown rust. Huerta-Espino et al. (2011) also describe yield losses higher than 50 % if the infection occurs in an early development stage of the host. The high damage depends largely on the early occurrence of the pathogen. An infection on the highest leaf of the wheat with a disease severity of 60 to 70 % would lead to a yield loss higher than 30 % if the disease severity has this level in the BBCH stage 51 (compare Hack et al. (1992)) and only 7 % if the level is reached in stage 85 (Huerta-Espino et al., 2011).

2.2.3 Existing models to simulate brown rust infestations

A number of methods have been developed predicting the behaviour of brown rust. The damage threshold concept, described in chapter 2.1.3, is also used for brown rust. For *Puccinia triticina* Klink (1997) identified a disease incidence threshold of 30 % which should be considered to avoid yield relevant damage. To predict the disease severity also for brown rust models were constructed which are outlined below and in table 2.2.

Model according to Rossi et al. (1997) Rossi et al. (1997) developed the RUSTDEP (RUST Development of EPidemics) model to predict the disease severity of brown rust on winter wheat. The mechanistic model requires the input of hourly temperature, precipitation and humidity data as well as the first observed disease severity. In combination with the WHEGROSIM (WHEat GROwth SIMulation) model RUSTDEP calculates the potential spread of the observed disease severity on individual wheat leaves. Rossi et al. (1997) used the holdout method to validate their model which resulted in 80 % of the simulated disease severity to fall into the confidence interval of the observed data.

Model according to Willocquet et al. (2008) The mechanistic WHEATPEST model developed by Willocquet et al. (2008) does not only simulate the influence of powdery mildew on wheat but also the effects of brown rust. Willocquet et al. (2008) use the observed disease severity and the leaf area index to calculate the pathogen's influence on the wheat plants development. The leaf area index itself is reduced by the observed disease severity of different pathogens including brown rust and powdery mildew.

Model according to Gouache et al. (2015) Gouache et al. (2015) studied the impact of climatic variables on the disease severity caused by brown rust on wheat plants in France using observations from untreated plots of the years 1980 to 2011. They calculated logistic regressions between the climatic parameters derived from the temperature, precipitation, solar radiation and evapotranspiration by the work of Thépot and Gouache (2009). The selection of the climatic parameters followed the analysis of a window pane algorithm. The window algorithm was developed by Coakley et al. (1988) to forecast stripe rust (*Puccinia striiformis*) disease severity on winter wheat also using weather parameters. The window pane algorithm automatically selects weather variables of different overlapping time windows and searches for multiple linear regressions

of the variables and the pathogens disease severity. With this approach Gouache et al. (2015) received an root-mean-squared-error (RMSE) of 0.29 representing 22.4 % of occurred disease severity values and a ROC Area Under the Curve (ROC AUC) (page 42) value of 0.85.

2.3 Life cycle of the winter wheat

The winter wheat (*Triticum aestivum* L.) is a grass of the family *Poaceae* of the genus wheat (*Triticum* L.) (Miedaner, 2014). Depending on the location, the soil, the previous crop and the weather it is usually sowed between late September and early November with a seed rate of 250 to 450 seeds per m^2 (Hanus et al., 2008; Diepenbrock et al., 2005). The germination usually starts after seed dormancy's end, when the minimum germination temperature of 2 to 4 °C is exceeded (Diepenbrock et al., 2005). Temperatures of 6 to 8 °C allow quick germination until temperatures of 1 to 5 °C over a period of 40 to 80 days enable the plant's vernalisation. The wheat's frost resistance is around -15 to −20 °C. After the vernalisation, the bolting process starts with an optimal temperature of 15 to 20 °C (Diepenbrock et al., 2005). At this time in addition to the temperature, the availability of water is significant for the count of secondary sprouts and ears (Hanus et al., 2008). Most important for the development of winter wheat therefore are temperature and precipitation (Diepenbrock et al., 2005; Hanus et al., 2008). Harvest is usually carried out between late July and early August. The sowing usually follows until mid-October. Typically wheat yield is about 10 t/ha in Schleswig-Holstein (Langensiepen et al., 2008). Figure 2.3 shows the growth cycle of winter wheat. The depiction is schematic, since the development is strongly dependent on meteorological influences, especially on temperature. For example, the BBCH stage of wheat varies in May in Schleswig-Holstein from 30 (bolting begin) to 59 (ear pushing end) and in June from 37 (last leaf appearance) to 83 (early dough-ripe).

2.3.1 Existing models to simulate the growth of winter wheat

To simulate growth and yield of various crops as wheat, a number of models have been developed. The approaches have already been mentioned in table 2.2, below the most common simulation models are described briefly.

The CERES-wheat model CERES-wheat is a growth simulation model for winter wheat developed by Ritchie et al. (1988). The model simulates the growth of wheat in nine stages by an accumulation of daily thermal time (Hodges and Ritchie, 1990).

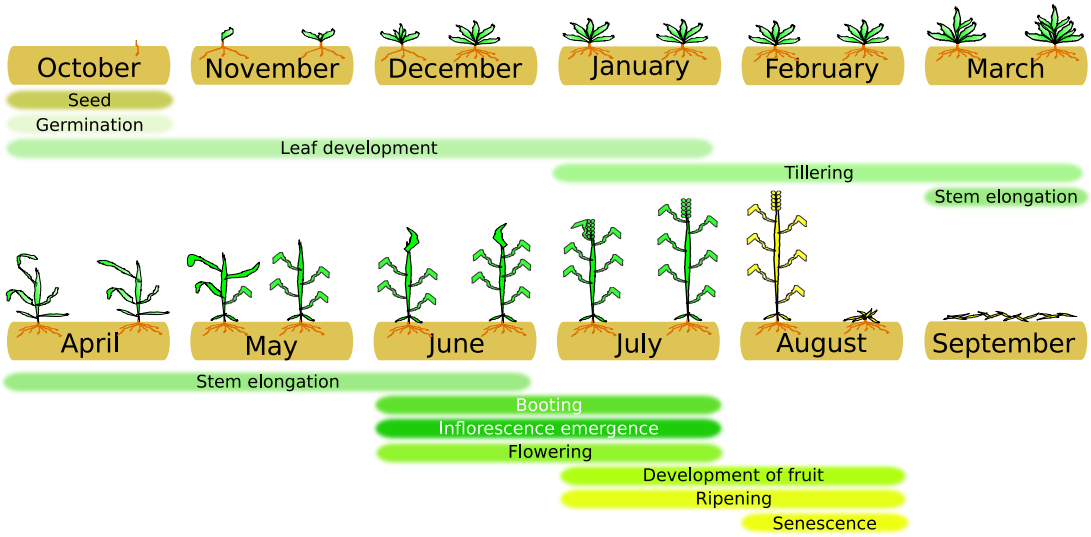


Figure 2.3: Schematic development of winter wheat throughout the year

Also, the vernalisation is calculated by the CERES model by the cumulation of daily vernalisation values during the growth stages 9 and 1. Moreover, the photoperiod influences the model. It is calculated by the day of the year and the latitude of the study area (Hodges and Ritchie, 1990). The CERES model was developed further and became part of the DSSAT (Decision Support System for Agrotechnology Transfer) cropping system model outlined by Jones et al. (2003). The DSSAT model consists of different independent cooperating programs and aims at the simulation of crop growth, development and yield, considering the meteorological conditions as well as genetics and pests and even soil water, carbon and nitrogen (Jones et al., 2003). Amongst the growth of wheat, this model enables the simulation of soybean, peanuts, dry bean, tomato, maize, potato, rice and other crops (Jones et al., 2003).

Langensiepen et al. (2008) tested the CERES-wheat module of the DSSAT model in Schleswig-Holstein.They could show that the calculation of the phenological events was satisfactory, but Langensiepen et al. (2008) considered the model to be not applicable as an optimization tool for nitrogen management.

The Onto model Wernecke and Claus (1996) developed the ontogenesis model Onto, which models the development of wheat, barley and rape. In addition to the development expressed as DC-value "Onto" enables the calculation of the vernalisation and the grains water content. The required input variables of "Onto" are the daily air temperature and humidity, the relative soil moisture and the day length (Wernecke and Claus, 1996). Similar to the CERES-wheat model "Onto" predicts the speed of ontogenesis proportional to the summed up daily temperature values.

On the basis of the Onto and the CERES-wheat model Roßberg et al. (2005) developed SIMONTO. It enables the user to calculate the development of winter crops in the more common BBCH-scale described by Hack et al. (1992). For this purpose Roßberg et al. (2005) generated a function to translate the linear development to the non-linear BBCH-scale. Furthermore, SIMONTO requires inputs of temperature and latitude and foregoes soil moisture. Roßberg et al. (2005) admit, that this simplification can deteriorate the simulation under dry conditions. Altogether Roßberg et al. (2005) found out, that 70 % of the simulations get acceptable to good results. According to this, SIMONTO is in use by the Germany-wide provider of decision-making aids Informationssystem Integrierte Pflanzenproduktion (ISIP) e.V..

The AFRCWHEAT models AFRCWHEAT1 is a mechanistic wheat simulation model developed by Porter (1984) and Weir et al. (1984). The model does not include water and nutrient limitations. As a mechanistic model, it is composed of modules to calculate the phenology, the development of leaves, the photosynthetic processes and the dry matter production (Jamieson et al., 1991). The required inputs are the daily temperature and solar radiation, the sowing date and the location's latitude. According to Jamieson et al. (1991) ARCWHEAT1 was good at simulating the growth and development of wheat, but less well at predicting the final grain yield. Porter (1993) developed AFRCWHEAT2 from AFRCWHEAT1 by combining it with the functional soil water model SLIM (Solute Leaching Intermediate Model) of Addiscott (1977), which can simulate the distribution of water and nitrogen within the soil. Ewert et al. (1996) tested AFRCWHEAT2 for the development of main stem and tillers in winter wheat in Rostock over several years and achieved satisfactory results.

The APSIM model McCown et al. (1996) developed the soil oriented modelling framework APSIM (Agricultural Production Systems Simulator) on the basis of the soil model PERFECT (Littleboy et al., 1992) and the crop model AUSIM (McCown et al., 1994). APSIM aims at yield estimation as well as predicting the influence of farming practices on soil processes (Keating et al., 2003). Due to the framework environment, the data required as input variables depend on the aim of the modeling and the used modules of APSIM (Keating et al., 2003). As Holzworth et al. (2014) pointed out, the APSIM framework has grown and evolved over the years. The framework allows an extensive adaptation to the user's demands including the input of the environment data (e.g., weather and soil data), the management data (e.g., sowing date, plant density, ...) and also information about the plant's genotype (Holzworth et al., 2014).

2.4 Regionalisation methods

Regionalisation methods cover extrapolation and interpolation methods. These methods are used to predict values of point data at every point in a specific environment. The regionalisation between the observed points is called interpolation whereas the prediction beyond the observed points is referred to as extrapolation. Li and Heap (2014) mention more than 40 spatial prediction techniques, of which the most common ones are presented below. Table 2.3 provides an overview of the methods covered and their possible applications.

Table 2.3: Overview of interpolation methods used on meteorological data

Classification	Method	Applied on	Used by
deterministic	Thiessen polygon	Precipitation	Thiessen (1911)
	Inverse distance weighting (IDW)	Wind speed	Luo et al. (2008)
		Precipitation	Wagner et al. (2012) Piazza et al. (2011) Borges et al. (2016)
probabilistic	Ordinary Kriging (OK)	Air temperature	Monestiez et al. (2001) Benavides et al. (2007)
		Air humidity	Nguyen et al. (2015)
		Wind speed	Luo et al. (2008) Eguía et al. (2016)
		Precipitation	Wagner et al. (2012) Piazza et al. (2011) Borges et al. (2016)
	Universal Kriging (UK)	Air temperature	Hudson and Wackernagel (1994) Benavides et al. (2007)
		Air humidity	Nguyen et al. (2015)
		Wind speed	Luo et al. (2008)
	Kriging with External Drift (KED)	Air temperature	Hudson and Wackernagel (1994) Monestiez et al. (2001) Benavides et al. (2007)
		Wind speed	Eguía et al. (2016)

The field of spatial interpolation methods covers mechanical or deterministic models as well as linear statistical or probabilistic models as described by Goovaerts (1997) and Hengl (2011). The mechanical models, e.g. Thiessen polygons, IDW or Regression on coordinates, use deterministic prediction techniques Hengl (2011).

Thiessen polygons also known as Voronoi diagrams or Dirichlet tessellation are quite simple mechanical interpolation methods. They partition an area only by the locations of the observed points (Thiessen, 1911). The studied area is separated into polygons, one for each observation point representing the shortest distance to every other neighbouring point. For each polygon, the observed value of the measured point is used (Thiessen, 1911).

As an illustrative example, three points were placed in a simple Cartesian coordinate system and then provided with values (figure 2.4). There is no value for the location marked with a question mark. According to the method of Thiessen polygons, the area was divided in such a way that sub-areas were created representing the known locations. The location marked with a question mark would be assigned the value of the closest location, in the example the one with the value 17 (x_1).

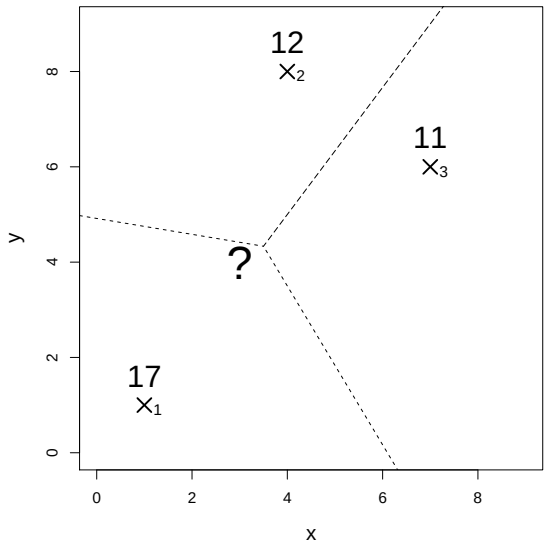


Figure 2.4: Exemplary representation of Thiessen polygons with three locations for which values are known (x_1, x_2 and x_3) and one location ($?$) for which the value is interpolated

Inverse distance weighting (IDW) is a very common empirical spatial interpolation method. Shepard (1968) defined it as a weighted average of the data point values. The weighting itself is a function of the distance:

$$\hat{z}(x_0) = \frac{\sum_{i=1}^{n}(d_i)^{-p} * z(x_i)}{\sum_{i=1}^{n}(d_i)^{-p}} \text{ if } d_i \neq 0 \tag{2.3}$$

with $\hat{z}(x_0)$ as value to be estimated at location x_0, $z(x_i)$ as known value at a specific location x_i, d_i as distance between the estimated and the known data point, p as inverse distance power and n as the count of the known data points closest to the estimated location. Thus the result of an IDW interpolation can be influenced by the count of considered data points n and the inverse distance power p. As equation 2.3 shows, a higher inverse distance power increases the influence of the measured point data in the distance until the result converges to the polygonal shape of Thiessen

polygons. According to Shepard (1968) and Babak and Deutsch (2009) empirical studies show an inverse distance power of 2 usually leads to the best results.

The IDW method was also applied to the example known from the Thiessen polygons (figure 2.5). Again, a location was assumed for which no value is known.

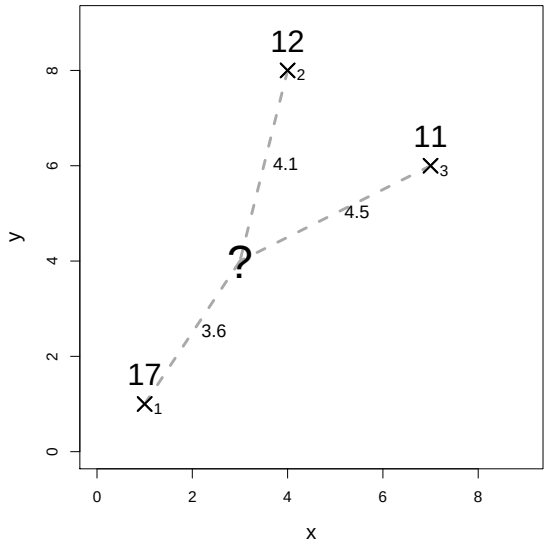


Figure 2.5: Illustration of the distances from locations for which values are known (x_1, x_2 and x_3) to one location ($?$) for which the value is interpolated using Inverse distance weighting

Using an inverse distance power of 2 the application of equation 2.3 to the values and distances shown here will then look like this:

$$\hat{z}(x_0) = \frac{3.6^{-2} * 17 + 4.1^{-2} * 12 + 4.5^{-2} * 11}{3.6^{-2} + 4.1^{-2} + 4.5^{-2}} = 13.8 \tag{2.4}$$

At the location marked with a question mark, a value of 13.8 is interpolated with the IDW method. In order to generate spatial raster data, this process would be carried out iteratively for the centres of the raster cells.

Apart from this mechanical models more complex statistical models like Ordinary Kriging (OK) or techniques using additional independent variable like Regression Kriging (RK), Universal Kriging (UK) or Kriging with External Drift (KED) are used to interpolate point data.

Ordinary Kriging (OK) is a specialized form of Kriging interpolation, first established by Krige (1951) to improve predictions made for mining operations. According to Cressie (1990) the mathematical fundament of Kriging was firstly derived by Math-

eron (1963). The difference between mechanical and statistical interpolation methods, like kriging, is that the spatial dependence of the variable is not just assumed, but analysed and integrated in the regionalisation. The idea of Krige (1951) was, to implement the spatial dependence by weighting the values near to the unknown location and summing them (equation 2.6). The OK method uses variography to identify the spatial dependence by plotting the distance between the sampled points versus the points semivariance Matheron (1963). The semivariance ($\gamma(h)$) is calculated following the expression:

$$\gamma(h) = \frac{1}{2 * N(h)} \sum_{i=1}^{N(h)} [z(x_i + h) - z(x_i)]^2 \tag{2.5}$$

with $z(x_i)$ as observed value at location x_i, $z(x_i + h)$ as observed value with the distance of h to location x_i and $N(h)$ as the number of point pairs with the distance h. With the generated plot a theoretical variogram can be fitted to the data points representing the spatial dependence of the variable. With this variogram it is possible to identify the weight values (λ_i) for the kriging equation (Matheron, 1963; Cressie, 1990):

$$\hat{z}(x_0) = \sum_{i=1}^{n} \lambda_i * z(x_i) \tag{2.6}$$

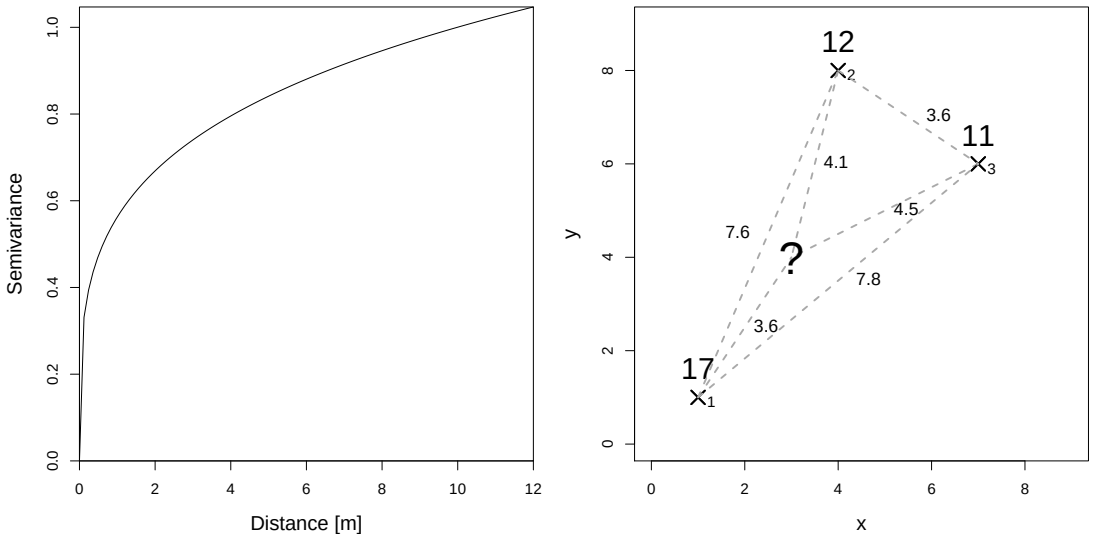
To to identify the weight values (λ_i) it is necessary to calculate the kriging variance (σ^2) according to Cressie (1991) assuming fulfillment of the intrinsic hypothesis and unbiasedness of the estimate ($\sum_{i=1}^{n} \lambda_i = 1$):

$$\sigma^2(x_0) = 2 \sum_{i=1}^{n} \lambda_i * \gamma(x_i x_0) - \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i * \lambda_j * \gamma(x_i x_j) \tag{2.7}$$

$\gamma(x_i x_0)$ describes the semivariance between the known value at location x_i and the unknown value at location x_0. Equally $\gamma(x_i x_j)$ stands for the semivariance between the known value at location x_i and another known value at location x_j. To examine the weight values (λ_i) it is necessary to get the global minimum of equation 2.7 since this solution offers the interpolation with the lowest kriging variance. Following Schaeben et al. (2013) this is possible by the implementation of a Lagrange multiplier μ which allows the transformation to the system of equations:

$$\begin{cases} \sum_{j=1}^{n} \lambda_j * \gamma(x_i x_j) + \mu = \gamma(x_i x_0) \quad (i = 1, ...n) \\ \sum_{i=1}^{n} \lambda_i = 1 \end{cases} \tag{2.8}$$

Using this system of equations the weight values resulting in the lowest kriging



(a) Exemplary theoretical semivariogram

(b) Spatial dataset with the distances between locations for which values are known (x_1, x_2 and x_3) and to one location ($?$) for which the value is unknown

Figure 2.6: Most important components of the Ordinary Kriging procedure

variance can be identified for each predicted value, based on the semivariance of the fitted semivariogram.

To illustrate the explanation above, the OK procedure is applied to an example dataset (figure 2.6). Figure 2.6(a) shows a semivariogram which has not been adapted to the actual values of the example, since this is not useful for only three point pairs. It is only used here for an example calculation of the OK procedure.

The system of equations 2.8 is used to calculate the weights for the influence of the values shown in figure 2.6(b) on the value being interpolated:

$$\begin{cases} \lambda_1 * 0 + \lambda_2 * 0.93 + \lambda_3 * 0.94 + \mu = 0.77 \\ \lambda_1 * 0.93 + \lambda_2 * 0 + \lambda_3 * 0.77 + \mu = 0.80 \\ \lambda_1 * 0.94 + \lambda_2 * 0.77 + \lambda_3 * 0 + \mu = 0.82 \\ \lambda_1 + \lambda_2 * + \lambda_3 + 0 = 1 \end{cases} \tag{2.9}$$

The values inserted in the system of equations are derived from the distance values (figure 2.6(b)) and the semivariogram (figure 2.6(a)). The semivariance $\gamma(x_1 x_2)$ for example results in a value of 0.93 for the distance 7.6 m. The system can be solved by the Lagrange multiplier μ which leads to the weights $\lambda_1 = 0.40$, $\lambda_2 = 0.31$ and $\lambda_3 = 0.29$.

These weights can then be inserted in equation 2.6:

$$\hat{z}(x_0) = 17 * 0.40 + 12 * 0.31 + 11 * 0.29 = 13.71 \tag{2.10}$$

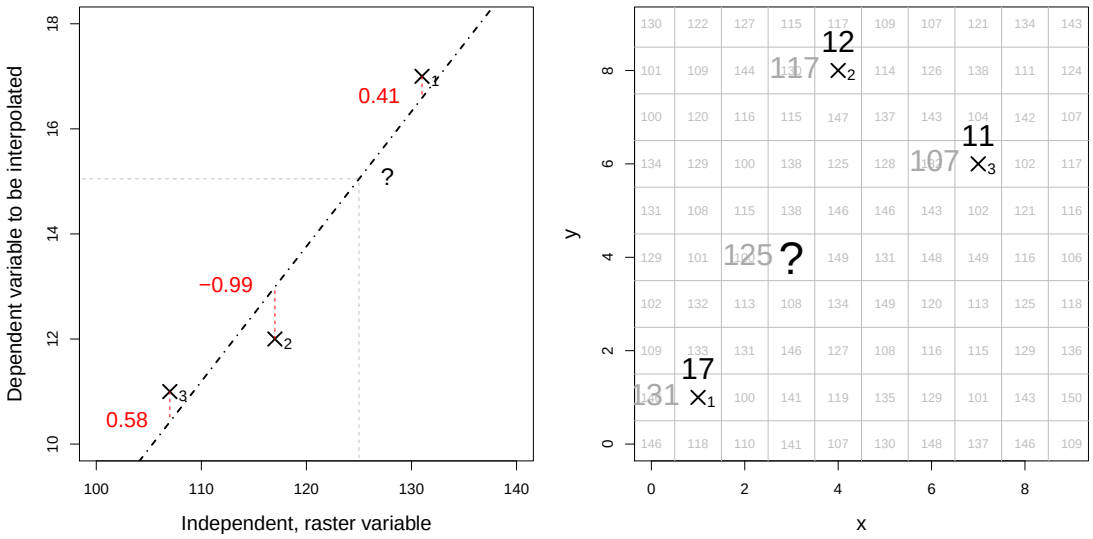
Accordingly, by applying OK interpolation and using the exemplary theoretical semivariogram (figure 2.6(a)), a predicted value of 13.71 is obtained for the position in question.

Regression Kriging (RK), Universal Kriging (UK) and Kriging with External Drift (KED) work mathematically the same way (Hengl et al., 2003; Hengl, 2011). The basic idea of this procedures is the integration of independent variables in the process of kriging interpolation. According to Webster (1994) and McBratney et al. (2000) the UK procedure was established by Matheron (1969) as a combination of OK and a multiple-linear regression of the dependent with the independent variable. Equation 2.11 shows how this combination works. A multiple linear regression is fitted to the variables to predict the dependent by the independent variables. Thereby the independent variables should be available as regionalised data. The regression model ($\hat{m}(x_0)$) then can be applied to the independent variables. The kriging procedure ($\hat{z}(x_0)$ as in equation 2.6) is used to interpolate the residuals of the regression model.

$$\hat{e}(x_0) = \hat{m}(x_0) + \hat{z}(x_0) = \sum_{k=0}^{p} \beta_k * q_k(x_0) + \sum_{i=1}^{n} \lambda_i * z(x_i) \tag{2.11}$$

So for each of the p independent variables $q_k(x_0)$ regression coefficients (β_k) are derived and used for the prediction of $\hat{m}(x_0)$ at the unknown location x_0. This value gets summed up with the interpolated residuals $z(x_i)$ at the known locations x_i to receive the interpolation methods prediction of the dependent variable ($\hat{e}(x_0)$). The independent variables are in the case of UK the geographical coordinates, which therefore define the drift in the kriging equation (McBratney et al., 2000). KED and Regression Kriging (RK) work similar to UK but not the geographic coordinates are used to determine the drift in the study area. Instead ancillary variables, like a digital elevation model, are used (Goovaerts, 1997; McBratney et al., 2000; Hengl et al., 2003).

Figure 2.7 presents an exemplary application of these spatial interpolation methods. The dataset from the previous examples is shown in figure 2.7(b), but this time it is assumed that the variable to be interpolated depends on an independent parameter, already available as a spatial raster dataset. The known locations, therefore, have not only a value to be interpolated (black number), but also the value of the independent variable (grey number). The independent variable is also available for the location to



(a) Linear regression fitted on raster variable, including the residuals (red)

(b) Spatial dataset with locations for which values are known (x_1, x_2 and x_3) and one location ($?$) for which the value is unknown, including the values of the independent variable (grey)

Figure 2.7: Major components of the Kriging with External Drift procedure

be interpolated (question mark). A linear model is adapted to the variables (figure 2.7(a)) which produces a prediction of 15.05 for the covariate value of 140 at the location in question.

To correct this value using the residuals of the linear model, they are interpolated using the OK method. For reasons of simplicity, it is assumed that the semivariogram, shown in figure 2.6(a), can also be used here. Taking into account the already solved system of equations 2.9, the value of 15.05 determined by the linear model (figure 2.7(a)) and the equation 2.11, the following equation is created:

$$\hat{e}(x_0) = 15.05 + (0.41 * 0.40 - 0.99 * 0.31 + 0.58 * 0.29) = 15.08 \tag{2.12}$$

The predicted value for the place marked with a question mark is 15.08. The small difference to the value determined with the linear model shows the strong influence of this model on the KED interpolation. Nevertheless, the three-point linear model shows only small residuals. For models with larger point sets, which may also contain outliers, the interpolation of the residuals can have a stronger influence on the interpolated results.

2.4.1 Examples of the application of regionalisation methods

Regionalisation procedures are widely used in various contexts. In the course of this work, the methods are used for the spatial interpolation of weather data. Some examples from this field are outlined below.

Air temperature data A number of authors tested the regionalisation methods presented above to interpolate temperature data. Hudson and Wackernagel (1994) used the technique of UK to interpolate January mean temperatures in Scotland. Since they judged the results of UK to be not satisfactory Hudson and Wackernagel (1994) tested KED with the elevation as an ancillary variable, which improved the spatial interpolation. Monestiez et al. (2001) used a modified KED to interpolate the air temperature in a 250 km by 150 km large area in south-east France and compared the results to OK. The presented KED approach uses environment classes of CORINE landcover data as an ancillary variable to predict the daily maximum summer temperature. Since they used the categorical environment classes, it was named "categorical external drift kriging" by Monestiez et al. (2001). Cross-Validation was used to find out, the prediction of the modified KED worked out better for each class than OK. Benavides et al. (2007) compared OK, KED and UK predicting air temperature in a mountainous region of Northern Spain. The KED method used elevation as ancillary variable like Hudson and Wackernagel (1994) did. To evaluate the performance of their model, they used a leave-one-out cross-validation (LOOCV) and calculated the root-mean-squared-error normalized by the standard deviation (NRMSE) from the observed and predicted values, normalised by the standard deviation. This comparison showed the lowest NRMSE with the KED procedure, followed by UK and OK (Benavides et al., 2007).

Air humidity data Nguyen et al. (2015) studied different methods to interpolate the relative humidity and the temperature at study sites in Vietnam. They compared a OK, a UK and a KED approach - using the elevation as ancillary variable - with a 10-fold cross validation, a subform of LOOCV. The KED method was only used for the temperature. The RMSE for the temperature was lowest for the KED method followed by UK and OK (Nguyen et al., 2015). The comparison for the relative humidity showed the lowest RMSE for the UK method followed by the OK method (Nguyen et al., 2015).

Wind speed data Luo et al. (2008) compared different deterministic and statistical regionalisation methods to interpolate wind speed surfaces in England and Wales including IDW, OK and UK. LOOCV and RMSE were used to evaluate the prediction.

OK showed the smallest RMSE ($1.61\,\mathrm{m\,s^{-1}}$), followed by UK and IDW (Luo et al., 2008). Eguía et al. (2016) tested OK, KED, using longitude, latitude and altitude, and a second KED approach, additionally using the distance to the coast, to interpolate the wind speed in north-west Spain. Again the RMSE was derived from the LOOCV. The lowest RMSE was derived using the first KED approach without the distance to the coast followed by the second KED approach and OK (Eguía et al., 2016).

Precipitation data Thiessen (1911) created the Thiessen polygons to interpolate the precipitation in Utah. Since then more complex interpolation methods were used for this purpose. Wagner et al. (2012) for example compared the Thiessen polygons with the methods IDW, OK and RK in the catchment of the Mula and the Mutha Rivers in India. The RMSE of the LOOCV showed the best result for the IDW method followed by OK method and the Thiessen polygons method (Wagner et al., 2012). Piazza et al. (2011) interpolated rainfall data in Sicily in Italy, measured at 247 locations. Among others they tested IDW with an RMSE of and OK with an RMSE of (Piazza et al., 2011). Borges et al. (2016) also used a cross validation to test these methods for precipitation in Central Brazil. Unlike the results of Piazza et al. (2011), Borges et al. (2016) obtained a lower RMSE with the deterministic IDW method in comparison to the statistic OK procedure.

2.5 Machine learning methods

Chapters 2.1.1 and 2.2.1 have shown that numerous information is available on the parameters influencing the development of *Blumeria graminis* and *Puccinia triticina*. Linking this information with knowledge about infestation strengths of the pathogens is a task for machine learning techniques. As Langley (1986) pointed out it is difficult to capture the field of machine learning in a hard definition. It can only be contoured as a field of intelligent systems that enhance their adaption over time (Langley, 1986). The field of machine learning is roughly structured into the branches of supervised and unsupervised learning algorithms (Hastie et al., 2009; Sutton, 1992).

2.5.1 Supervised learning techniques

Supervised learning techniques are the most common ones. They study the effects of input variables, so-called independent variables, on the target variable, the dependent variable (Hastie et al., 2009; Forte, 2015). Both the independent and the dependent

variables are known. These techniques are further separated into classification and regression predictions. As the names imply, classification predictions are used for categorical variables, while regression predictions are best for metric target variables (Lantz, 2015). Popular examples of supervised learning models are Nearest Neighbour, Naive Bayes, Decision Trees (DTs), Linear Regression, Regression Trees and Neural Networks (Lantz, 2015).

Nearest Neighbor classifications create a multidimensional feature space with each dimension representing an independent variable (Lantz, 2015). For the identification of a new element's target variable a Nearest Neighbor algorithm determines the Euclidean distance between this new element and the elements of this multidimensional feature (Fix and Hodges Jr, 1951). If only the closest observation based on the independent variables is selected, this is referred to as the 1-NN method (Cover and Hart, 1967). If distant values are also taken into account, the term k is used to refer to the number of closest neighbors (Harley et al., 1963). The most frequent target variable value of the k-Nearest Neighbors (k-NNs) then gets assigned to the new element (Patrick and Fischer, 1970). The output of the model strongly depends on the choice of the k value. While a small k is prone to overfitting a larger k is liable to generalisation (Cover and Hart, 1967). Also, the scale of the independent variables has a major influence on the prediction results, as they are used for the distance measurement. For this reason Larose (2005) suggests that these variables should be normalised using a min-max or a z-score normalisation. This classification is also called lazy learner, since it is only a comparison with the existing data and no model is created during the prediction process to abstract and generalise the information (Lantz, 2015).

Figure 2.8 shows a simplified example of the function of the k-NN method. A data set is assumed for which the characteristic of the dependent variable ("Circle" or "Triangle") relies on the two independent variables 1 and 2. An observation "X" is also assumed for which only information on the independent variables is available. Considering only the closest known neighbour (k=1), the value "Triangle" is assumed for the unknown point. Taking into account the four closest neighbours (k1 to k4), the point would be assigned the characteristic "Circle" by three of four observations which leads to a probability of 75 %. Figure 2.8 also indicates the influence of not normalised covariables with different value ranges. If the "Independent Variable 2" reached from 150 to 450 instead of 1.5 to 4.5, the circles to the left would be closest to the "?". The use of two covariables is of course only for illustrative purposes. The distances can also be calculated in n-dimensional spaces.

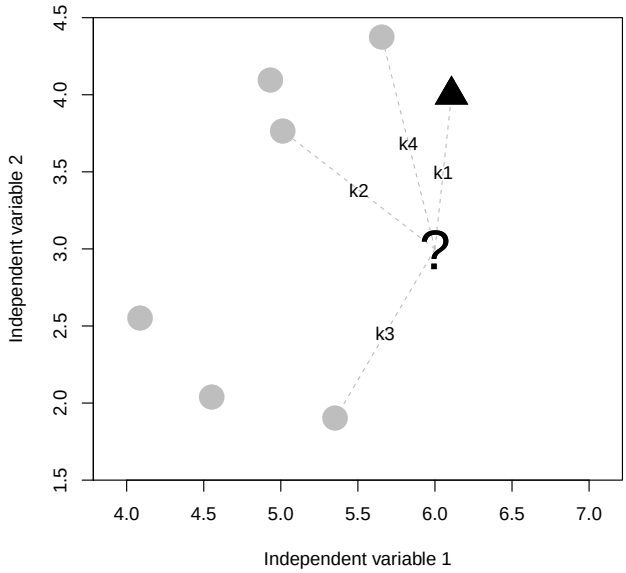


Figure 2.8: Illustration of the *k*-Nearest Neighbors of an observation for which the values of two independent variables are known but not whether is has the characteristic "Cirlce" or "Triangle"

Although the k-NN method has not yet been used in the context of this work, it has already been used in fields close to this thesis. In the field of phytopathology, the k-NN method is currently used mainly for image recognition. Zhang et al. (2015) used a k-NN classifier to recognise plant diseases by RGB images of leafs. Delwiche et al. (2013) also used the k-NN method to derive the infestation of wheat kernels based on photographs. Lu et al. (2017) used the k-NN approach to detect fungal infections in strawberry based on hyperspectral data.

Naive Bayes classifications are so called probabilistic learners. They are based on Bayes' theorem (Bayes, 1763):

$$\Pr(A|B) = \frac{\Pr(B|A) * \Pr(A)}{\Pr(B)} \tag{2.13}$$

which describes the conditional probability ($\Pr(A|B)$) that event A, the dependent variable, occurs under the condition that B, the independent variables, has happened. The Naive Bayes classification algorithm is built on this equation, but it includes the product of the probabilities of the independent variables since the basic assumptions

include the independent variables independency and their equal importance:

$$\Pr(A|B) = \prod_{i=1}^{n} \Pr(B_i|A) \tag{2.14}$$

with B as the vector containing the independent variables and A as the target variables class (Rish, 2001). By assuming that the elements of B are conditionally independent of each other, the general expression for Bayes theorem is simplified (Dybowski et al., 1993). According to Rish (2001) and Lantz (2015) Naive Bayes classifications usually are used for text classification, since the method uses frequency tables, what results in the need of discretisation in case of numeric features.

The Naive Bayes model was also used for image classification. Mwebaze and Biehl (2016) used Naive Bayes to recognise crop disease on images of cassava. The model was also applied by Sankaran and Ehsani (2013). They classified images of a handheld fluorescence sensor to identify diseases in citrus leaves.

Decision Trees (DTs) are based on the idea of recursive partitioning. It splits the dataset repeatedly into smaller subsets until the subsets are homogeneous regarding the target variable. Since the first algorithms in Morgan and Sonquist (1963) (as cited in Loh 2014) different algorithmic methods were developed to reach this aim like the C5.0, the 1R and the CART algorithm (Loh, 2011).

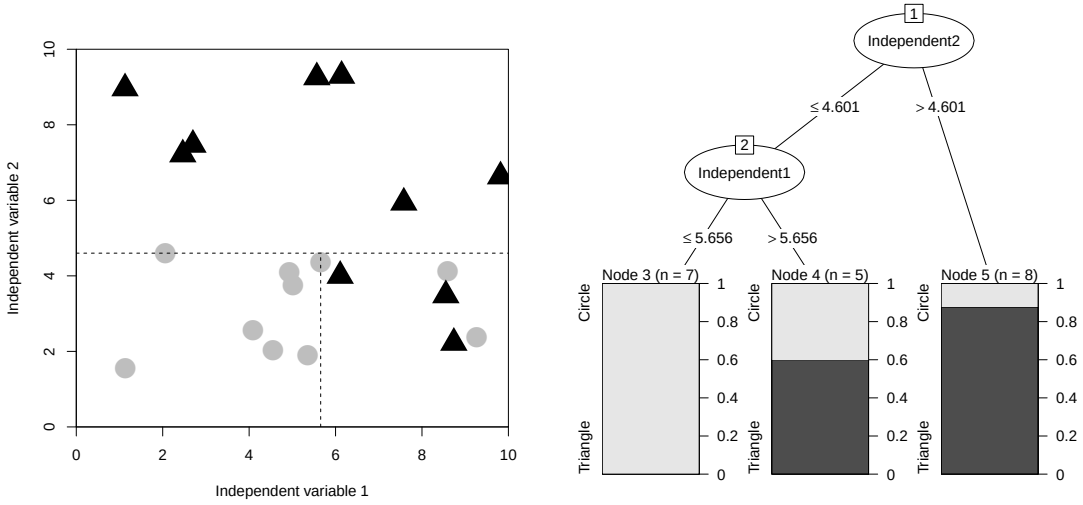
The most common C5.0 DT algorithm, a further development of the C4.5 algorithm, uses the concept of entropy to create subsets with a maximum purity. The entropy for a specific subset is calculated:

$$Entropy(S) = \sum_{i=1}^{c} -p_i * log_2(p_i) \tag{2.15}$$

with S as the subset, c as the number of class levels and p_i as the proportion of values in the specific class (Lantz, 2015). After each split of the data set, the summed entropy values of all subsets represent the quality of the split:

$$Entropy(T) = \sum_{i=1}^{c} w_i * Entropy(P_i) \tag{2.16}$$

with the total entropy (T) and the weighting (w_i) by the proportion of examples in the subset (Lantz, 2015).The entropy values of the potential splits are weighted against each another, which results in the information gain of potential splits so that the split with the highest information gain is chosen. According to Kotsiantis (2013) besides



(a) Dataset with the observations "Cirlce" and "Triangle" depending on the independent variables 1 and 2 including the splits of (b)

(b) Decision Tree fitted to the dataset depicted on the right in figure (a)

Figure 2.9: Visualisation of a Decision trees classification of a dataset

the information gain splitting measure the gain ratio or the Gini value are standard evaluation methods.

The main problem of DT classification is the overfitting of the model at the noise of the calibration dataset (Fürnkranz, 1997). To overcome this problem pruning strategies are applied to most models. One option is pre-pruning. These methods consider the noise during the learning process. Heuristics are used to stop the learning process to avoid a theory that is complete and consistent but also influenced by the noise (Fürnkranz, 1997). Post-pruning algorithms work inversely. They create a model fitted completely on the training data. The branches of this tree that are not representative are then pruned back (Fürnkranz, 1997).

To illustrate a simplified example, the data record shown in figure 2.8 has been extended (figure 2.9(a)). A DT was then adapted to this data (figure 2.9(b)). The first split (Node 1), based on the "independent variable 2", produces an almost pure subset (Node 5). The DT shows, that this subset consists of eight observations of which 88 % are "Triangle". One of the observations in this subset is a "Circle", because the algorithm divides at the observations and not between them. The next split (Node 2) splits the remaining data set based on the "independent variable 1", creating a pure "Circle" subset (Node 3) and a mixed subset (Node 4).

There are several examples for the use of decision trees in Phytopathology. Sankaran et al. (2012), for example, used bagged DTs to differentiate between in-

tact and damaged plant tissue using visible-near infrared spectroscopy. Olatinwo et al. (2009) applied DTs to predict diseases of peanuts in the USA. For this purpose, they compared the infestations on the peanuts of the year with a number of parameters, such as the average daily temperature and the accumulated rainfall in March and April. Temperature and precipitation data were also used by Kim et al. (2014) for the prediction of the disease risk of downy mildew in boysenberry. They summarised these meteorological data on a monthly basis and connected these using a DT with the seasonal infestation risk.

Random Forests (RFs) were developed by Breiman (2001). They combine the aforementioned DTs with the bagging procedure, also developed by Breiman (1996). The bagging procedure generally uses bootstrap sampling, a random sampling with replacement method, to generate multiple predictions for a dataset using the same prediction method (Breiman, 1996). The individual results are combined to one final prediction using plurality vote for classificated and the average value for numerical outputs. The RF approach works similar. Therefore it is referred to as ensemble learner, which allocates parts of the original dataset to many learners, creates individual prediction models and combines those models to one ensemble model (Khoshgoftaar et al., 2015). The RF algorithm does not only split the original dataset once, but creates random subsets of the dataset at each node of the developing tree (Breiman, 2001). The splits are evaluated using the out-of-bag (OOB) estimates. These estimates test the ongoing grown trees at the cases, not yet integrated in the model to give an error estimate (Breiman, 2001). This way, a forest of randomly grown DTs emerges. The combination of these trees as average for regression and as most frequent value for classification tasks results in the final prediction of the RF model. RF are less prone to overfitting, since only parts of the dataset are used to generate the individual trees (Wyner et al., 2017). Also they are better at dealing with larger datasets with a large number of features (Lantz, 2015).

As an example in figure 2.10, the RF approach is applied to the data set shown in figure 2.9. In this basic example, 3 trees were generated, each for a random subset of the original data set. Due to the specific reduction of the data set, various DT models were created, which subdivide the data set differently. The first one (2.10(a) and (b)) depends on the "independent variable 1", the second (2.10(c) and (d)) on the "independent variable 2" and the third (2.10(e) and (f)) on both variables. Table 2.4 outlines two predictions based on the independent variables and the RF model. In the first case, all DTs forecast "Circle". The majority decision determines a "Circle" as the prediction of the RF model. In the second case, the first tree differs from the

others by predicting a "Circle". Since the RF algorithm follows the majority decision again, a "Triangle" is being forecast here.

Table 2.4: Predictions of the Random Forest model depicted in figure 2.10

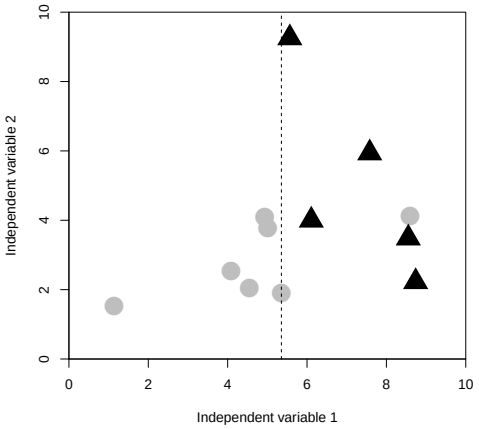
Independent variable 1	Independent variable 2	Tree 1	Tree 2	Tree 3	RF prediction
2	2	Circle	Circle	Circle	Circle (3/3)
2	8	Circle	Triangle	Triangle	Triangle (2/3)

RF models were also applied in the field of Phytopathology. Chemura et al. (2017) used such models to identify coffee leaf rust infection levels from Sentinel-2 spectra. Wen et al. (2017) studied the short distance transport of soybean rust in dependence of environmental parameters, such as humidity, temperature and wind direction and speed. They predicted the spread of the rust in the canopy applying a RF model. Harteveld et al. (2017) created a model to forecast the spore release of a fungal pathogen infecting blueberries. They used several parameters observed by in-field weather stations and trained different machine learning algorithms by which the RF achieved the highest accuracy.

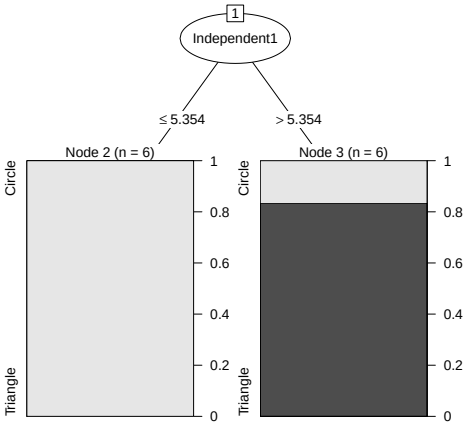
Boosted Decision Trees (BDTs) are another method based on the creation of multiple DTs. BDTs are utilising the boosting procedure. Similar to the bagging method used by RFs, the boosting procedure creates a number of DTs (Quinlan et al., 1996). In contrast to the RFs the Decision Trees of the BDTs do not consist of random sub-data sets. Instead, only one DT is generated at first based on the entire data set. The weight of misclassified instances in this first tree then is increased when the next tree is generated (Schapire and Freund, 2012). Until the requested number of trees is reached, this process continues. As with RF, the prediction of this procedure is based on the majority decision of the generated DTs.

Figure 2.11 presents a simplified example using the same dataset as figure 2.9. Figure 2.11(a) and (b) demonstrate that the first adapted DT came to the same result as in the example in Figure 2.9. The misclassifications are highlighted in red in figure 2.11(a). The increased weight of these, when the next DT was generated (subfigure (d)), is also reflected by the misclassified "Triangles" in the lower half of subfigure (c). The "Triangles" were then considered to a greater extent when creating the tree (f), which again led to a different division of the data set (subfigure (e)).

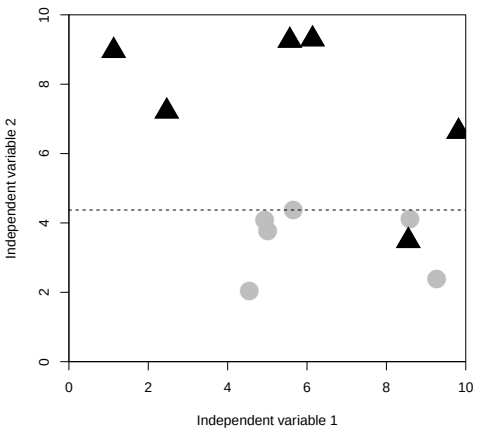
The application of BDT methods is not very popular in the field of Phytopathology. Shah et al. (2014), however, used Boosted Regression Trees to predict Fusarium head blight infestations on winter wheat in the USA. They compared the prediction with the



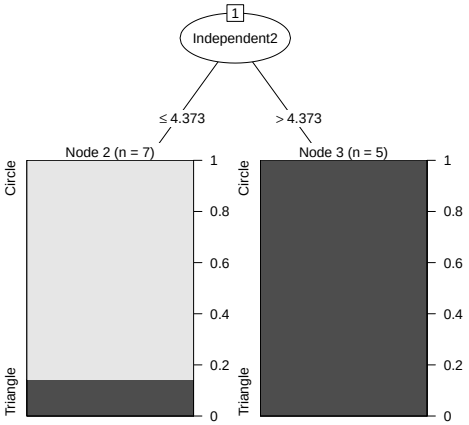
(a) First random subsample of dataset



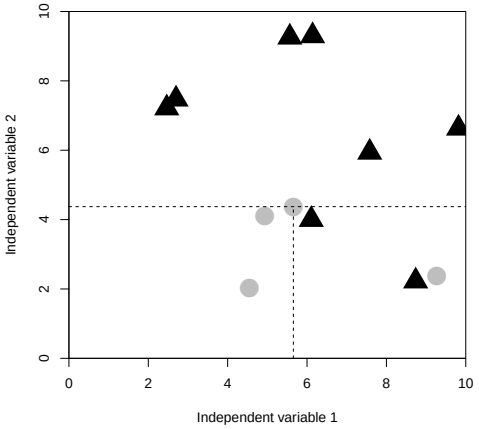
(b) Decision Tree fitted to (a)



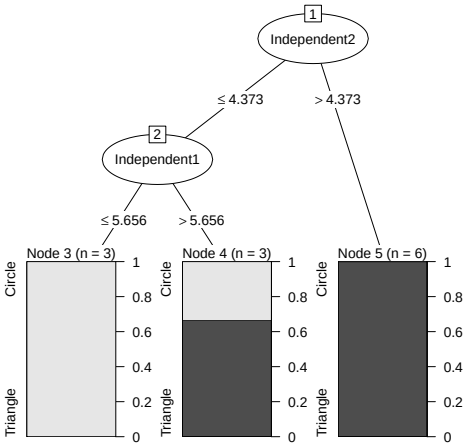
(c) Second random subsample of dataset



(d) Decision Tree fitted to (c)

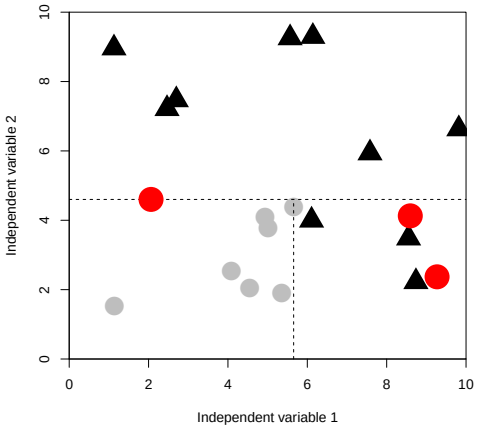


(e) Third random subsample of dataset

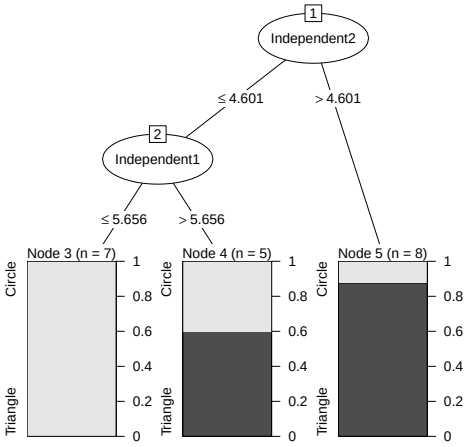


(f) Decision Tree fitted to (e)

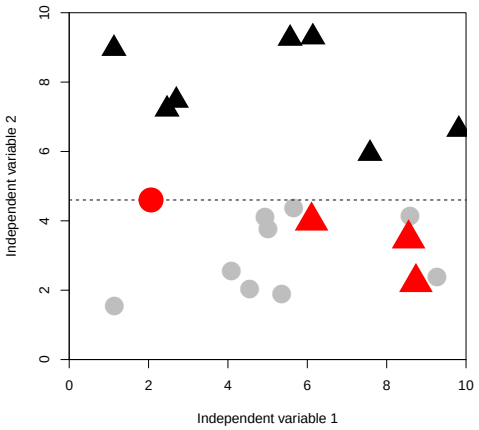
Figure 2.10: A Random Forests classification of the dataset depicted in figure 2.9(a)



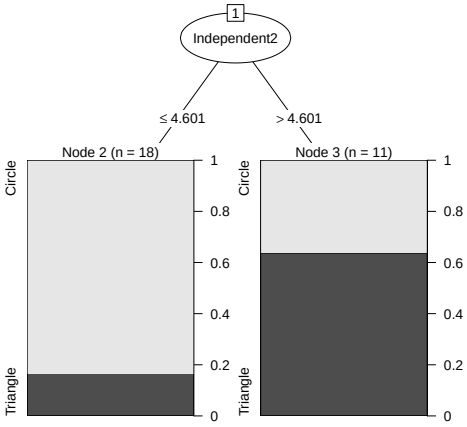
(a) Dataset including splits of (b) and misclassified observations (red)



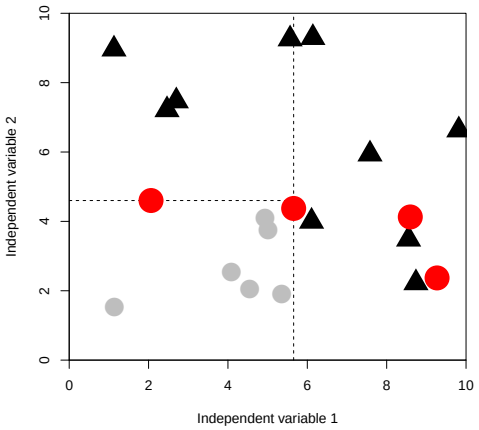
(b) Decision Tree fitted to (a)



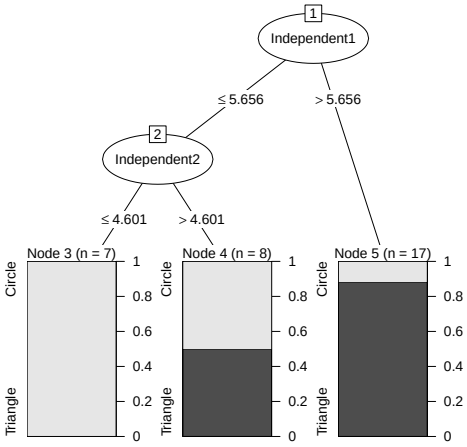
(c) Dataset including splits of (d) and misclassified observations (red)



(d) Decision Tree fitted to (a) with larger weigth on misclassified values



(e) Dataset including splits of (f) and misclassified observations (red)



(f) Decision Tree fitted to (c) with larger weigth on misclassified values

Figure 2.11: A Boosted Decision Trees classification of a dataset (figure 2.9(a))

forecasts of logistic regression models and achieved considerably better results with the Boosted Regression Trees.

Regressions model the relation between a metric target and one or several also metric independent variables. This relation is expressed by the general regression model:

$$y_i = m(x_i) * \varepsilon_i \tag{2.17}$$

with the target variable (y), the independent variable (x), the relation between them (m) and the error variable (ε) (Fahrmeir et al., 2009). In terms of linear regressions, this general equation is simplified to the linear regression model:

$$y_i = \gamma + \beta x_i * \varepsilon_i \tag{2.18}$$

with the constant axis intercept (γ) and the slope of the regression line (β) (Fahrmeir et al., 2009). The most common way to fit the regression line to the observations is the method of Ordinary least squares (OLS). The regression line following the OLS method is fitted to the observations by the lowest sum of residues ($min \sum_{i=1}^{n} \varepsilon_i^2$) (Wooldridge, 2015).

As a combination of regression models and DTs Breiman et al. (1984) presented the Classification and Regression Tree (CART) algorithm. The key element of the regression trees is the reduction of the Sum of Squared Errors (SSE). Like the lowest sum of residues, the SSE is an expression of the best fit of the model to the data:

$$SSE = \sum_{i=1}^{n} (y_i - \bar{y}_i)^2 \tag{2.19}$$

with y_i as values of the branches observations and $\bar{y}_i$ as the average value of these n observations (Forte, 2015). The regression tree nodes are selected so that the summed SSE values of all branches are as low as possible.

The use of regression models is quite common, also in the field of Phytopathology. Smith et al. (2007), for example, used regression techniques to predict the disease incidence of a fungus on peanut plants based on meteorological data. Harikrishnan and Ro (2008) generated predictions about the disease incidence of a fungus that can infest the dry bean. They used information on the total rainfall, average minimum temperature and number of rainy days in the first half of June, July and August and combined it with known disease incidence data by applying regression methods. Mehra et al. (2017) produced a regression between the independent variables temperature,

relative humidity and precipitation and the probability of disease onset of a fungus on wheat plants.

Artificial Neural Networks (ANNs) are based on the function of biological neurons. Roughly speaking, the biological neuron consists of a nucleus, which receives weighted electrical signals from the dendrites by the cells soma (Forte, 2015). These weighted electrical signals are summed up and if the sum crosses a threshold the signal is sent to the neurons axon, creating an output signal (Kruse et al., 2015). This idea of a threshold deciding if or if not an action will occur inspired McCulloch and Pitts (1943) to take the first steps in the direction of ANN referred to as McCullogh-Pitts model. It was further developed to the so called perceptron by Rosenblatt (1958). In mathematical equations the idea of this perceptron as counterpart to the biological neuron is:

$$y(x) = g(\sum_{i=1}^{n} w_i x_i) \tag{2.20}$$

with y as the target variable, x_i as the independent variables, w_i as the weights of this variables and $g()$ as activation function (Forte, 2015). The activation function closest to the biological neuron is the step function:

$$g(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{2.21}$$

although Lantz (2015) points out, that the most used activation function is the logistic sigmoid activation function:

$$g(x) = \frac{1}{1 + e^{-x}} \tag{2.22}$$

This model was developed further by Bishop (1995) to the so called multi-layer perceptron (MLP) neural network. The output of a single-layer perceptron directly results from the weighted input signals. Whereas the MLP uses so called hidden layers which are influenced by the input signals and equally influence the output or further hidden layers which then would influence the output (Lantz, 2015).

Phytopathology also applied ANNs in the past. Wolf et al. (2000) predicted the occurrence of the fungi Tan spot and Stagonospora blotch of spring wheat in dependence of temperature, relative humidity, precipitation, and leaf wetness duration using ANNs. Paul and Munkvold (2005) also created a ANN forecast model. They predicted the disease severity of the fungal gray leaf spot of maize using the independent variables temperature and humidity.

2.5.2 Unsupervised learning techniques

Unsupervised learning techniques are used, if there are no known values for the dependent variable. Such techniques try to structure and cluster the independent variables (Forte, 2015). Examples of unsupervised learning techniques are Association Rules, k-means clustering and Principal Components (Hastie et al., 2009; Lantz, 2015).

2.6 Statistical measures of the models performances

When using machine learning methods, the performance of the applications is a key issue. Various parameters are widely used for this purpose. The parameters accuracy, sensitivity, specificity, precision and ROC Area Under the Curve (ROC AUC) were selected to summarise the classification performance of the models created in this work. They are most comfortable to visualize when the predicted and observed values are compared in a cross-table like table 2.5 (Altman and Bland, 1994).

Table 2.5: Sample representation of a cross-table

		Predicted	
		No event	Event
Observed	No event	True Negative (TN)	False Positive (FP)
	Event	False Negative (FN)	True Positive (TP)

Dangerous infestation with pathogens can or cannot be observed in the field. If such an event has not occurred and no one has been predicted, this is referred to as "True Negative". In contrast, a positive forecast would be called "False Positives". A correctly predicted event would then be "True Positive", while an unpredictable event would be "False Positive". Using this terminology, the above parameters are explained.

The *accuracy* is the proportion of all correct predictions in all predictions:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.23}$$

The *sensitivity* is the proportion of true predictions of an dangerous infestation of all observed events:

$$sensitivity = \frac{TP}{TP + FN} \tag{2.24}$$

The *specificity* is the proportion of true predictions of no dangerous events of all observed cases without dangerous events:

$$specificity = \frac{TN}{TN + FP} \tag{2.25}$$

The *precision* is the proportion of true predictions of an endangering occurrence of pathogens of all predicted events:

$$precision = \frac{TP}{TP + FP} \tag{2.26}$$

Receiver Operating Characteristic (ROC) The ROC Area Under the Curve (ROC AUC), derived from the ROCR package (Sing et al., 2005), is a parameter easiest to understand by thinking of the Receiver Operating Characteristic (ROC) curve, presented in figure 2.12. Here the sensitivity, also known as "True positive rate", is plotted against the "False positive rate", the proportion of observed absences of serious events classified wrong (Forte, 2015). If the ROC AUC is high on a scale between 0.5 and 1, a large percentage of the predicted events are classified correctly without predicting the absence of an event wrong. It can be understood as the probability that a randomly selected exceedance of the yield relevant threshold will receive a higher classification probability than a randomly selected underrun of this threshold (Fogarty et al., 2005). Looking at the curve of the ROC AUC value 0.8 (figure 2.12), for example, it can be seen that in the model representing this curve, 80 % of the events occurring are also recognised as such if 20 % of the observed non-occurring events are erroneously predicted to be dangerous. With a ROC AUC value of 0.5, 80 % of the events that did not occur would be incorrectly classified at the same true positive rate. The forecast would be random.

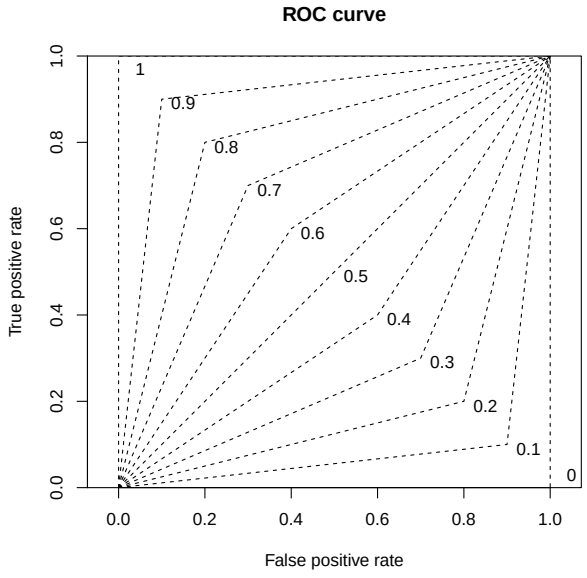


Figure 2.12: Exemplary Receiver Operating Characteristic (ROC) curves including ROC Area Under the Curve (ROC AUC) values

Chapter 3

Study area, data and methods

3.1 The study area

The study area Schleswig-Holstein (SH) is the northernmost federal state of Germany. It is located between the southern border of Denmark in the north and the two federal states of Hamburg and Lower Saxony in the south (figure 3.1). The western boundary of the study area is set by the North Sea and the eastern border by the Baltic Sea and Mecklenburg-Vorpommern. As described by Fränzle (2004), the research area is divided into four distinctive regions (figure 3.2):

(1) The eastern uplands (*Östliches Hügelland*) are characterised by a series of ice advantages which were coupled with extensive meltwater dynamics during the Weichselian Glaciation.

(2) The outwash plains (*Niedere Geest*) are alluvial cones in front of the tunnel valleys of the Weichselian Glaciation western of the eastern uplands.

(3) The residues of Saalian moraines (*Hohe Geest*) western of the outwash plains are shaped by Weichselian solifluction and slope wash processes.

(4) The marshes and mudflats in the West of the study resulted from transgressions in the Holocene.

The landform structure of the study area follows this classification. The highest relief can be seen in the eastern uplands, as the comparison of figure 3.1 and figure 3.2 shows. In this region, where the systems of moraines of the Weichselian Glaciation are closely interspersed with each other, lies the highest elevation, the Bungsberg (164 m). Only east of the Bungsberg, the dominant ground moraine creates a flatter and smoother relief (Meynen and Schmithüsen, 1962). In the east of the Bungsberg, the

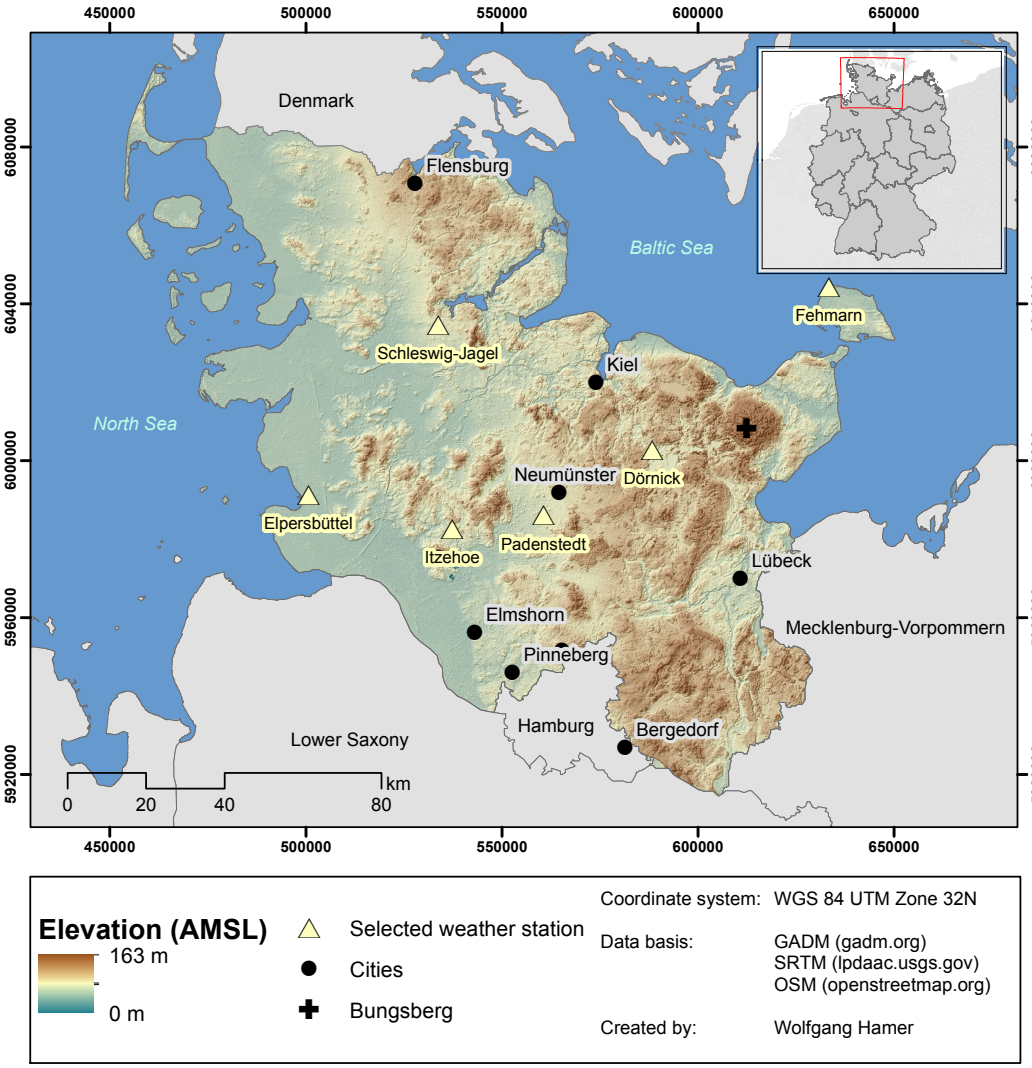


Figure 3.1: Relief, largest cities and the weather stations used for the climographs depicted in figure 3.3 in the study area Schleswig-Holstein

oscillating glaciers rearranged glaciofluvial deposits and created textural compositions that diversify within short distances from sandy to loamy and clay-rich textures (Horn et al., 2006). The till deposits which can be expected in this area tend to have a high carbonate content, but under the postglacial humid climatic conditions carbonates were dissolved and washed out of the soil. This decalcination followed by loamication resulted in a clay migration creating Luvisols or under stagnic conditions Stagnic Luvisols (Horn et al., 2006). On drier, sandier moraines Cambisols occur. In the sandy depressions, however, Gleysols, Humic Gleysols, and Histosols are predominant (Fränzle, 2004). On average, the soils of the area are rated with 50 points according to the 100 point scale of the German soil taxation system (Horn et al., 2006). Conse-

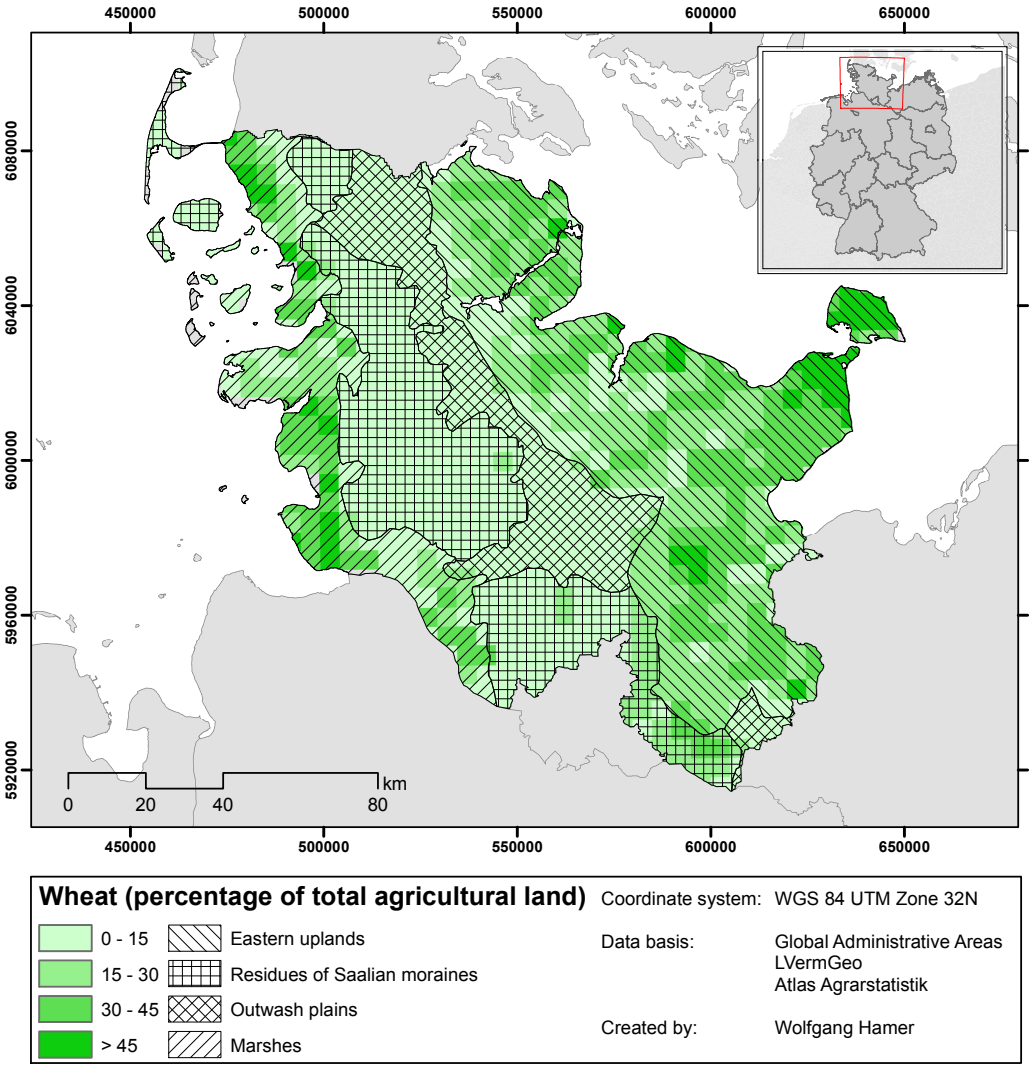


Figure 3.2: Wheat cultivation in 2016 and main natural regions in Schleswig-Holstein

quently, the soils in the eastern uplands are suitable for agricultural use, which is used to a high percentage for wheat cultivation (figure 3.2).

Glaciofluvial deposits, mainly composed by middle and fine sands, in the west of the last glaciations tunnel valleys are characteristic for the outwash plains surrounded by the residues of Saalian moraines in the west and the eastern uplands in the east (Meynen and Schmithüsen, 1962). Since vegetation only sparsely covered this area during the Weichselian glaciation, it is characterised by aeolian cover sands and dunes on top of the sanders. In the outwash plains, the primary soil formation processes are podzolization, gleyization, and peat formation (Horn et al., 2006). In accordance with these soil formation processes, the soil groups Podzols, Gleysols, and Histosols are characteristic for the outwash plains. Furthermore, bogs have developed in the

wide, flat depressions of the area (Fränzle, 2004). The soils of the outwash plains are rated with around 30 points according to the German soil taxation system (Horn et al., 2006). The lower quality of the soil could only be accommodated with the drainage of the bogs and the marl and artificial fertilisation of the agricultural soils. Nowadays grassland is still of greater importance for agricultural purposes in the outwash plains (Meynen and Schmithüsen, 1962).

The residues of Saalian moraines western of the outwash plains are composed of base moraines, terminal moraines and sand dunes mixed by cryoturbation to a mantle of sandy till (Fränzle, 2004). During the Holocene period, it developed into acid Cambisols, Podzols and Luvisols on dry summits and ridges. Stagnic Luvisols, Gleysols and Stagnosols are also frequently found in depressions and valleys (Fränzle, 2004). The average score of soils on the residues of Saalian moraines in the German soil taxation system is only slightly higher than on the outwash plains. Accordingly, the cultivation of winter wheat has minor importance in this area (figure 3.2).

The Marsh area western of the residues of Saalian moraines is characterised by alluvial tidal mud soil, consisting of fine-grained soils (Schlenger et al., 1969). It originated from the transgression of the North Sea during the Holocene. The topography of the entire area is flat and near sea level (Schlenger et al., 1969). Especially the young Calcaric Gleysols are loose and well ventilated and achieve a high soil quality, which is reflected in the average score of 60 points in the German soil taxation system and the proportion of wheat cultivation (figure 3.2) in the marshes (Fränzle, 2004).

Schleswig-Holstein can also be climatically classified according to the four different regions. The marshes can be categorised as part of the Atlantic climate zone, with western weather conditions determining the course of the meteorological elements most strongly. The climate diagram of Elpersbüttel (Figure 3.3) shows the highest rainfall from July to October, with a maximum of almost 100 mm in October. The lowest precipitation can be seen from March to May, which corresponds to the observations of Meynen and Schmithüsen (1962) for the marsh area.

While the temperature changes only slightly between the stations in the study area, variations in precipitation can be detected at the locations of the central area of Schleswig-Holstein. In the southern part (e.g. Itzehohe, Padenstedt) the lowest precipitation can be observed between March and May, but the highest precipitation is to be found in July. In the northern part (e.g. Schleswig-Jagel) an additional peak of precipitation in October can be seen, indicating the increasing effect of continental influence towards the south-east, as described by Meynen and Schmithüsen (1962), which is associated with lower rainfall.

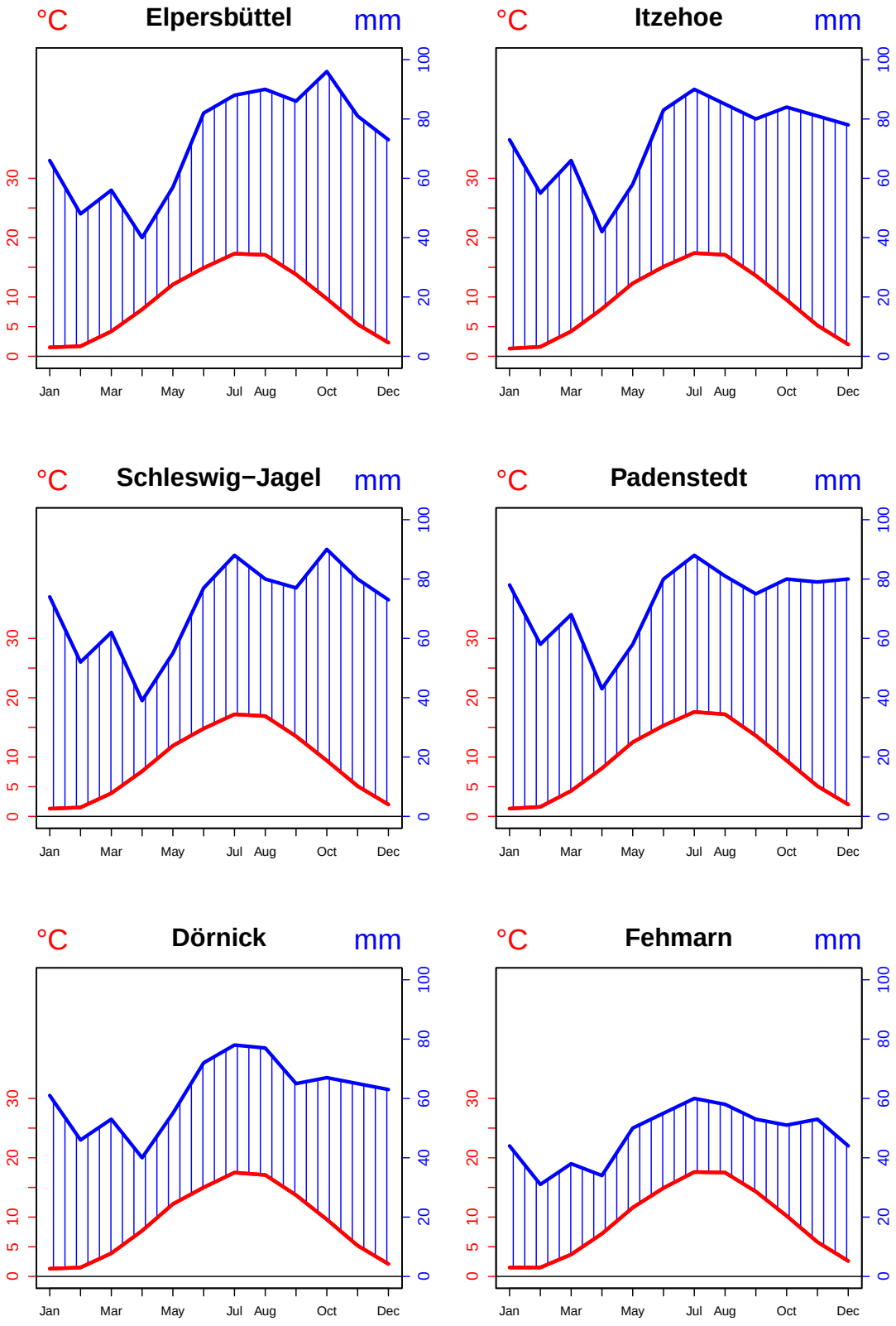


Figure 3.3: Climographs of German Meteorological Service locations in Schleswig-Holstein (data of the period 1981-2010) depicted in figure 3.1

Considerably lower precipitation quantities can be seen at the stations in the eastern uplands (figure 3.3). The monthly rainfall at Dörnick in a heterogeneously relieved region (figure 3.1) is always below 80 mm. In Fehmarn, in the plain area of the base moraine, the precipitation hardly reaches 60 mm in July, which corresponds to the descriptions of Meynen and Schmithüsen (1962), who explain that precipitation in the eastern uplands decreases rapidly to the east.

3.2 Data

Chapters 2.1.1 and 2.2.1 show, that the main influence on the development of *Blumeria graminis* f. sp. *tritici* and *Puccinia triticina* is the weather at the time of infection. The temperature, humidity, wind speed and precipitation are of special interest during the infection. Afterwards only the temperature influences the development of the incubation. This is also reflected by the empirical models described in chapters 2.1.3 and 2.2.3. Based on this knowledge the decision was made to use weather data to predict the infestation.

3.2.1 Meteorological and climate data

The climate and weather information was derived from the DWD. The DWD provides data of weather stations in SH on hourly, daily monthly and multi-annual basis via the Climate Data Center (CDC) FTP-server (`ftp://ftp-cdc.dwd.de/pub/CDC/`) or via the web interface WebWerdis (`https://werdis.dwd.de/werdis/start_js_JSP.do`). The data acquisition follows the official instructions presented by DWD (2015). As figure 3.4 shows, the station network providing hourly data is evenly distributed in and around SH. The number of the available weather stations provided by DWD changes over time. The variation during the years is displayed for SH and Northern Germany (NG) in table 3.1.

Table 3.1: Number of DWD weather stations in SH and NG and measured meteorological elements

Year	Temperature		Humidity		Precipitation		Wind	
	SH	NG	SH	NG	SH	NG	SH	NG
1996	8	41	8	41	8	35	16	57
2001	9	43	9	43	11	44	14	56
2006	23	94	23	94	19	80	15	56
2011	25	109	25	109	23	96	13	56
2016	24	105	24	105	23	92	14	59

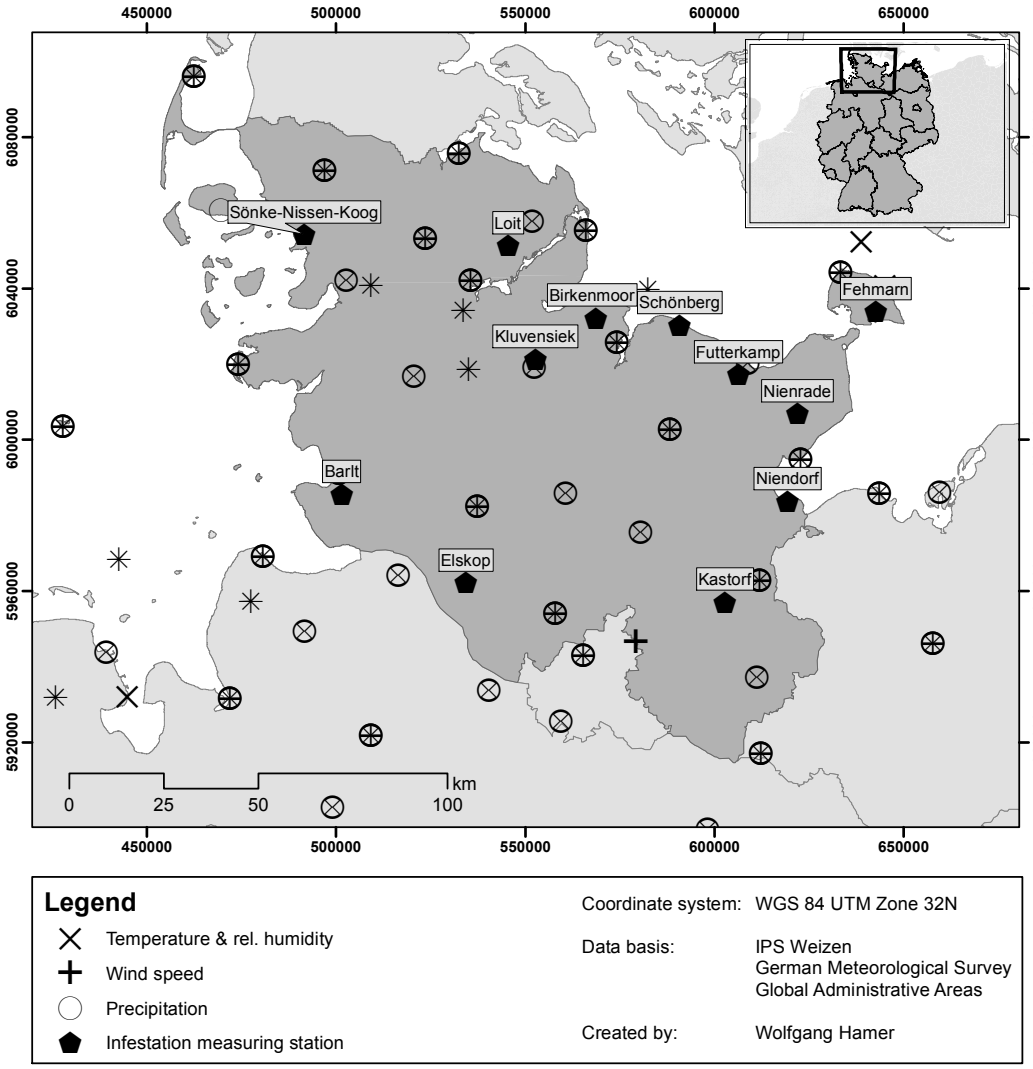


Figure 3.4: German Meteorological Service station network and infestation measuring stations of the Integrated plant protection system monitoring in Schleswig-Holstein

NG includes the weather measurement stations in Lower Saxony, Mecklenburg-Vorpommern, Bremen, Hamburg and Schleswig-Holstein. Additional regionalised data is available from the CDC including multi-annual climate data with a spatial resolution of 1 km x 1 km (figure 3.5). Apart from the DWD measurement stations other partner stations have been used for the regional multiple linear regression interpolation method using longitude, latitude and elevation information as independent variables described in DWD (2016a). The CDC allows inter alia the access to soil and air temperature, precipitation and wind parameters, the count of frost and hot days and evapotranspiration and drought indices, partly used to generate figure 3.3. Frost days are defined as days with a daily minimum temperature below 0 °C, while hot days are

days with a daily maximum temperature above or equal to 30 °C (DWD, 2016b). The evapotranspiration is calculated using the AMBAV model of the agrometeorological research center in Braunschweig for grass above sandy loam (DWD, 2016a; Löpmeier, 2014) and the drought index is calculated using a modified version of the Standardized Precipitation Index (DWD SPI) (Pietzsch and Bissolli, 2011).

3.2.2 Long term infestation monitoring data (IPS)

For the use of supervised machine learning algorithms, as suggested in chapter 2.5, it is necessary to access not only the weather situation but also the infestation situation. For this purpose data of the IPS wheat model monitored by the Department of Phytopathology of the University of Kiel was used. The infestation situation in Schleswig-Holstein was monitored from 1993 to 2017, except for 2004, at the locations shown in figure 3.4. Over the years the number of monitored locations varies from 4 in 2013 to 12 in 1996 with a mean of 9 monitored locations. As described by Klink (1997), Verreet et al. (2000) and Engel (2015), monitoring includes weekly sample of:

- 30 plants regularly treated with fungicides to keep them free of pathogens
- 30 plants treated if the pathogen specific damage threshold is exceeded as described by Klink (1997)
- 30 untreated plants

during the summer months. The untreated plants showed significant yield losses, while the other varieties did not show any significant differences (Engel, 2015). In addition to the yield all the plants were analysed for visible signs of infestation with *Blumeria graminis*, *Puccinia recondita*, *Puccinia striiformis*, *Septoria tritici*, *Septoria nodorum* and *Drechslera tritici-repentis*. Based on this analysis the disease severity and disease incidence were determined for the seven upper leaf levels and the overall plant. From 1995 onwards this monitoring took place for the winter wheat variety Ritmo with a susceptibility to powdery mildew of 5 and a susceptibility to brown rust of 8 (Bundessortenamt, 2017). Table 3.2 shows other varieties tested across all years. Although several varieties were only monitored in single years, the combination covers the whole range of susceptibility values considering powdery mildew and brown rust.

Figure A.1 in appendix A shows the count of different samples for each powdery mildew susceptibility class next to the proportion of observed exceedances of the 70 % disease incidence threshold explained in chapter 3.3. Thereby the class 0 covers the

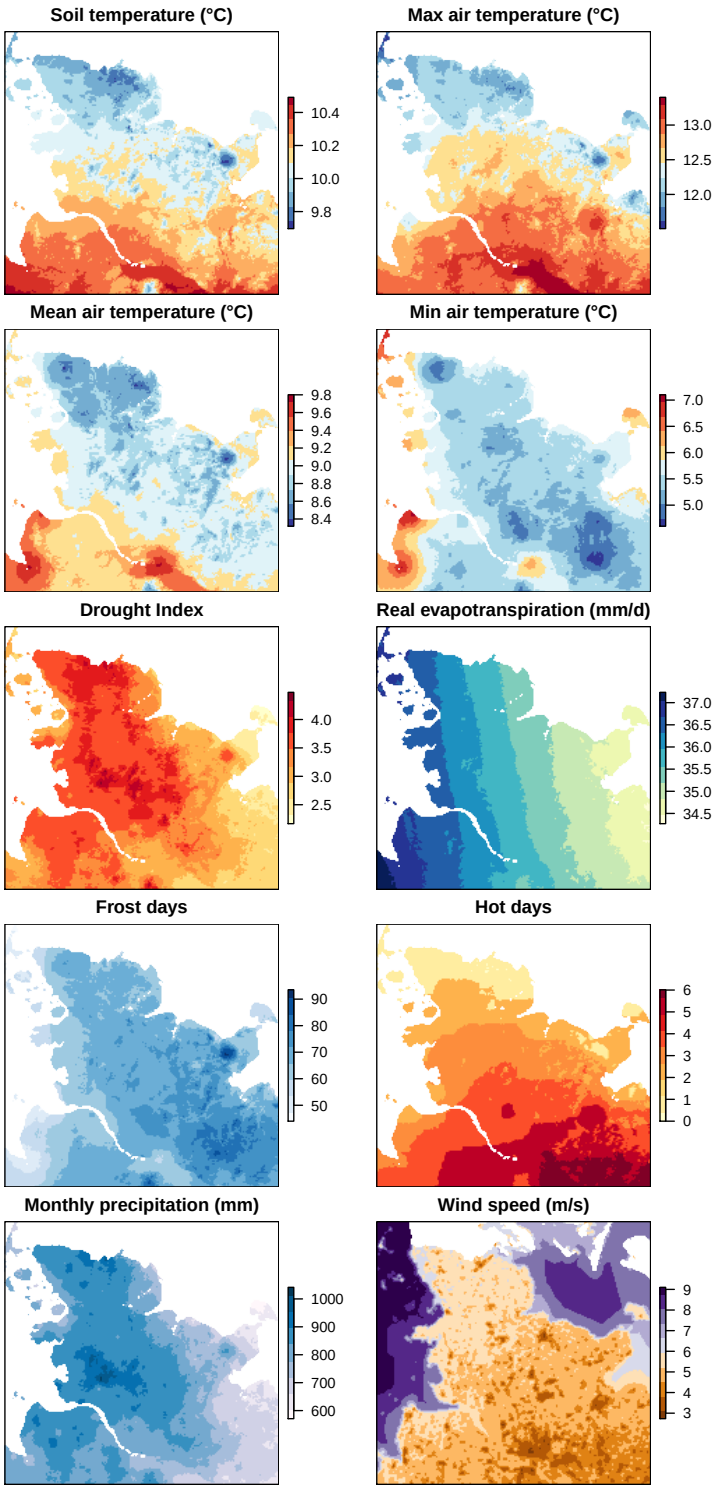


Figure 3.5: Regionalised climate data and indices of the Climate Data Center

Table 3.2: Wheat varieties monitored over the years including susceptibility (SUS) to powdery mildew and brown rust. Based on Bundessortenamt (2017)

Wheat varieties	SUS powdery mildew	SUS brown rust	Monitoring Years
Kanzler	9	Not listed	1994 - 1999
Kraka	Not listed	Not listed	1994 and 1995
Orestis	Not listed	Not listed	1994
Pepital	Not listed	Not listed	1994 and 1995
Xanthos	Not listed	Not listed	1994
Zentos	2	9	1994 - 1997
Contra	6	5	1995
Toronto	8	8	1995
Hanseat	Not listed	Not listed	1998
Cardos	2	3	1999
Flair	4	6	1999 and 2000
Aspirant	3	4	2000
Drifter	3	5	2000
Dekan	1	8	2002 and 2015 - 2017
Asano	3	5	2011
Julius	4	4	2011
Inspiration	3	5	2012 - 2017

wheat varieties which are denoted as *Not listed* in table 3.2. Figure A.2 shows the same for the brown rust observations with a threshold of 30 % disease incidence.

The usual infestation situation of untreated Ritmo in SH is displayed in figure 3.6. The multi-annual observed disease incidences were averaged on the week of the year for each location in a first step. In a next step, these averaged values were averaged again to create figure 3.6. The locations of the east coast in the eastern hilly region (compare figure 3.2) show higher powdery mildew disease incidences than the locations of the western coast. This can be seen in figure A.3 in appendix A as well. This heatmap shows the observed exceedances of the 70 % disease incidence threshold. A dark red colour expresses many high disease incidences during the year while a green colour implies no threat by the powdery mildew at the specific location and year for Ritmo wheat. Figure A.3 shows, that even during a year with high powdery mildew infestation like 1996, a yield threatening event only occurred in Barlt, whereas other western locations like Sönke-Nissen-Koog and Elskop displayed no threat.

Such a trend cannot be identified for the brown rust. East and west coast locations show no clear difference of the brown rust disease incidence as a comparison with figure A.4 shows. Altogether fewer observations show a dangerous brown rust infestation. If many of such infestations occur, they can be identified at each study site, as the year

2007 shows. In addition the infestations behaviour over time is expressed in the figures A.5 and A.6. The IDW method ($p = 4$) was used to interpolate the disease incidences averaged according to the week of the year for every second week. It can be observed, that the powdery mildew infestations (figure A.5) show high disease incidences over the whole monitoring time. The brown rust infestations (figure A.6) are not as severe as the powdery mildew infestations. This can be seen in figure 3.6 as well. Additionally, the average brown rust disease incidences begin to rise late in the monitoring period from the 23rd week on.

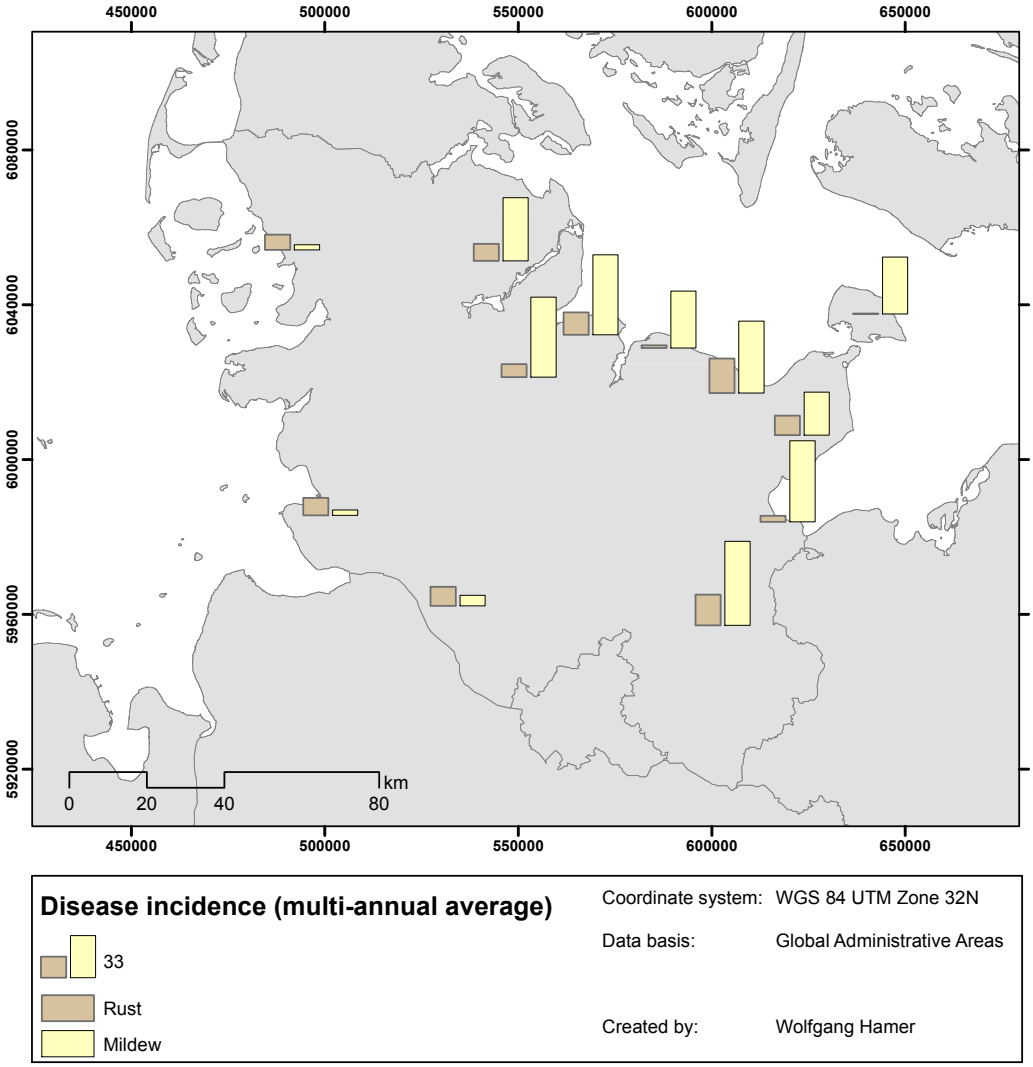


Figure 3.6: Multi-annual averaged observed disease incidences of powdery mildew and brown rust in Schleswig-Holstein

3.3 The modelling approach

The model should aim at giving advice, whether or not to use fungicides for more purposeful use of these, considering:

- a spatial interpolation of meteorological data influencing the pathogens
- a temporal aggregation representing the fungal life cycle
- the creation of a machine learning model to detect a defined disease incidence threshold, that indicates the need of fungicidal treatment as mentioned in chapters 2.1.3 and 2.2.3

Käsbohrer et al. (1988) and Klink (1997) suggest a disease incidence threshold of 70 % for infestations with powdery mildew and of 30 % for infestations with brown rust. Therefore the model's purpose should be, to predict the exceedance of these thresholds in SH based on current weather data. Such a prediction would give farmers time to react prior to an increasing infestation and it would reduce the number of unnecessary preventive fungicide treatments.

Based on the models aim, the pathogens life cycles (chapters 2.1 and 2.2), the available data (chapters 3.2.1 and 3.2.2) and the strategy to use a machine learning predictive approach (chapter 2.5) the model displayed in figure 3.7 was created.

As a first step the local weather data are downloaded from the FTP-sever and interpolated in R (described in B.2). Afterwards the regionalised data of several time units are summarised and combined with the already interpolated climate and the local infestation data. Finally the combined dataset was transferred to a machine learning algorithm which creates a prognosis model for the prediction of the probability of an exceedance of the pathogens disease incidence threshold.

3.3.1 Regionalisation of weather data

The first step of the model generation is the regionalisation of weather data in the best possible way. The different approaches for interpolating meteorological data have already been listed in table 2.3 on page 22. The methods described there have also been tested for the weather data presented in chapter 3.2.1 (table 3.3). These methods were used to regionalise the variables for the hours of the first three days of every third month. The two deterministic methods Thiessen polygons and IDW were used as well as the three statistic methods OK, UK and KED. For the UK method only the longitude and latitude information were used as independent variables. For the KED method

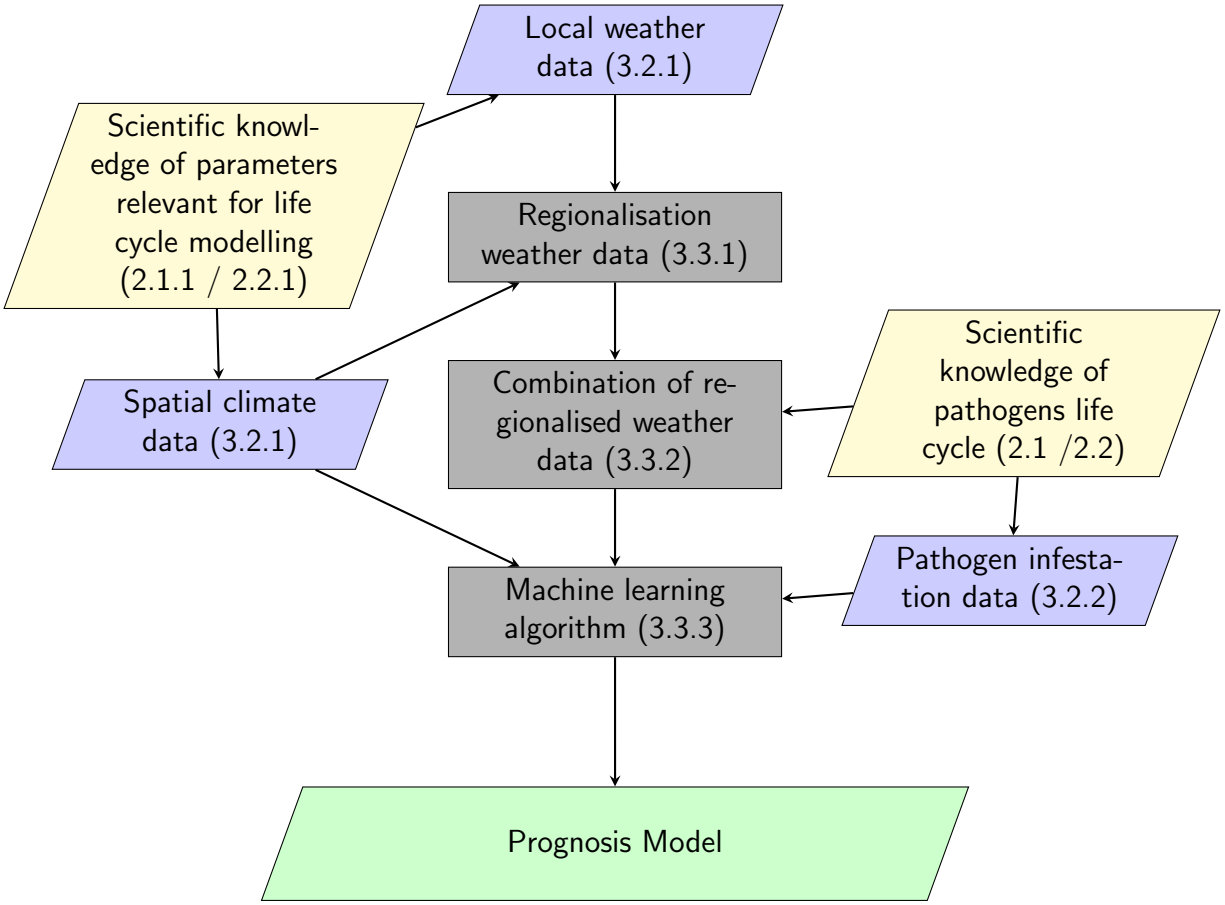


Figure 3.7: Expert knowledge (yellow), input data (blue), processes (grey) and result (green) of the modelling approach including the chapters describing these components

the three combinations KED1 (longitude, latitude and elevation information), KED2 (longitude, latitude and climate information) and KED3 (longitude, latitude, climate and elevation information) were compared. For the temperature interpolation the mean air temperature of the CDC (chapter 3.2.1) was chosen as climate variable. For the interpolation of windspeed the mean wind speed, for the precipitation the average precipitation and for the relative humidity the drought index were used. Following these grids, depicted in figure 3.5, the interpolation was calculated with a spatial resolution of 1 km x 1 km. Also, the 30 m x 30 m elevation SRTM, distributed by the Land Processes Distributed Active Archive Center (`http://lpdaac.usgs.gov`), was resampled to this resolution. The interpolation itself was progressed using the function 'autoKrige' of the library 'automap' (Hiemstra et al., 2008) with the R programming environment (R Core Team, 2016).

For the validation a leave-one-out cross-validation was used. Each interpolation is repeated multiple times omitting one location each time allowing a comparison of

predicted and observed values (Zhang, 1993). Such a comparison can be made by the root-mean-squared-error:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(\hat{z}(x_i) - z(x_i))^2}{n}} \tag{3.1}$$

with $\hat{z}(x_i)$ as predicted and $z(x_i)$ as observed value at location x_i (Armstrong and Collopy, 1992). For a better comparability the RMSE can be normalised to form the NRMSE by the range of the observed variable $(x_{max} - x_{min})$:

$$NRMSE = \frac{RMSE}{x_{max} - x_{min}} \tag{3.2}$$

Table 3.3: NRMSE of the interpolation methods

	Thiessen poly.	IDW	OK	UK	KED1	KED2	KED3
Temperature	0.199	0.172	0.192	0.172	0.172	**0.146**	0.151
Humidity	0.492	0.417	0.420	0.405	0.415	**0.401**	0.420
Windspeed	0.724	0.612	0.625	0.604	0.604	**0.526**	0.549
Precipitation	1.066	**0.843**	1.186	1.214	1.153	1.165	1.187

Following the results, presented in table 3.3 the temperature, humidity and wind speed data of the available years were interpolated using the KED2 method and the precipitation data were regionalised using the IDW approach. The technical implementation in R is described in appendix B.3.

3.3.2 Aggregation of weather data

Since the observed disease incidence does not only result from the actual weather situation, it is necessary to relate those measured disease incidences to the foregone weather conditions. As described in chapter 2.1.1, the highest influence of the weather on infestation happens during sporulation and infection, before the spore is metabolic dependent on the host. The disease incidence does not indicate the completed infections but the completed incubations. Therefore weather data recorded directly before the measurement of the disease incidence represent the period of the potential incubation. Following the infection chain of *Blumeria graminis*, presented by Hau (1985) (chapter 2.1.3), the weather data before this period, representing the sporulation and infection are of interest, predicting the disease incidence. Consequently, the duration of the sporulation and infection period and the incubation period must be determined. This periods should be based on the average temperatures in SH (figure 3.3) in the time period of interest combined with the functions of Friedrich (1995b,c) delineated

in figure 2.2 and 3.8. The average temperature in the relevant months ranges from 7.8 °C in April to 17.4 °C in July. This results in an average infection period of 3.3 d and an average incubation period of 8.8 d (figure 3.8). Therefore the weather data of three days are aggregated and related to the disease incidences measured nine days later. The aggregation includes the minimum, the maximum and the arithmetic mean values of precipitation, wind humidity and temperature data. For the brown rust different time intervals result from the differing life cycle, presented by Hau and de Vallavieille-Pope (2006) (chapter 2.2). Following Roelfs et al. (1992) an incubation period of 10 d is assumed. The infection period, however, is shorter than a day. Since the aggregated weather data are linked to one day, this is the shortest considered time interval. Consequently, for the brown rust, an infection period of 1 d is assumed.

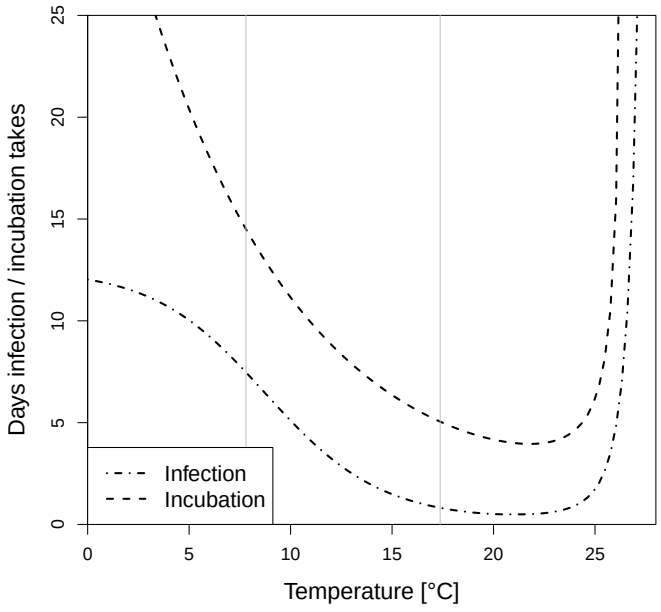


Figure 3.8: Infection and incubation days depending on the temperature according to Friedrich (1995b,c)

Additionally a value derived from all previous temperature values is included, representing the potential development of the host plant. Chapter 2.3.1 outlined the field of crop models and the data basis, necessary for an accurate crop model. The method with the smallest possible number of input variables is the CERES-wheat model developed by Ritchie et al. (1988) based on the accumulation of daily thermal time. Similar to Ritchie et al. (1988), Soltani and Sinclair (2012) presented functions to calculate the Daily Thermal Unit (DTU) as a function of the temperature, which is weighted according to the potential productivity of the wheat at this temperature (figure 3.9).

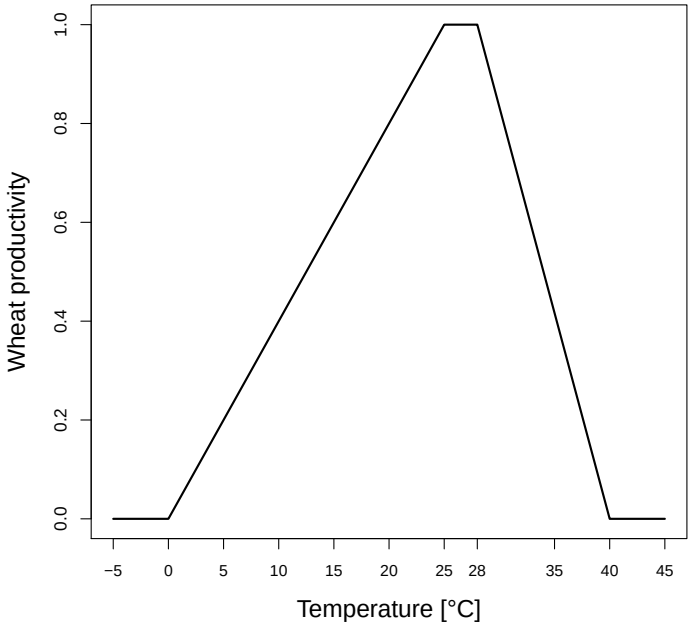


Figure 3.9: Productivity of winter wheat as a function of daily temperature according to Soltani and Sinclair (2012)

After the daily temperature has been multiplied with the value of wheat productivity to derive the DTU, it is summed up to the Cumulative Thermal Unit (CTU) beginning with the first October of the previous year. At the beginning of October, winter wheat is usually sown in the study area Schleswig-Holstein, as explained in chapter 2.3 and illustrated in figure 2.3.

3.3.3 Prediction based on an machine learning algorithm

A number of factors influences the disease incidence of powdery mildew and brown rust on winter wheat. Among these are hourly weather data as well as multi-annual climate data and indices.

As pointed out in chapter 2.5, supervised machine learning techniques can be trained to find connections between the independent variables and the target variable disease incidence. Following the studies of Käsbohrer et al. (1988) and Klink (1997) (chapter 2.1.2) this work aims at the identification of a disease incidence of more than 70 % for powdery mildew and of 30 % for brown rust. Therefore a machine learning algorithm using a classification prediction is appropriate. These include k-NN, Naive Bayes and Decision Tree methods. Since the Naive Bayes classification is adapted to

text classifications, as mentioned in chapter 2.5, the following k-NN and Decision Tree methods are tested for their usage:

- *k*-Nearest Neighbor (k-NN)
- Decision Tree (DT)
- Boosted Decision Tree (BDT)
- Random Forest (RF)

k-Nearest Neighbor (k-NN) prediction The k-NN approach, also shown in appendix B.5, was tested using the class package in R (Venables and Ripley, 2002). The first step of the prediction procedure is the use of the *normalize* function on the independent variables of the dataset, containing the weather and multi-annual climate data as independent variables and the infestation data, classified by the threshold of 70 % or 30 % respectively, as target variable (table 3.4). As mentioned in chapter 2.5 this step is necessary, because the prediction results from the Euclidean distance of the values. Without normalisation, this would lead to a larger influence of variables with higher values.

The second step of the prediction is the application of the function *kv_fit*. It is used to evaluate the best number of neighbours (k) to be considered during the classification process. The function works by separating the dataset into two halves. One of the halves is now used to predict the factorised disease incidence of the other half iterating through different k-values.

Finally, the function returns the number of neighbours for which the highest ROC AUC (page 42) value was obtained. This amount of neighbours is used for the prediction fitted to the whole dataset. The k-NN approach does not create an interpretable model which could be analysed or applied to another dataset. The prediction always requires the dataset.

Decision Tree (DT) prediction As also shown in the appendix (B.5), the decision tree approach was evaluated using the C50 package in R (Kuhn et al., 2015). In contrast to the k-NN approach, it is not necessary to determine an appropriate k value for the decision tree approach. However, it is possible to define, which error is "costlier" in a decision tree prediction. Since there is a larger threat to the farmer by an unpredicted exceedance of the threshold than by a wrong prediction of an exceedance, this error was defined as "costlier" using a cost matrix in an early approach. This idea was discarded as the use of a cost matrix excluded the output of the probability of

Table 3.4: Parameters considered for the machine learning algorithms

Data category	Parameter	Label	Chapter
Phytopathogenic	Disease incidence Pathogen VUL	IBHBF Susceptibility class	3.2.2
Elevation	Resampled SRTM	Elevation	3.3.1
Climate (multi-annual)	Soil temperature	Soil temperature (C)	3.2.1
	Air temperature	Max temperature (C) Mean temperature (C) Min temperature (C)	
	Drought Index	Drought Index (C)	
	Real Evapotranspiration	Real Evaporation (C)	
	Frost days	Frost days (C)	
	Hot days	Hot days (C)	
	Monthly precipitation	Monthly precipitation (C)	
	Windspeed	Wind speed (C)	
Aggregated weather (daily)	Air temperature	Min temperature Max temperature Mean temperature	3.3.2
	Windspeed	Min wind speed Max wind speed Mean wind speed	
	Precipitation	Min precipitation Max precipitation Mean precipitation	
	Humidity	Min humidity Max humidity Mean humidity	
	Daily Thermal Unit	DTU	
	Cumulative Thermal Unit	CTU	

predictions. Also, potential overfitting of the decision tree model to the calibration dataset should be avoided by the definition of a minimum number of elements that need to be in one of the branches of the tree. In contradiction to the k-NN approach mentioned above a decision tree model is created, which can be stored and applied to other data sets. The results of this evaluation are described in chapter 4.

Boosted Decision Tree (BDT) prediction In addition to the approach to calculate one single decision tree, the C50 package offers the option to create an BDT following an approach similar to the adaptive boosting procedure of Schapire and Freund (2012) using the *trials* parameter of the function *C5.0* (Kuhn et al., 2015). Although the the *C5.0* function also calculates several DTs, the approach differs from the RF approach. While the RF algorithm creates random subsets of the original data, the *C5.0* function

uses all available data to create a decision tree. Then it creates another decision tree, giving the false classifications of the previous tree more weight. This is repeated until the number of trees, defined by the trials parameter is reached. For the BDTs prediction the majority decision of all trees is used (Schapire and Freund, 2012). The optimum number of trials is evaluated using the function *evtrials*. Similar to the function *kv_fit* for the k-NN approach *evtrials* searches for the number of trees which receive the largest ROC AUC.

Random Forest (RF) prediction Finally an even further developed approach to DTs, the RF, was evaluated using the RandomForest package (Liaw and Wiener, 2002). Again the iterative selection of calibration and validation years displayed in figure 3.10 has been used (B.5). For each created model 1,000 trees were generated. From this number on, the OOB error did not change in the first tests. Like the cost matrix of the decision tree algorithm the RF can be fitted to more important predictions using the *classwt* parameter of the *randomForest* function (Liaw and Wiener, 2002). With this parameter, the exceedances can be weighted more than the underruns. The optimum weight can be evaluated using the function *evclasswt* which again searches for the weight with the largest ROC AUC. In addition to the weighting, the *evclasswt* function searches for the most suitable number of features that are selected randomly at each node. The results of this evaluation are described in chapter 4.

3.3.4 Assessment of overall model performance

In order to assess the performance of the machine learning methods, four different concepts have been applied:

- Holdout validation
- LOOCV validation
- Estimation of the number of years and locations required
- Real time modeling of infestation risks in 2017

Holdout validation procedure The holdout procedure is a standard method for the validation of machine learning models (Langley et al., 2017; Steger et al., 2016; Timm and McGarigal, 2012). During the process, the dataset is separated randomly into calibration and test datasets (Table 3.4). The random selection is made in such a way that the calibration datasets include two-thirds of the time-series and the test

dataset one-third of the series. This process is repeated until 200 combinations of different years are used for the evaluation of the predictions described below. 200 combinations represent 0.25 % of all possible combinations of the mentioned size. The holdout procedure is presented in the process scheme 3.10.

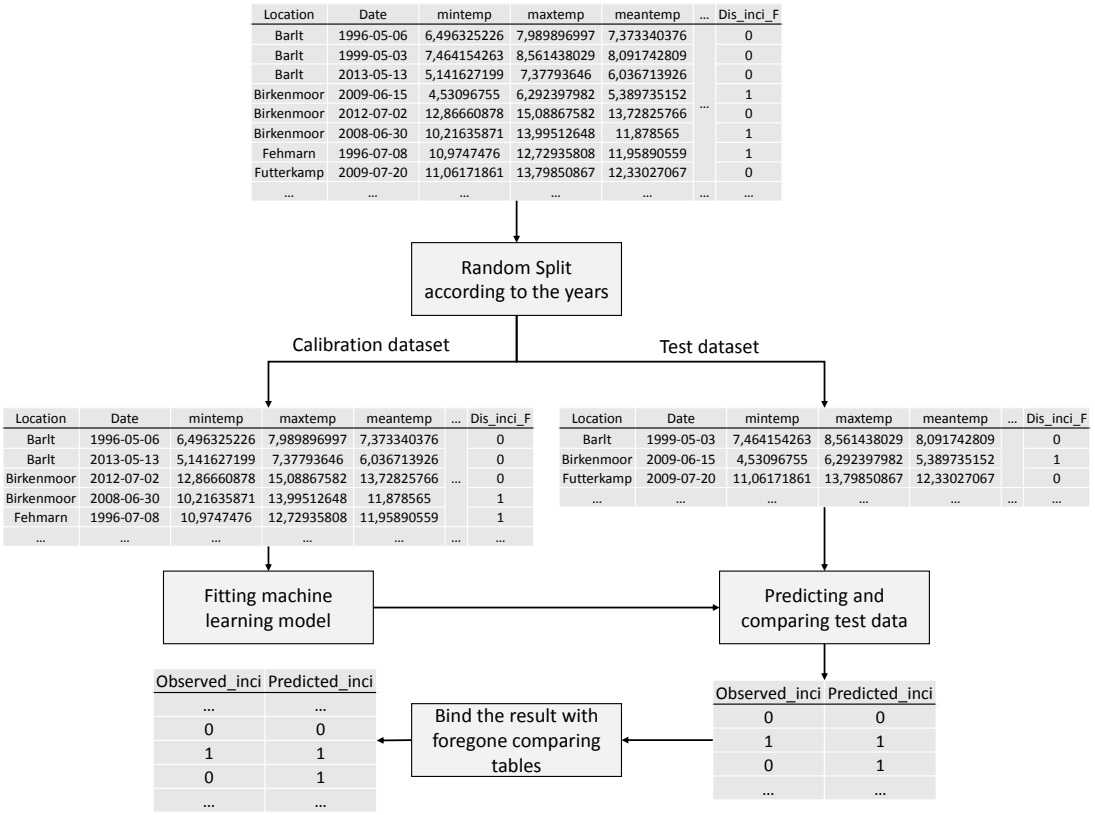


Figure 3.10: Process scheme of the holdout validation

LOOCV validation procedure As described in chapter 3.3.1, leave-one-out cross-validation is a common method for validating the predictive accuracy of a regionalisation method. As the name of the methods implies, LOOCV is an iterative holdout validation. For each iteration, one location is left out until for each site comparison of predicted and observed data is available (Zhang, 1993). The method can be adapted to the iterative model comparison explained before. As shown in figure 3.11 and described as code in the appendix (B.5) again the original dataset is split randomly according to the years. In a second step, the test dataset is reduced to only one location which is also removed from the calibration dataset. A model is fitted to the calibration dataset, ignoring the location left in the test dataset. The model is used to

predict the probability of exceeding the disease incidence threshold at the test data sets location. The complete process is repeated until each location of the test dataset was once removed from the calibration dataset and predicted using the test dataset. Then the next random split of the original dataset occurs and the procedure is repeated 200 times. In the end, each prediction was not only made using the infestation information of different years but also using the infestation information of different locations.

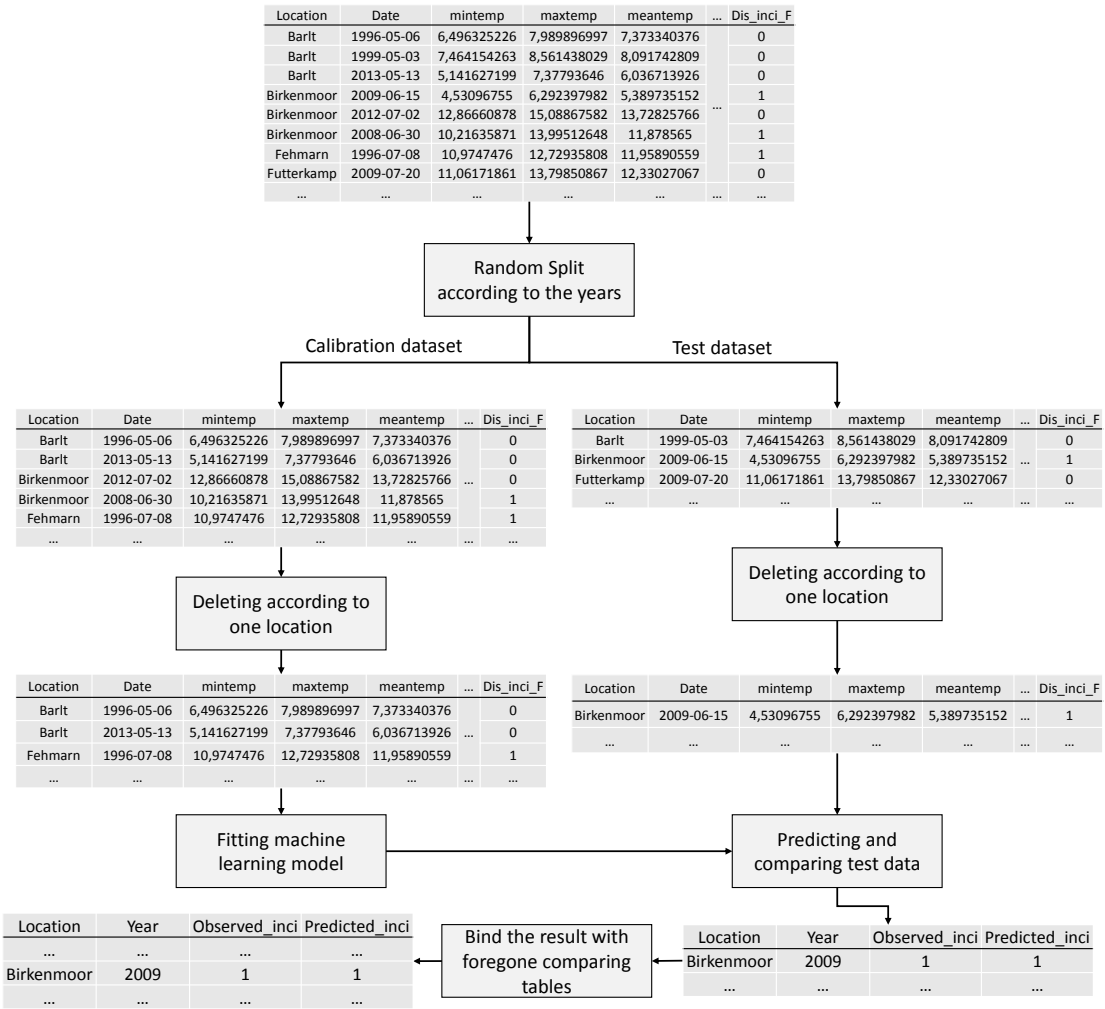


Figure 3.11: Process scheme of the leave-one-out cross-validation

Estimation of the number of years and locations required The LOOCV evaluation showed the behaviour of the machine learning methods using a maximum of available data of years and locations which are not part of the validation dataset. The test described below evaluated the reliability of the approaches under differing data availability conditions. To do so, as described for the other evaluations, the origi-

nal dataset was separated 200 times into each a calibration (two-thirds of the years) dataset and a validation (one-third of the years) dataset. For each of the 200 passes, the data set was iteratively broken down again based on the number of locations and years. Of the calibration dataset one year was randomly selected and of the sites, available for this year, one was chosen randomly. The machine learning algorithms were trained on this dataset and tested on the validation dataset. On the next iteration, two locations of the randomly selected year were selected randomly and so on, until all sites were chosen to predict the validation dataset. This procedure was repeated iteratively for more years until again all years and locations were used for the prediction. The process, as displayed in figure 3.12 and described in the script (appendix B.5), aims at the identification of the number of sites and years, needed for appropriate predictions. Since this evaluation method similar to the LOOCV method is quite time-consuming the packages foreach and doSNOW were used to allow parallel processing of the iterative processes (Revolution Analytics and Weston, 2015b,a).

Real time modelling of infestation risks in 2017 Finally, the different machine learning procedures were exemplarily fitted on all data available for the years until 2017 and then applied to the year 2017. Thereby the procedure followed the process scheme shown in figure 3.7. First, the local weather data were downloaded from the DWD as shown in B.2. Then the local weather data was regionalised using the deterministic IDW and the stochastic KED method following B.3 and including the spatial climate data described in chapter 3.2.1.

The regionalised weather data was combined based on the expert knowledge of the pathogens as described in chapter 3.3.2. That means for powdery mildew the weather data of three days is linked to a date nine days later. For brown rust, the data of one day is combined and linked to a day ten days later. Additional the approach of Soltani and Sinclair (2012) was used to derive the DTU and the CTU based on the regionalised temperature data. The aggregation and the calculation of DTU and CTU are described in B.4.

For the years from 1996 to 2016 the pathogen's infestation data was combined with this aggregated weather data. Using the different machine learning approaches, models were created, predicting the exceedance of the 70 % disease incidence for powdery mildew and of 30 % for brown rust. Only for the k-NN approach no model could be created. As described in chapter 3.3.3 all available data needs to be used for each prediction utilising the k-NN approach.

Before these models could be applied on the spatialised weather and multi-annual climate data of 2017 to create daily predictions of the probability of yield relevant

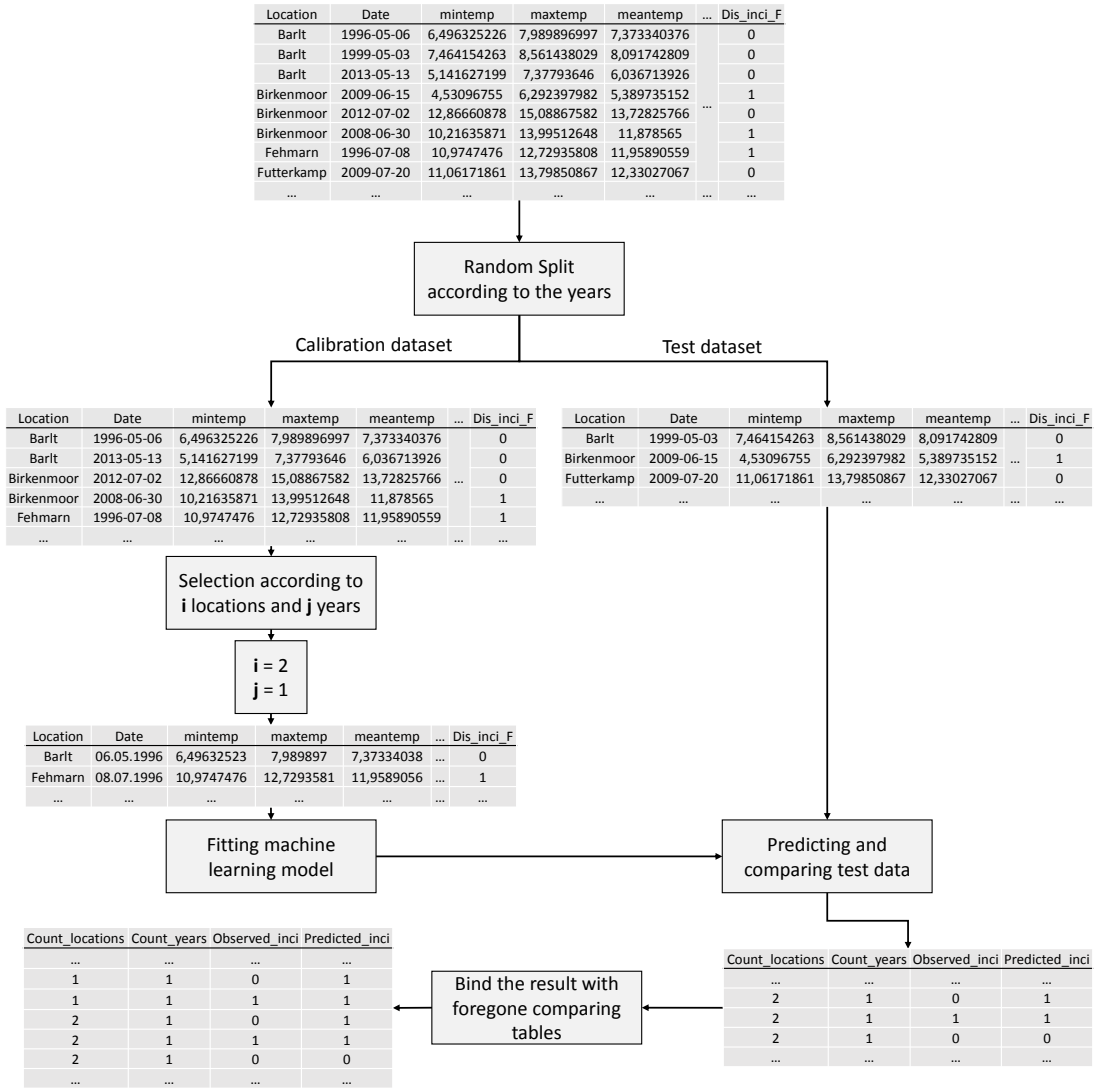


Figure 3.12: Process scheme of the estimation of the number of years and locations required for a reliable prediction

infestation events of powdery mildew and brown rust on winter wheat, the susceptibility class of interest needed to be selected. Since most of the data are available for Ritmo and this wheat variety is also available for 2017, the spatial and temporal predictions also were made for this variety. Therefore a susceptibility of 5 was selected for the prediction of powdery mildew and susceptibility of 8 was selected for brown rust. As the code in appendix B.6 shows, the *predict* function was applied to the data extracted for the 7 observation locations displayed in figures A.3 and A.4, as well as on the regionalised raster data for whole Schleswig-Holstein. Only for the k-NN approach a little more cumbersome way had to be taken especially for the spatial prediction.

3.3.5 Web-based prediction system

The modelling approach presented in figure 3.7 has been transformed into a web-based prediction system to make the forecasts accessible to a broad public. The operating steps of this system are illustrated in figure 3.13.

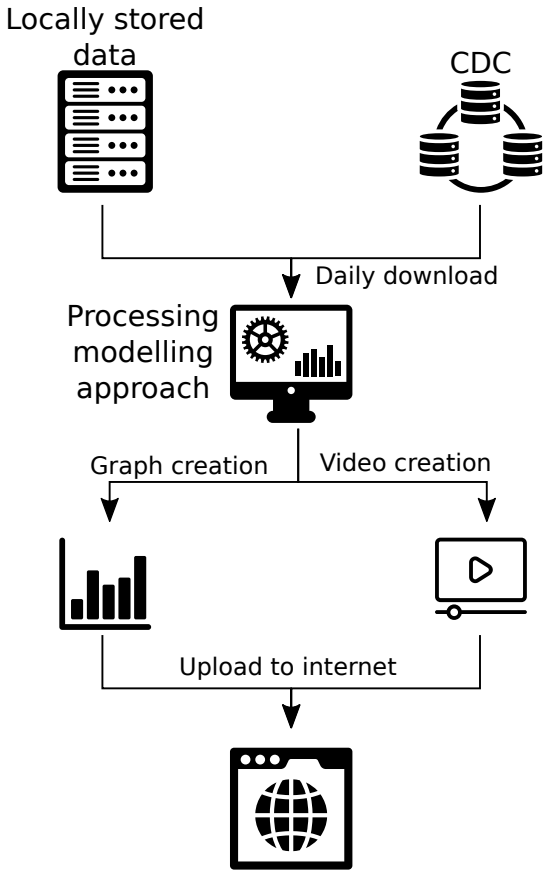


Figure 3.13: Representation of daily data processing (Icons created by Vectors Market from the Noun Project)

The scripts mentioned above for downloading, interpolating and summarising the weather data as well as the scripts for applying the model to it and the climatic data were executed automatically on a local computer every morning. Similarly, video files were automatically generated from the raster files generated in this way, showing the temporal and spatial behaviour of the forecasts on a daily scale. Besides, images were generated showing the temporal progression of the predicted probabilities at the sampling sites. In a final step, these images and the video files were uploaded to make them freely accessible. In order to inform farmers about this new procedure and the website, articles have also been published in the farmers' magazine *Bauernblatt Schleswig-Holstein* (Hamer et al., 2016a, 2017).

The whole system is integrated into the IPS wheat web environment of the Department of Phytopathology of Kiel University (`www.ips-weizen.de`). The websites, presenting the predictions results, are located on the authors GitHub account (`https://whamer.github.io/blugra.html` and `https://whamer.github.io/puctri.html`). Figure 3.14 shows an exemplary part of the website for predicting yield-relevant brown rust events. It is written in a combination of CSS and HTML and makes the predictions available as .png and .mp4-files. To ensure a better overview, only the RF models were used for prediction.

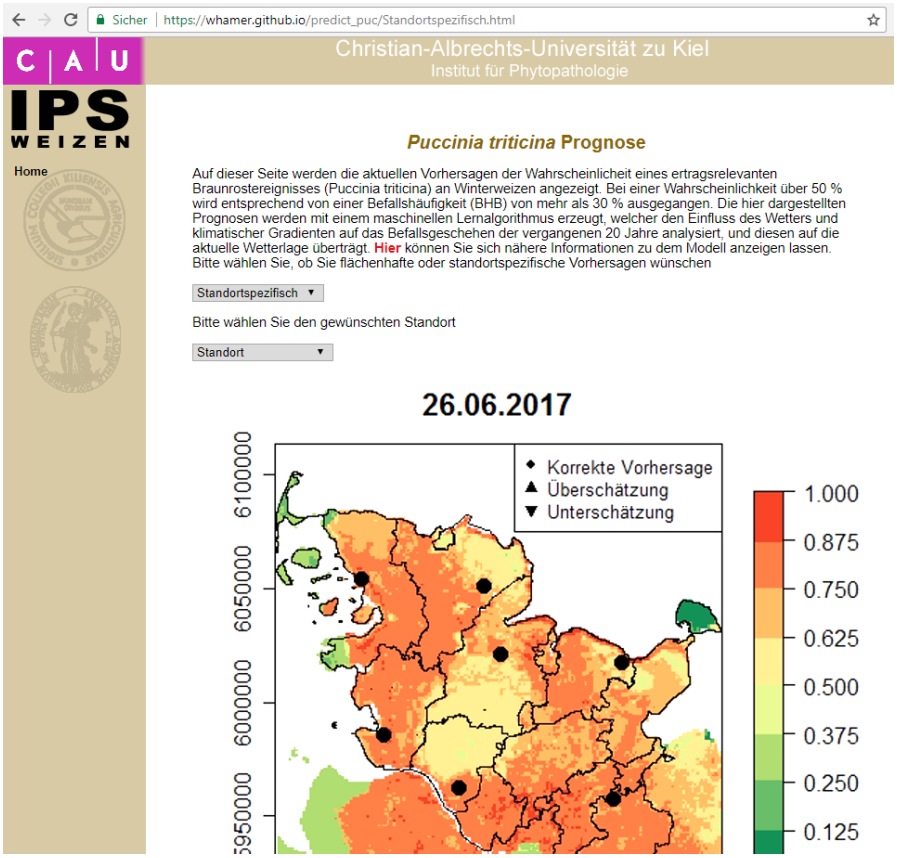


Figure 3.14: Impression of the web environment

Chapter 4

Results

4.1 Spatio-temporal prediction of powdery mildew events

In the first part of this chapter, the results of the forecast of powdery mildew events relevant to yield are shown. The results represent the four different validation concepts described in chapter 3.3.4. Table 4.1 and figure 4.1 depict the results of the holdout validation for powdery mildew.

Table 4.1: Performance of powdery mildew models using the holdout validation

Method	Accuracy	Sensitivity	Specificity	Precision	ROC AUC
k-NN	0.71	0.38	0.82	0.43	0.68
DT	0.73	0.40	0.84	0.47	0.73
BDT	0.72	0.42	0.83	0.46	0.73
RF	0.64	0.76	0.60	0.40	0.72

Table 4.1 shows the performance of the iterative predictions of the different machine learning methods using the statistical measures of the models' performances (chapter 2.6). The DT approach received the highest overall accuracy for the prediction following the holdout evaluation method of powdery mildew. The lowest accuracy as well as the highest sensitivity was achieved in the prediction of the RF approach. The sensitivity values of the other approaches are much lower reaching only half of the value of the RF approach. Not even half of all observed exceedances of a damage threshold have been predicted neither by the k-NN, DT nor the BDT approach. The specificity values behave similar to accuracy, with high values for DT, BDT and k-NN-method and lowest values for the RF method. With the RF approach, therefore, underruns of the damage threshold were more likely to be erroneously predicted as

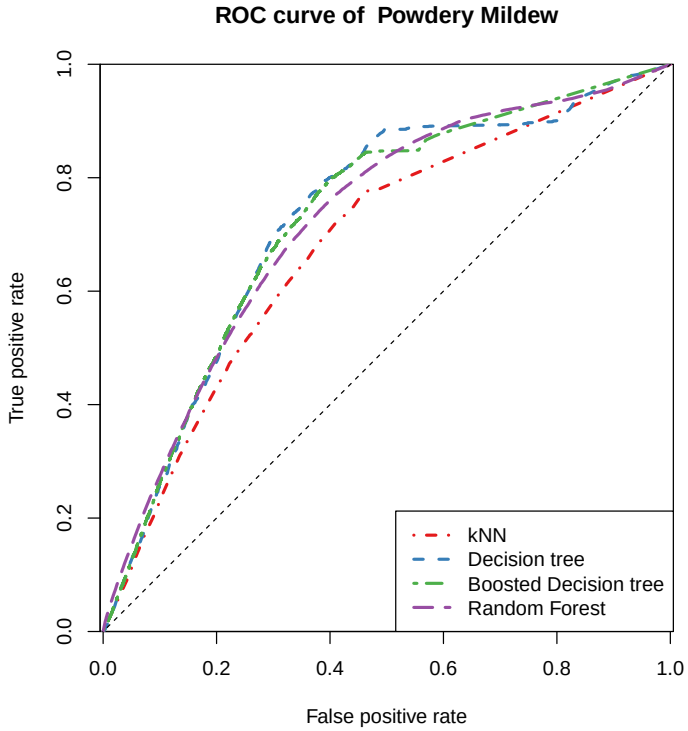


Figure 4.1: Receiver Operating Characteristic curve of the holdout validation for powdery mildew

incidents endangering yield than with the other procedures. The precision only shows little difference between the DTs and BDTs. The k-NN and RF methods show lower values. The precision of all methods is lower than 0.5. Corresponding to this, more than half of the predicted exceedances of any method were observed as underruns of the threshold. As also depicted in figure 4.1 the DT methods received the highest ROC AUC value closely followed by the RF predictions. Since the false positive rate of the ROC curve is equivalent to the term *1 - specificity* the curve shows the trade-off between sensitivity and specificity (page 42). An increase in sensitivity will come along with a decrease in specificity until all exceedances are predicted correctly and all underruns are predicted false. Figure 4.1 shows that the k-NN approach results in a lower true positive rate than the other methods at the same false positive rate. The ROC AUC values of table 4.1 reflect this impression.

Table 4.2 and figure 4.2 show the results of the LOOCV evaluation for powdery mildew. The BDT method resulted in the highest accuracy in contrast to the holdout evaluation method where the DT algorithm received the highest accuracy. While the accuracy of the RF prediction did not change in comparison with the holdout method, the accuracy of the other methods were reduced for the LOOCV method. Also the

values of sensitivity, precision and ROC AUC are lower for all predictions using the LOOCV evaluation. Only the values of the specificity showed an increase for the BDTs and the RF. The other methods did not change for this parameter. The method with the highest sensitivity using the LOOCV evaluation still is the RF approach. The highest specificity and the highest precision are achieved by BDTs. Also the highest ROC AUC value at the LOOCV evaluation no longer is reached by DTs and BDTs together. DTs obtained slightly higher values then BDTs.

Table 4.2: Statistical measures of the performance of the LOOCV prediction for powdery mildew

Method	Accuracy	Sensitivity	Specificity	Precision	ROC AUC
k-NN	0.69	0.30	0.82	0.37	0.64
DT	0.70	0.30	0.84	0.40	0.70
BDT	0.71	0.29	0.85	0.41	0.69
RF	0.64	0.58	0.65	0.37	0.65

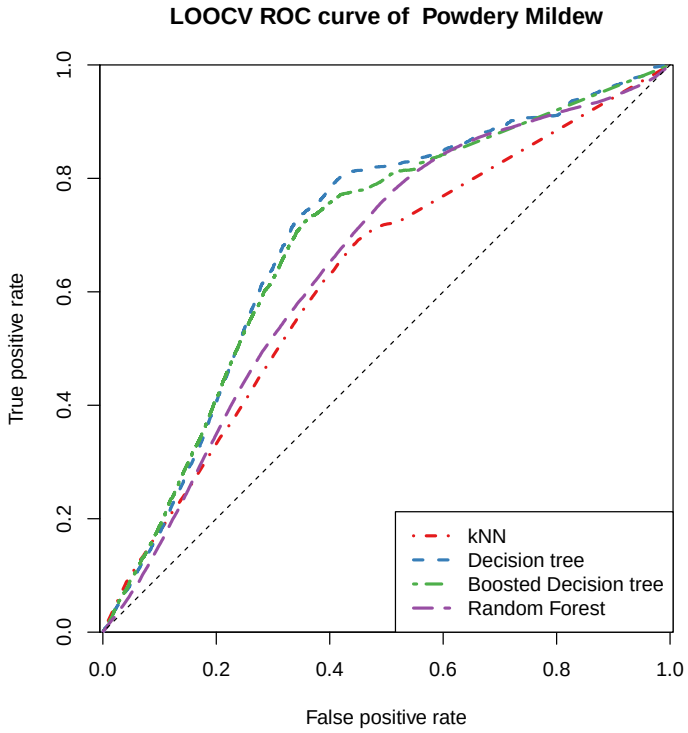


Figure 4.2: Receiver Operating Characteristic curve of the leave-one-out cross-validation for powdery mildew

The ROC curve (Figure 4.2) reflects the results of table 4.2. As for the holdout evaluation the k-NN approach shows the flattest curve. In contradiction to the ROC

curve of the holdout method (figure 4.1) the curve of the RF method is much closer to the k-NN curve as implied by the ROC AUC values of both methods. The highest true positive rate at a comparable false positive rate is reached by the DT method slightly followed by the BDT method.

Tables 4.3 and 4.4 show a specific output of the LOOCV evaluation (Figure 3.11). Since specific years and locations were studied it was possible to calculate the ROC AUC for the specific locations and years.

Table 4.3: ROC AUC measures of the LOOCV prediction for powdery mildew separated according to the locations and number of compared cases

Location	k-NN	DT	BDT	RF	Cases
Barlt	0.50	0.22	0.46	0.26	25090
Birkenmoor	0.49	0.60	0.60	0.58	20113
Elskop	0.52	0.33	0.46	0.34	22373
Fehmarn	0.46	0.27	0.35	0.36	1485
Futterkamp	0.47	0.58	0.58	0.54	21078
Kastorf	0.50	0.54	0.56	0.50	13880
Kluvensiek	0.54	0.57	0.59	0.57	22200
Loit	0.52	0.53	0.55	0.48	21826
Niendorf	0.49	0.58	0.57	0.48	10662
Nienrade	0.47	0.48	0.47	0.49	11774
Schönberg	0.48	0.48	0.50	0.48	4766
Sönke-Nissen-Koog	0.49	0.22	0.47	0.34	22550

Table 4.3 does not show large differences between the prediction methods. All predictions receive ROC AUC values around 0.5 at almost all locations. Such values imply random guessing and therefore no good prediction (Fawcett, 2006). Some locations show higher deviation from 0.5. The locations Barlt, Elskop and Sönke-Nissen-Koog receive very low ROC AUC values for the methods DT and RF. Fehmarn also stands out in table 4.3. Although the ROC AUC of the k-NN prediction is close to 0.5 the predictions of the other methods resulted in much lower values. As the comparison with figure 2.12 (page 42) shows, a ROC AUC below 0.5 implies a low true positive rate at rising false positive rates. The models tend to predict the opposite result at the aforementioned locations. Interestingly the location Fehmarn also has the lowest number of predictions taken into account for the LOOCV evaluation, probably due to the fact of limited monitoring years. Figure A.3 also shows that the other named locations have very little exceedances over all years. Also, location Birkenmoor stands out since all methods but k-NN received quite high ROC AUC values in comparison to the other values of the table.

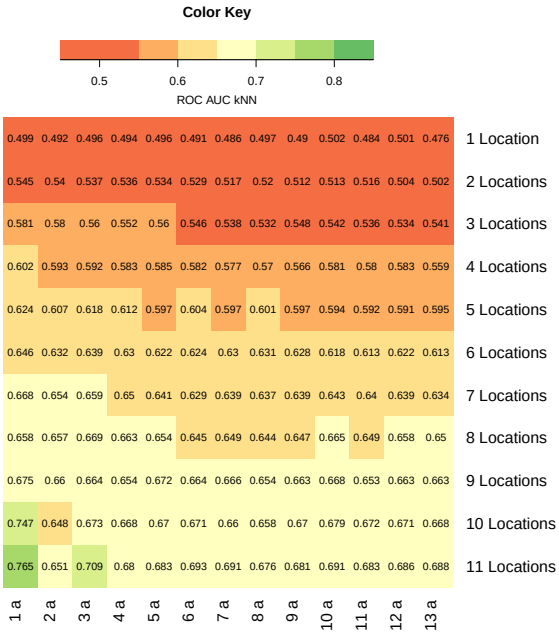
Table 4.4 presents ROC AUC values of the predictions, sorted according to the different years. The number of cases taken into consideration for the calculation differs between the years from 5,106 in 2016 to 17,457 in 1996. This discrepancy is caused by the deviating number of locations (Figure A.3) and the different count of wheat varieties (Table 3.2) taken into consideration. Figure A.3 also shows the reason there are no ROC AUC values available for the year 1999. Since no relevant infections were detected in this particular year on Ritmo, and the other wheat varieties, it was not possible to calculate the ROC AUC value. The ROC AUC values are higher if the predictions are split by year and not by location. The lowest values are achieved by k-NN and RF in 2005 and by DTs and BDTs in 2008. The highest values are reached by k-NN and RF in 2009 and by DTs and BDTs in 2012. In most of the years the BDT algorithm resulted in the largest ROC AUC value, while the k-NN algorithm produced the largest ROC AUC in only two years. Additionally in the years after 2008 higher values in the prediction methods besides k-NN can be identified as more constant than in the years before.

Table 4.4: ROC AUC measures of the LOOCV prediction for powdery mildew separated according to the years and number of compared cases

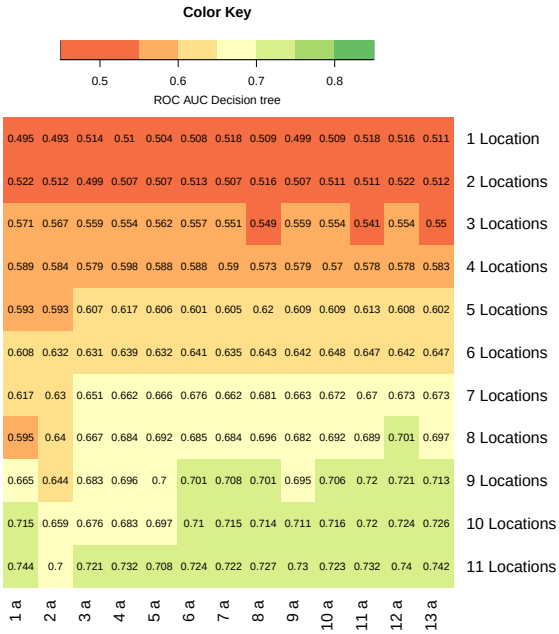
Year	k-NN	DT	BDT	RF	Cases
1996	0.64	0.74	0.69	0.67	17457
1997	0.70	0.76	0.75	0.79	14348
1998	0.73	0.74	0.74	0.65	16055
1999	*No Data*	*No Data*	*No Data*	*No Data*	*No Data*
2000	0.69	0.67	0.67	0.70	14129
2001	0.71	0.68	0.69	0.71	9440
2002	0.65	0.72	0.73	0.72	8211
2003	0.69	0.79	0.76	0.78	7260
2005	0.54	0.65	0.66	0.61	8184
2006	0.66	0.75	0.77	0.71	5757
2007	0.70	0.71	0.69	0.69	7590
2008	0.68	0.63	0.62	0.66	7384
2009	0.84	0.86	0.84	0.89	7770
2010	0.78	0.83	0.80	0.83	7776
2011	0.66	0.77	0.77	0.79	7350
2012	0.78	0.90	0.90	0.78	9177
2013	0.64	0.77	0.78	0.76	7957
2014	0.73	0.84	0.86	0.75	8584
2015	0.66	0.87	0.86	0.76	12530
2016	0.72	0.84	0.85	0.81	5106

Figure 4.3 shows the results of the holdout evaluation for different number of years and locations used for the calibration of the model. For all methods used to predict the exceedance of the powdery mildews damage threshold, a noticeable dependency on the number of locations used to create the prediction models can be identified. Another trend can be seen in the ROC AUC with fewer years and many locations. The predictions using only one year but eleven locations received the highest ROC AUC values for k-NN and BDT predictions. Since the random algorithm did not only test the year 1996, no values are available for the combination of 12 locations and one year. For two and three years and all locations also high values were reached, but then the k-NN no longer achieved such high values (Figure 4.3(a)). The other methods realise higher ROC AUC again with a combination of many years and many locations. Thereby the RF algorithm constantly achieves higher values even with combinations of less locations in comparison to DTs and BDTs (Figures 4.3(b), (c) and (d)). While the RF prediction always reaches ROC AUC values above 0.65 if more than 5 locations and one year are used for the prediction, the DT algorithm would need more than 6 locations and three years and the BDTs would need more than 6 locations and two years. The k-NN prediction reaches these values only if more than 8 locations are used. However, the DTs, BDTs and RFs reach ROC AUC values above 0.7 if at least 10 locations and 4 locations are used.

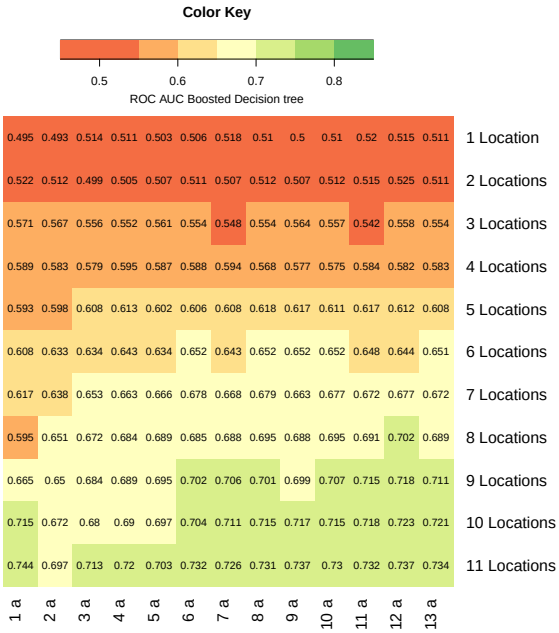
Figure 4.4 shows the logarithmic average "Mean Decrease Accuracy" of all RF prediction models created during the holdout evaluation. Following Nicodemus (2011), the "Mean Decrease Accuracy" is a robust indicator describing the influence of a parameter on the resulting RF model. A high "Mean Decrease Accuracy" represents a high influence of the parameter. Following figure 4.4 the primary influence on the decision of the RF model is the susceptibility class of the studied wheat species as presented in table 3.2. Following the multi-annual climatic variables real evapotranspiration and wind speed as well as the CTU, representing the wheat growth, and the elevation are of major importance for the decision of the RF models. Then the variables describing the weather situation during the infection period follow, beginning with the temperature and the DTU, derived from the temperature.



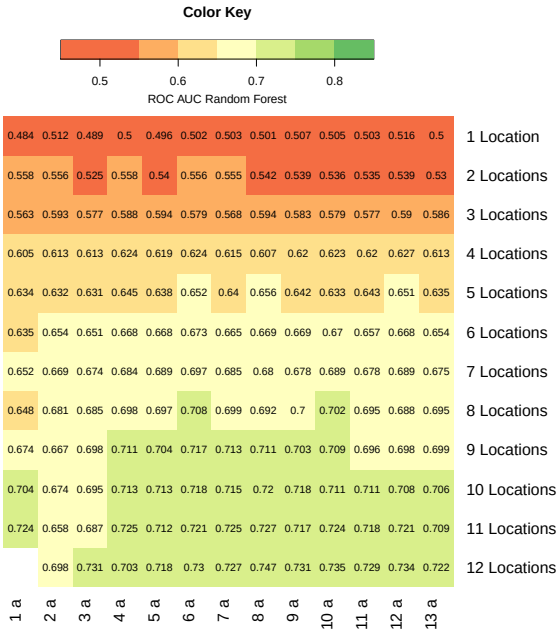
(a) *k*-Nearest Neighbor (k-NN)



(b) Decision Tree (DT)



(c) Boosted Decision Tree (BDT)



(d) Random Forest (RF)

Figure 4.3: Heatmaps of the Receiver Operating Characteristic Area under the curve for the machine learning procedure depending on number of years (a) and locations for powdery mildew

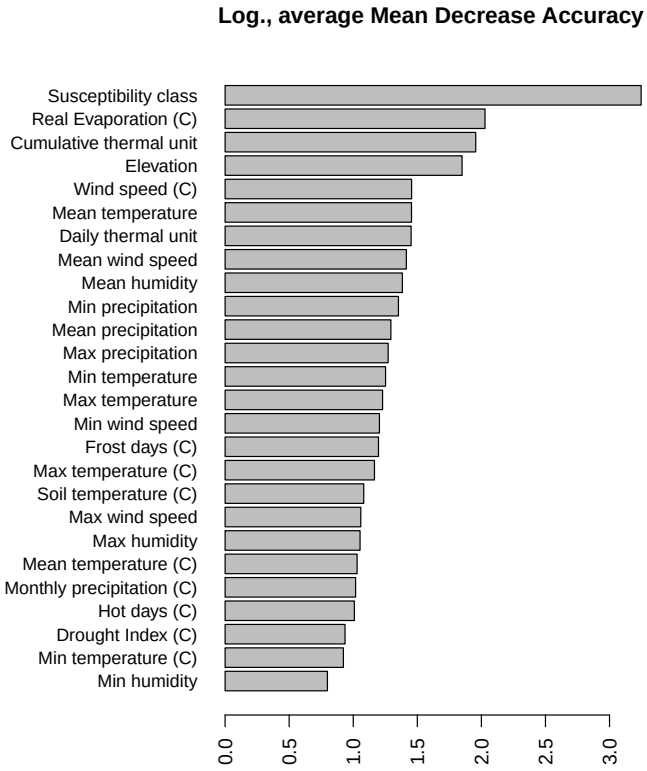


Figure 4.4: Logarithmic, average "Mean Decrease Accuracy" of the holdout Random Forest predictions for powdery mildew. Names according to table 3.4

Results of the real time modelling in 2017

Following, the results of the application of the different machine learning techniques to all known data are shown. As described in chapter 3.3.4, the methods were fitted to all known powdery mildew observations in the period from 1996 to 2016 and were used to predict the occurrence of disease threshold exceedances in 2017. Table 4.5 shows the statistical measures resulting from the comparison of the observed disease incidence for the three available wheat varieties Ritmo, Dekan and Inspiration (powdery mildew susceptibility: 5, 1 and 3 as shown in chapter 3.2.2) with the predicted exceedances. The DT and BDT procedures receive the highest accuracy, specificity, precision and ROC AUC values shortly followed by the k-NN method. Of those the DTs always have the highest values. The RF algorithm receive considerably lower values. The overall accuracy is more than 20 % below the accuracy of the DTs and the specificity is almost 40 % below the one of the DTs. More than 90 % of all observed underruns of the disease threshold were predicted correct using the DT. The sensitivity however shows how many of the observed exceedances of the disease threshold were classified correct. Here the RF almost reaches 90 % correct classifications while the DT classified more than 50 % of the observed exceedances as underruns.

Table 4.5: Statistical measures of the performance of the prediction of yield relevant powdery mildew events in 2017

Method	Accuracy	Sensitivity	Specificity	Precision	ROC AUC
k-NN	0.78	0.67	0.80	0.42	0.83
DT	0.84	0.44	0.92	0.55	0.86
BDT	0.83	0.67	0.87	0.51	0.83
RF	0.61	0.89	0.55	0.30	0.76

Figure 4.5 shows how the DT algorithm predicted whether an underrun or an exceedance would occur. The first split classifies all raster cells with real evapotranspiration (a climatic parameter) higher than 35.53 mm/d as *No exceedance* or value 0. This prediction is based on 1001 observations by which nearly all showed *No exceedance*. As can be seen in figure 4.6, separation according to this value divides the study area into an eastern and a western part. In the figures the unit mm/d*10 provided by the DWD is used (DWD, 2016a). The next split therefore only affects the eastern part of the study area. This split considers the susceptibility of the wheat variety. All wheat varieties with a susceptibility class which is not 2 or 5 would be predicted as *No exceedance* with a high probability. The next split classifies the raster cells with real evapotranspiration lower than 34.69 mm/d as *Exceedance* or value 1, although only

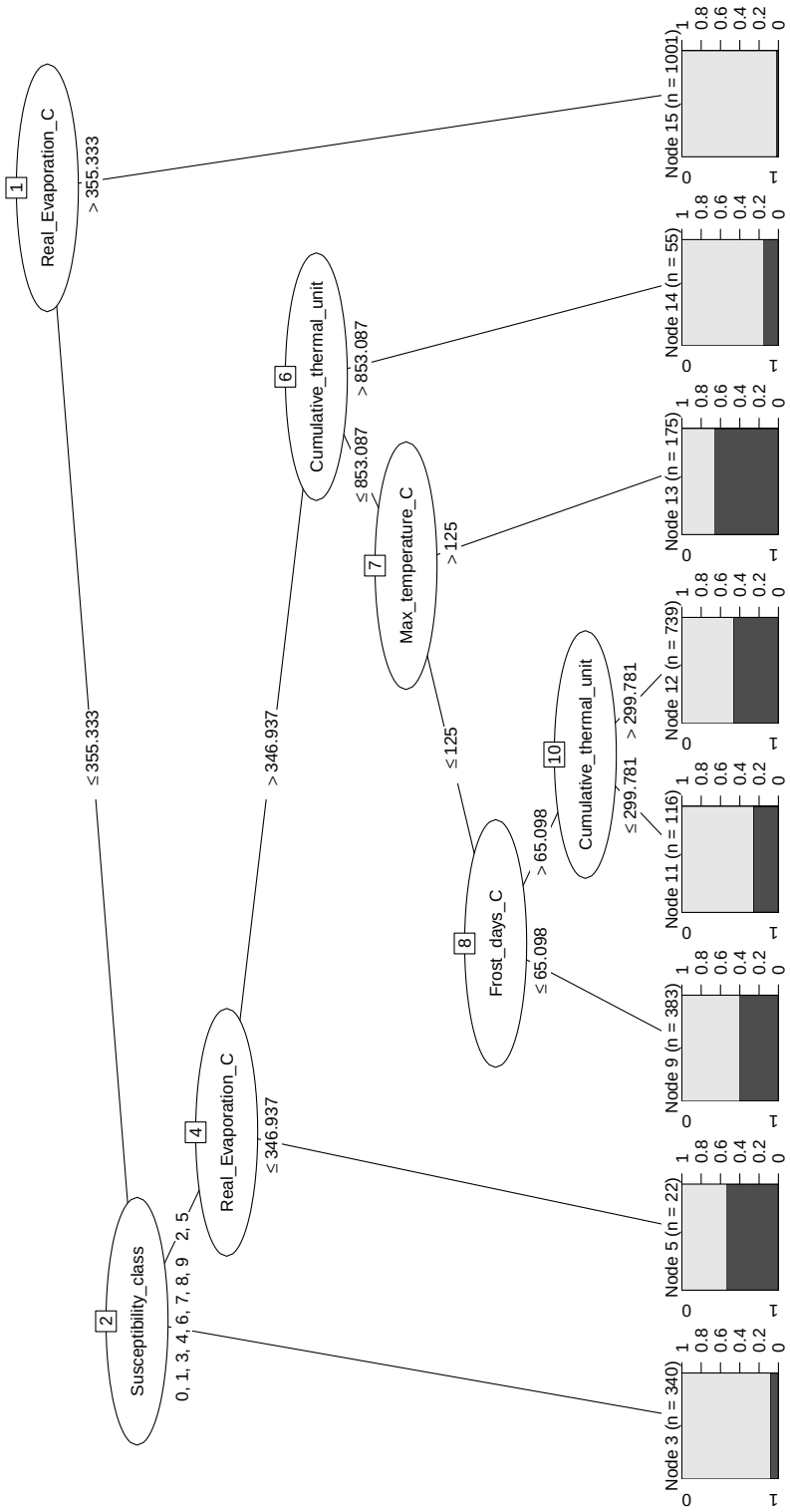


Figure 4.5: Decision Tree modelled to predict yield endangering powdery mildew events

more than half of the 22 observations with these properties showed exceedances. The distribution leads to a probability of exceedance of little more than 50 %. Figure 4.6 shows that this would affect the easternmost part of the study area.

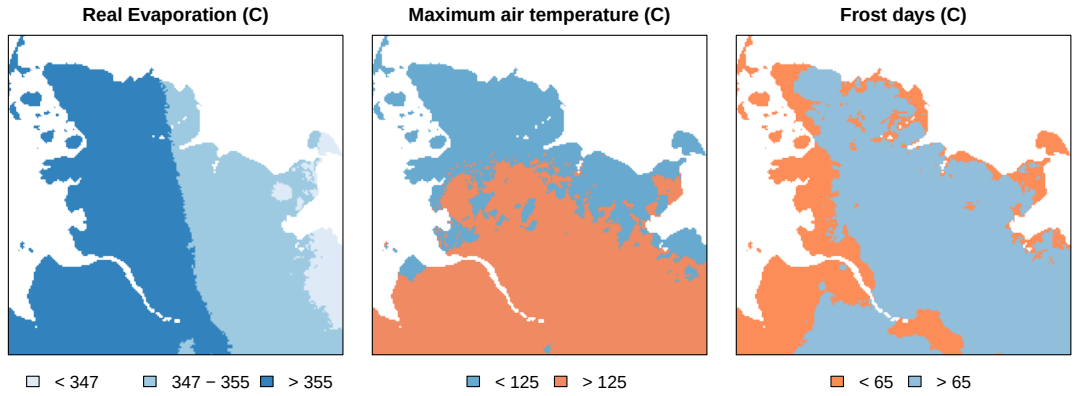


Figure 4.6: Multi-annual climatic real evapotranspiration (mm/d*10), maximum air temperature (°C*10) and number of frost days as classified by the Decision Tree model (figure 4.5)

The next split divides based on the CTU. If this value is above 853.087, which implies it is late in the year, or the year is rather cold (figure 4.7), *No exceedance* is expected. Otherwise, the next split occurs based on the climatic maximum air temperature. In more than 60 % of the observations used to create the model, which fulfilled the above conditions, an average temperature above 12.5 °C can be used to classify correctly as *Exceedance*. As the comparison with figure 4.6 shows, this affects the southern and central part of the study area. The next split separates based on the average count of frost days. Less than 65 frost days would imply the prediction *No exceedance*. As figure 4.6 shows, this affects the coastal areas, while the inland areas would be split again by the CTU. Both results of the split would result in the claim *No exceedance*. A CTU lower than 300 would result in an exceedance probability of only 20 % while a higher CTU would imply a probability of nearly 50 %.

Figure 4.8 shows the logarithmic average "Mean Decrease Accuracy" of the RF predictions for powdery mildew. The parameter influencing most of the results is the susceptibility class followed by the DTU and the climatic real evapotranspiration. The susceptibility class and the real evapotranspiration also have a major influence on the DTs classification, but the DT has not used the DTU. Whereas the CTU, which is the cumulative DTU, had a large influence on the classification of the DT. The next important parameters are all weather-related variables, representing the potential infection period. First the average temperature, precipitation and wind speed data are detected as relevant for the prediction. The climatic parameters, apart from the real

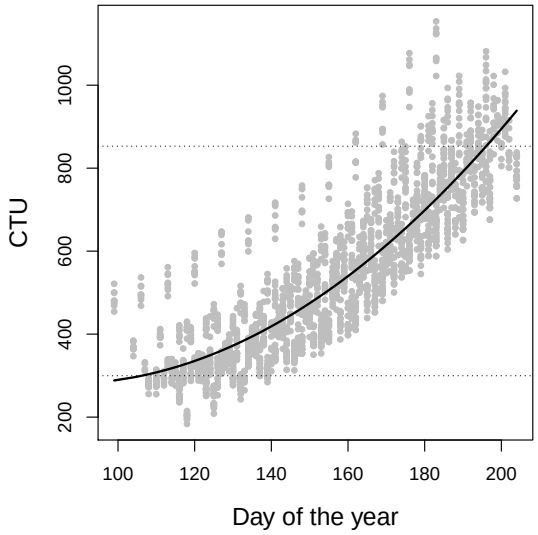


Figure 4.7: The Cumulative Thermal Unit (CTU) juxtaposed to the day of the year

evapotranspiration, are of minor importance for the RFs classification.

Figures 4.9 to 4.12 show the spatial prediction of the different machine learning procedures. In order to retain readability of the graph, only the prediction of the dates used for measurement of the infestation situations is shown. For the same reason, only predictions for the powdery mildew susceptibility class of Ritmo have been used for these depictions as mentioned in chapter 3.3.4. All methods besides the k-NN procedure show a clear separation of the study area into a western part and an eastern part. The k-NN method (figure 4.9) separates the westernmost part of the study area predicting correctly no exceedances in this area. The cut taken by the DT and BDT models (figure 4.10 and 4.11) goes through the centre as might have been expected from the model's split through the real evapotranspiration (figures 4.5 and 4.6). The real evapotranspiration also had a high "Mean Decrease Accuracy" in the RF model. The strong influence of this parameter can also be seen in figure 4.12. Unlike for the DT model the evapotranspiration is not such a sharp separator in the RF model as can be seen on May 29th, June 12th, June 19th and July 10th where exceedances were falsely predicted for the western part of the study area. Over the whole period, the RF and BDT prediction maps showed the highest probabilities of all methods. The predictions of the RF and the k-NN method are spatially more heterogeneous than the predictions of the DTs and BDTs. While the k-NN approach predicts higher probabilities in the central area with many areas of lower probability included the DT approach mainly separates the study area into two mostly homogeneous areas of exceedance and underrun probabilities.

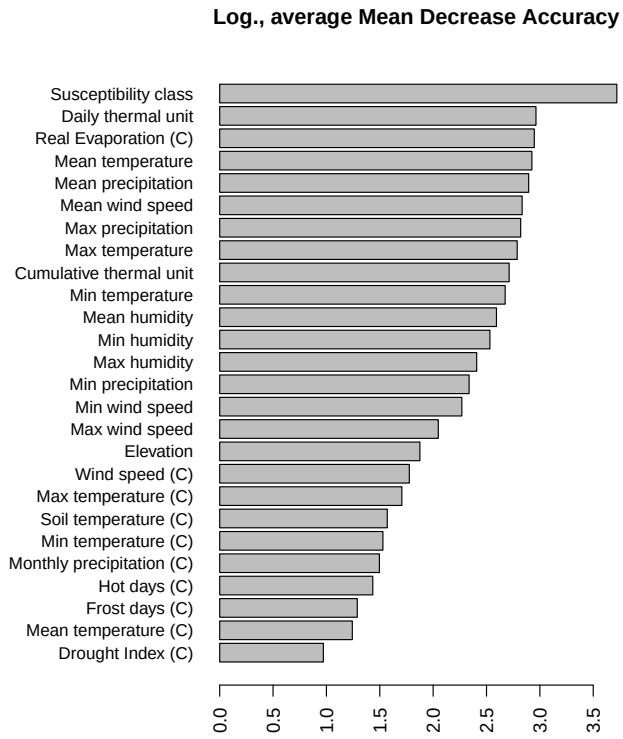


Figure 4.8: Logarithmic, average "Mean Decrease Accuracy" of the Random Forest predictions for powdery mildew applied on all data from 1996 to 2016. Names according to table 3.4

In addition to the spatial predictions depicted in figures 4.9 to 4.12, which also show the temporal dimension by the different time steps, figures 4.13 to 4.19 illustrate the predictions of the different machine learning techniques on a daily scale for the observation locations in the study area. These plots have two y-axes, the left one showing the probability of the exceedance of the disease threshold, and the right one showing the observed disease incidence. If the blue point lies above the black dotted line, indicating the disease threshold, the coloured lines of the machine learning methods should be above this line too, indicating a probability of more than 50 % that the threshold is exceeded. If this is not the case, a specific symbol shows, where which method made a wrong prediction.

Figure 4.13 depicts the predictions for Barlt, a location near the western coast of Schleswig-Holstein (compare figure 3.4). Over the whole observation period in 2017, no infestations with powdery mildew could be detected at this location. This was correctly predicted by all methods, except the RF technique. Beginning with the end of May the RF model predicted probabilities higher than 50 % in irregular intervals, which caused two observations to be predicted wrong.

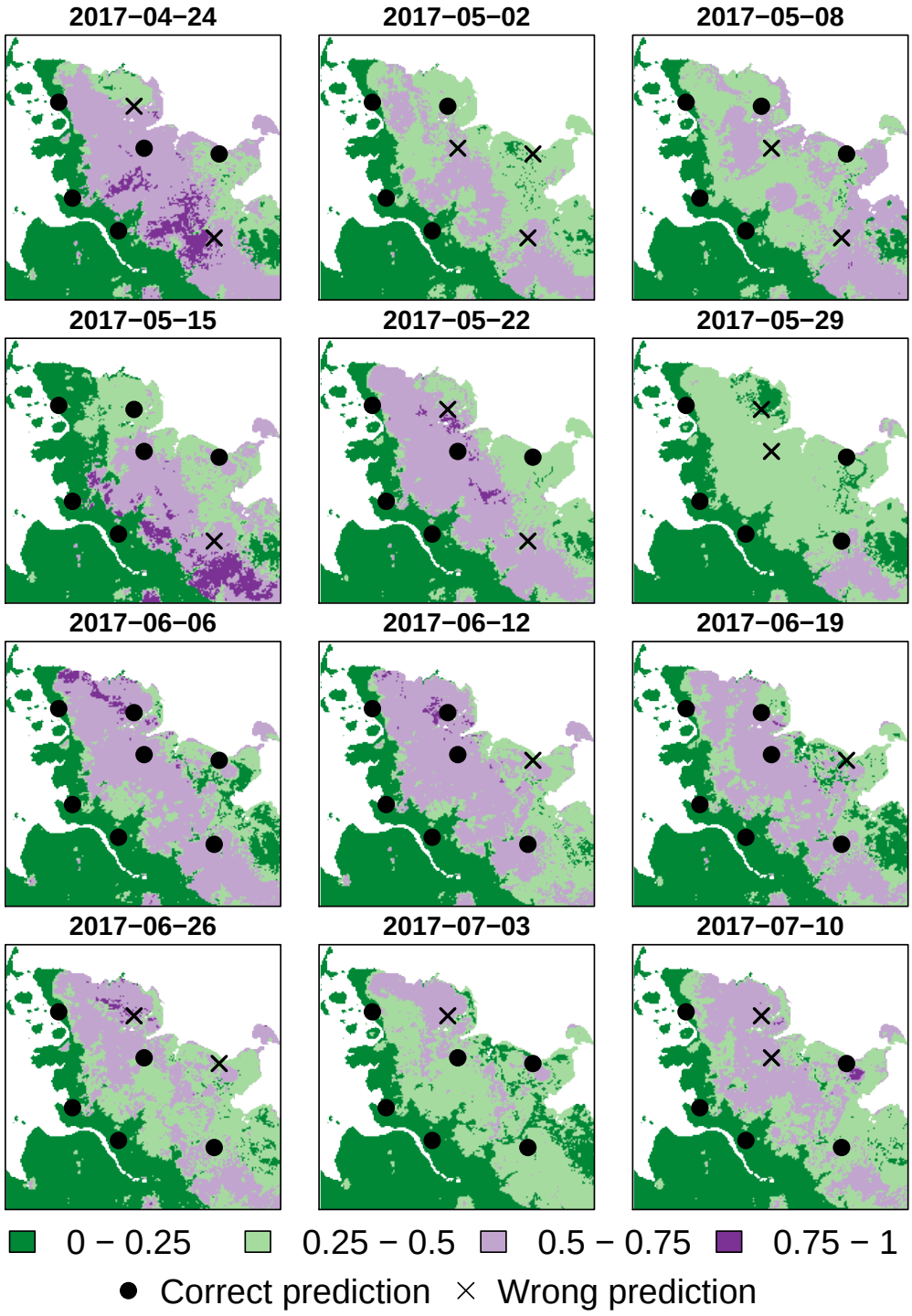


Figure 4.9: Spatial prediction of the probability of severe powdery mildew infections in 2017 using the *k*-Nearest Neighbor (k-NN) procedure

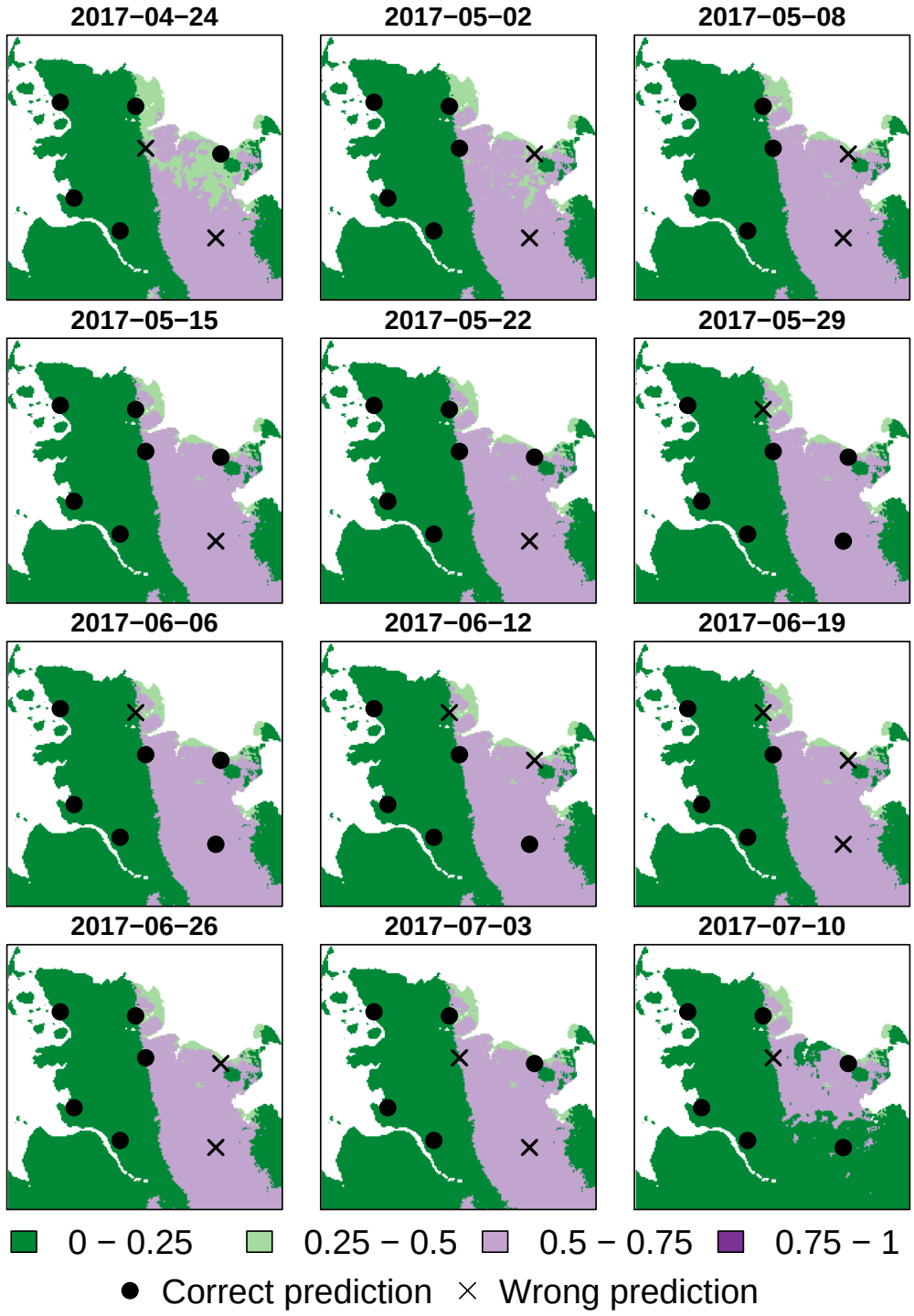


Figure 4.10: Spatial prediction of the probability of severe powdery mildew infections in 2017 using the Decision Tree (DT) procedure

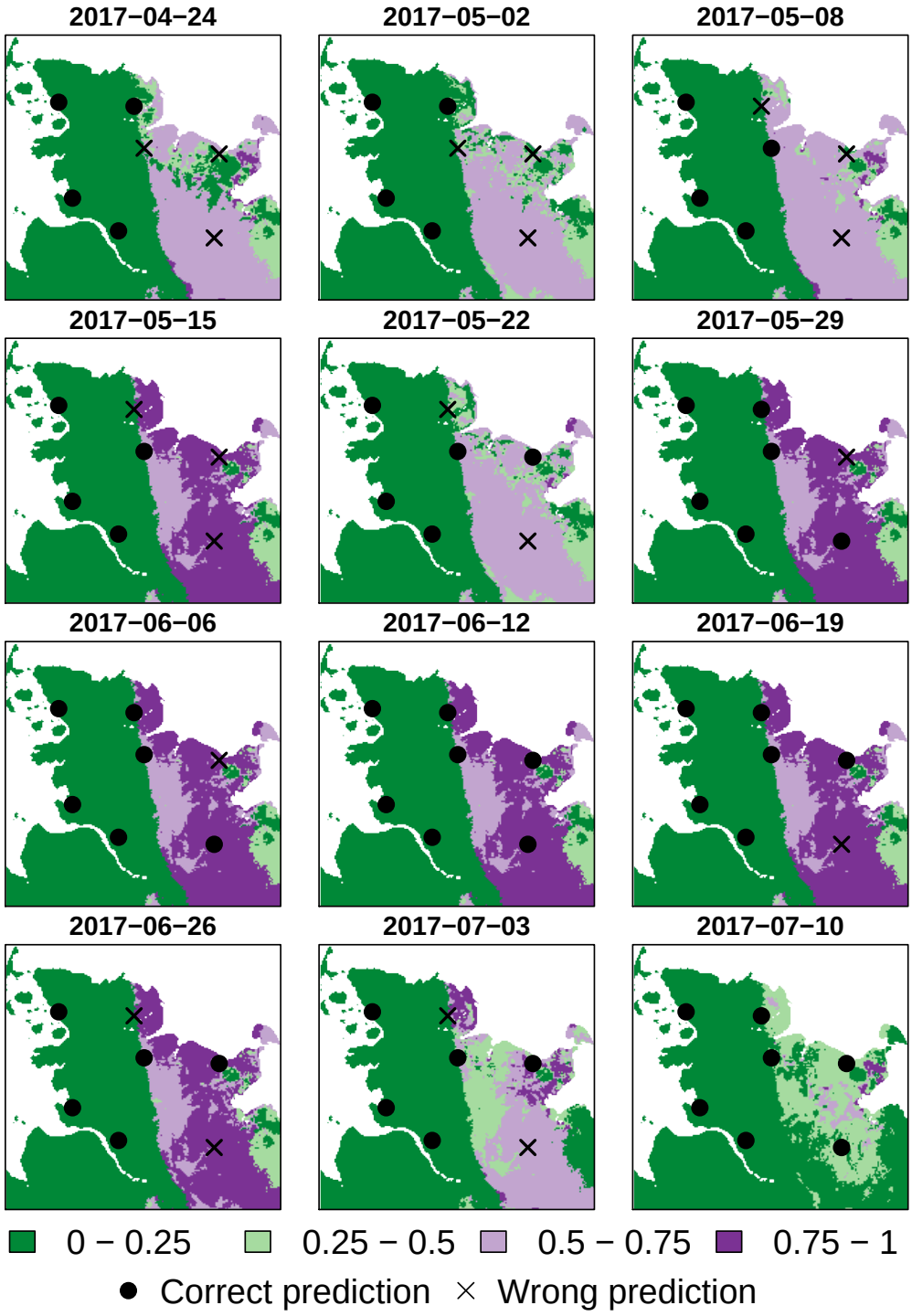


Figure 4.11: Spatial prediction of the probability of severe powdery mildew infections in 2017 using the Boosted Decision Tree (BDT) procedure

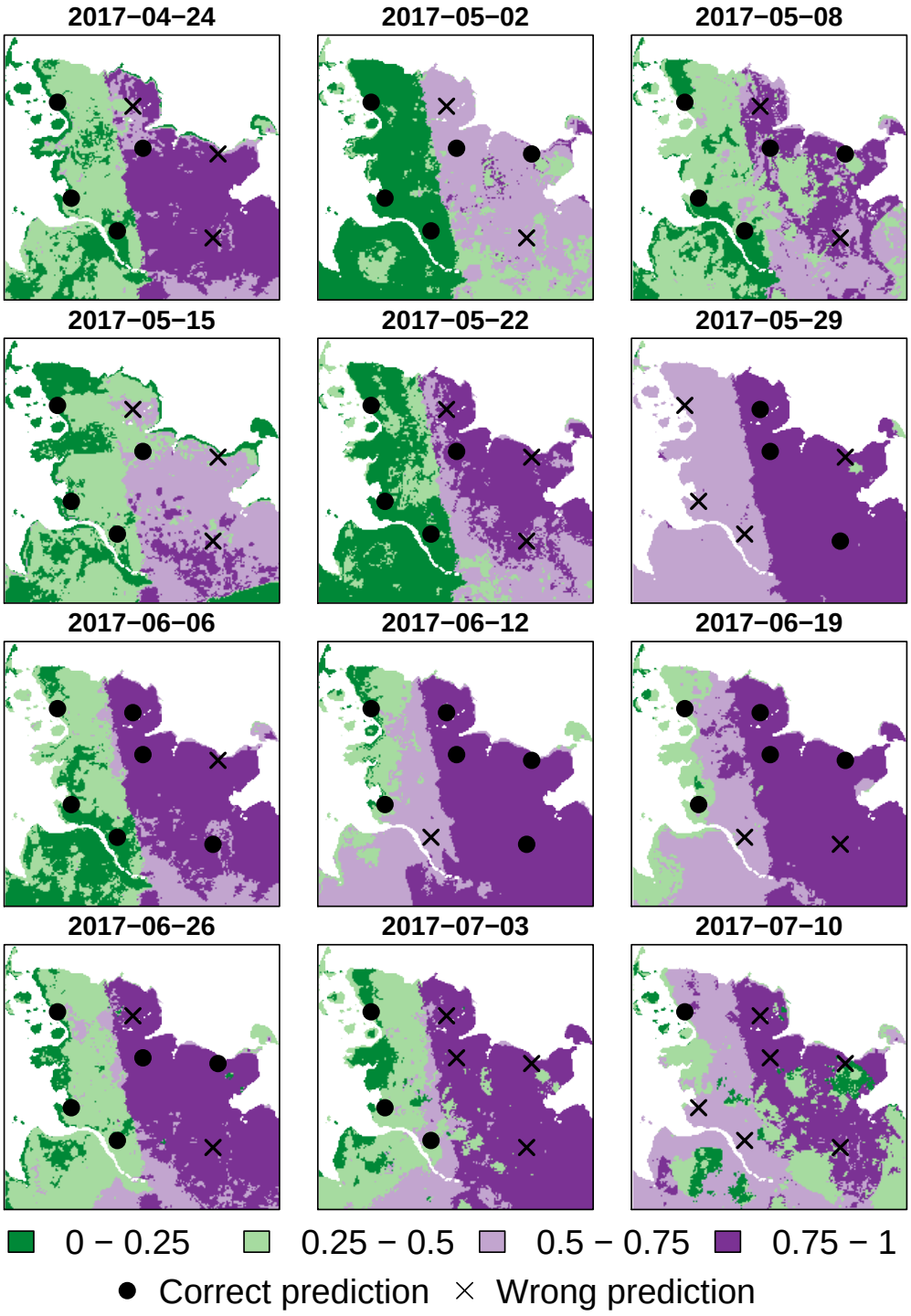


Figure 4.12: Spatial prediction of the probability of severe powdery mildew infections in 2017 using the Random Forest (RF) procedure

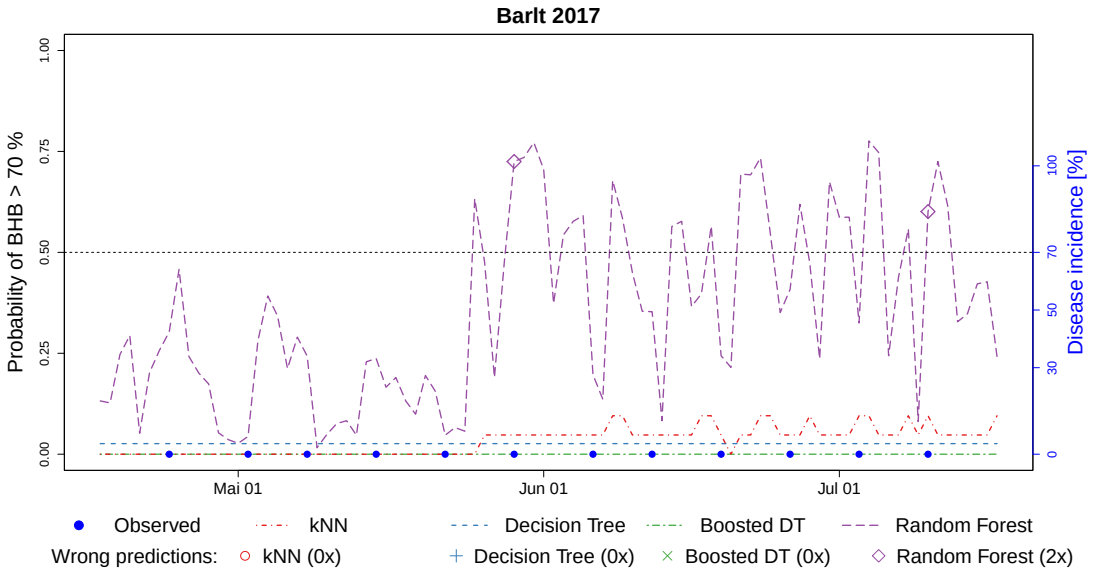


Figure 4.13: Temporal prediction of the probability of the exceedance of the powdery mildews disease threshold (70 %) in Barlt

Figure 4.14 shows the predictions for Elskop near the border to the federal state Lower Saxony (figure 3.4). Similar to the observations in Barlt only minor infestations could be observed at this location and no exceedance of the disease damage threshold could be detected. Again all methods but the RF method predicted this correctly. The RF model again predicted high probabilities from the end of May on which resulted in four wrong predictions.

In figure 4.15 the predictions for Futterkamp near the eastern coast (figure 3.4) can be seen. Here the disease incidence was high but not above the disease threshold at the first observation. The second and third observation, however, were above this threshold, followed by four underruns and three exceedances. The last two observations then showed a strong decline of the disease incidence. No method predicted this development completely right. Most of the correct predictions were made by the DT model, by assuming no exceedance of the threshold. The RF model got more than the half of all predictions wrong by predicting exceedances all the time. The k-NN and BDT prediction both made six mistakes, one more than the DTs. Thereby the k-NN approach only predicted one of the five exceedances as such, while the BDTs predicted three out of five exceedances correctly.

Figure 4.16 depicts the predictions for Kastorf in the southern part of Schleswig-Holstein (figure 3.4). The disease incidence is low at the first observations and starts to rise from the mid-May onwards. After it exceeded the disease threshold at the beginning of June, it declines and is close to 0 % from the end of June on. All machine

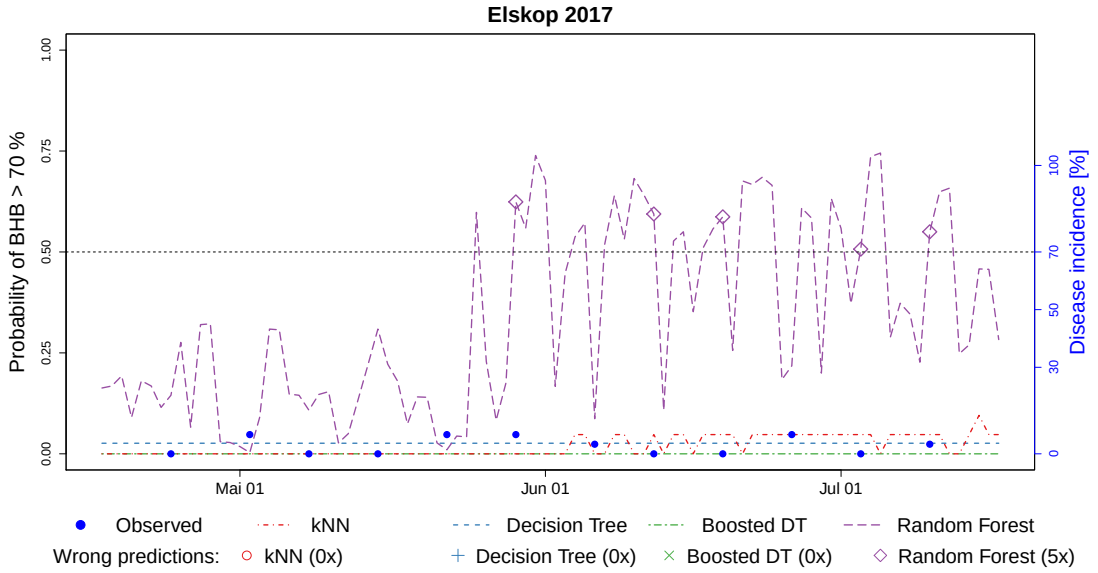


Figure 4.14: Temporal prediction of the probability of the exceedance of the powdery mildews disease threshold (70 %) in Elskop

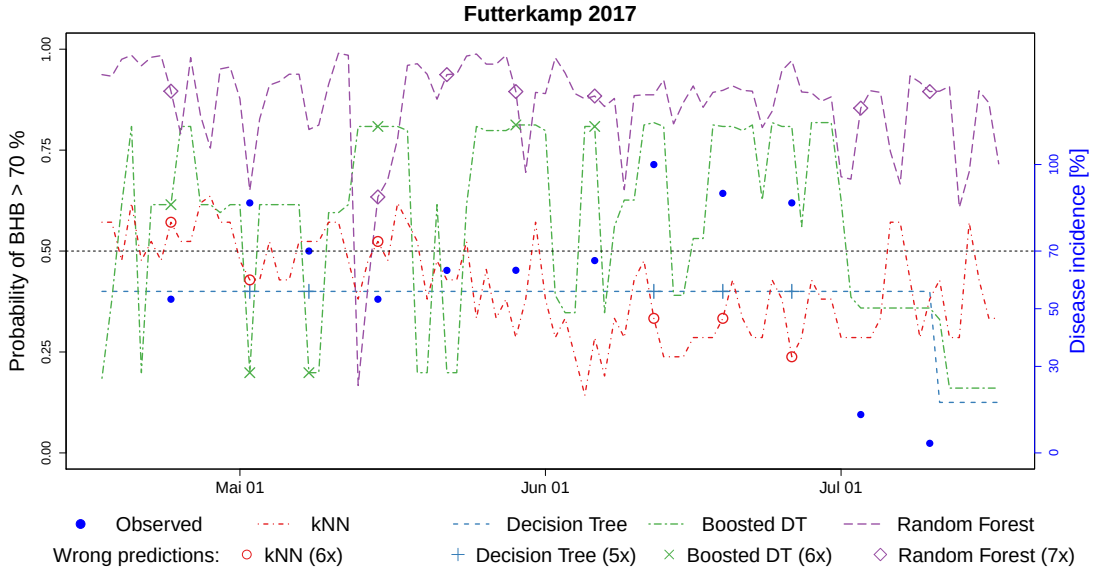


Figure 4.15: Temporal prediction of the probability of the exceedance of the powdery mildews disease threshold (70 %) in Futterkamp

learning methods predicted exceedances of the damage threshold over the whole season with a decline of all methods at the end, except for the RFs. The forecasts resulted in eight wrong predictions for each method but the RF model, which achieved nine wrong predictions.

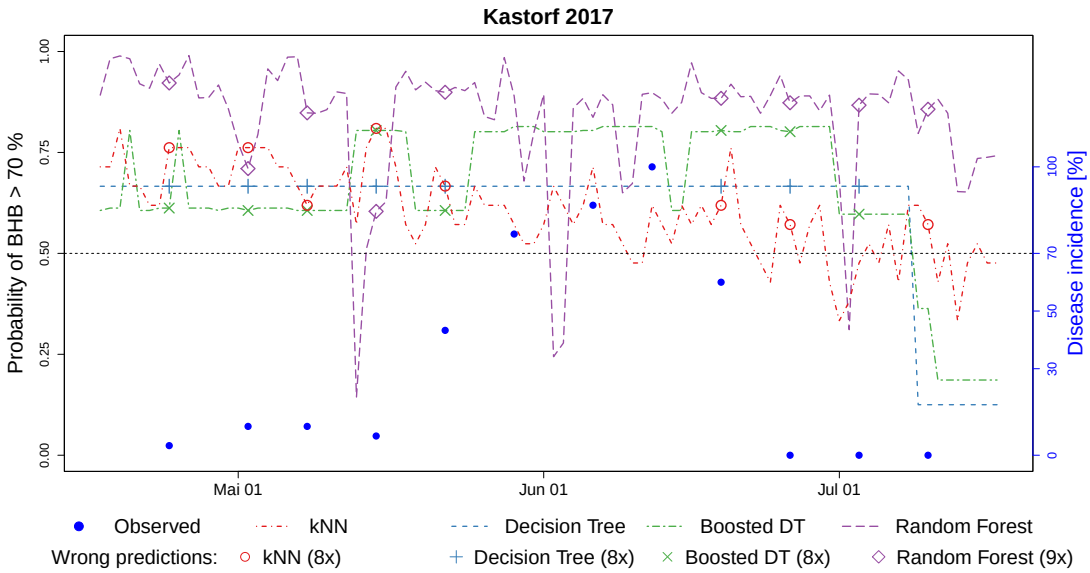


Figure 4.16: Temporal prediction of the probability of the exceedance of the powdery mildews disease threshold (70 %) in Kastorf

Figure 4.17 shows the predicted disease incidence for Kluvensiek in the north-eastern part of Schleswig-Holstein (figure 3.4). High disease incidences could be observed at this location over the whole season with a downfall at the end. Besides the decline in the end, the RF model predicted all observations correct at this location. The BDTs did not predict the high disease incidences at the beginning of the season, but predicted the decline correctly which resulted in the same amount of correct predictions. The k-NN prediction varied around the 50 % probability over the season and got five predictions wrong.

In figure 4.18 the predictions for Loit north of Kluvensiek (figure 3.4) are depicted. The development of the infestation was similar to Kastorf, with an exceedance at the beginning of June and a decline at the end of June. The predictions of the DTs are similar to the ones of Futterkamp. No exceedances were predicted over the whole season which resulted in four undetected exceedances. On the contrary, the RF model predicted exceedances the whole time which resulted in eight incorrect predictions for this location. The predictions of the k-NN method showed a slow rise of the probability over the season, missing the first exceedance but predicting the following exceedances correct. However, the method did not predict the decline at the end of the season.

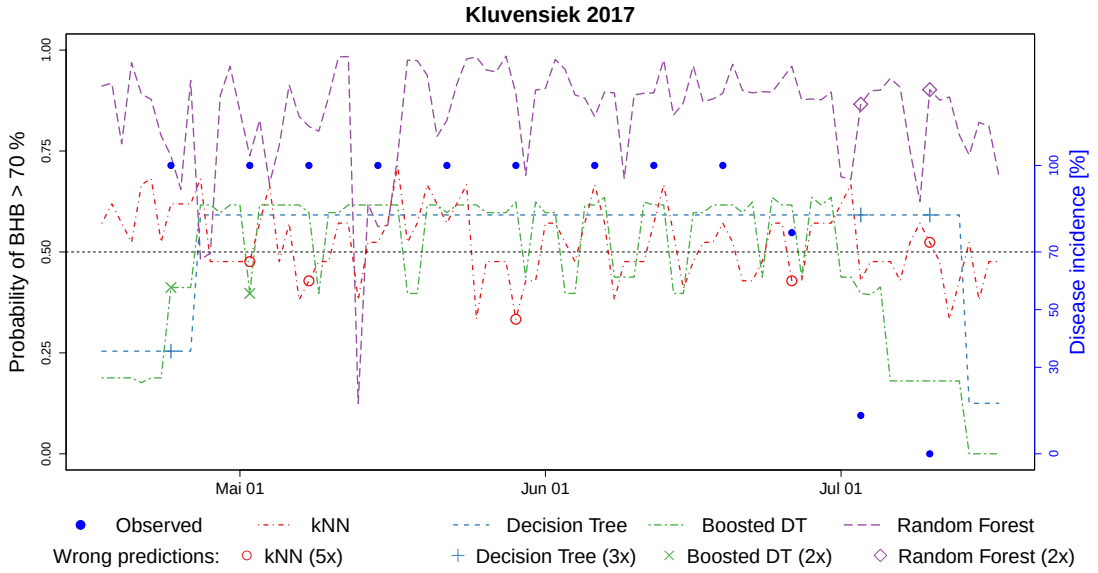


Figure 4.17: Temporal prediction of the probability of the exceedance of the powdery mildews disease threshold (70 %) in Kluvensiek

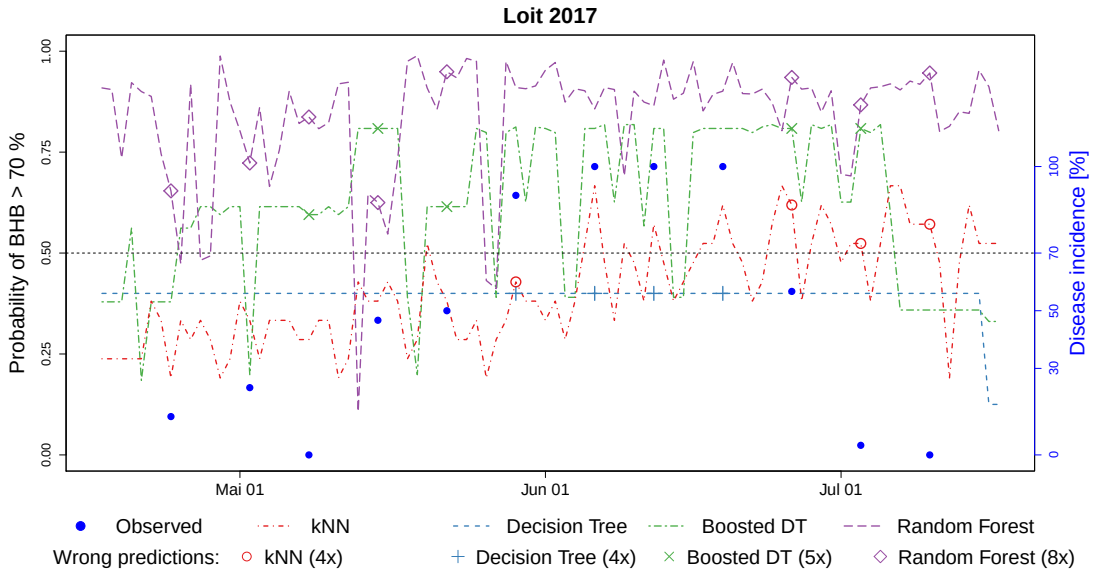


Figure 4.18: Temporal prediction of the probability of the exceedance of the powdery mildews disease threshold (70 %) in Loit

The BDT procedure predicted low probabilities in the beginning, an early rise and a late decline, which resulted in five incorrect predictions but no missed exceedance of the disease damage threshold.

Figure 4.19 depicts the predictions of the probability of a severe infestation event for Sönke-Nissen-Koog, the northernmost location near the western coast (figure 3.4). The disease incidences and the predictions for this location are similar to Barlt and Elskop. Also, in this case, no exceedance was observed and again the RF model made the only predictions of exceedances. From the beginning of June on high probabilities were predicted in irregular intervals which resulted in one wrong prediction for this model.

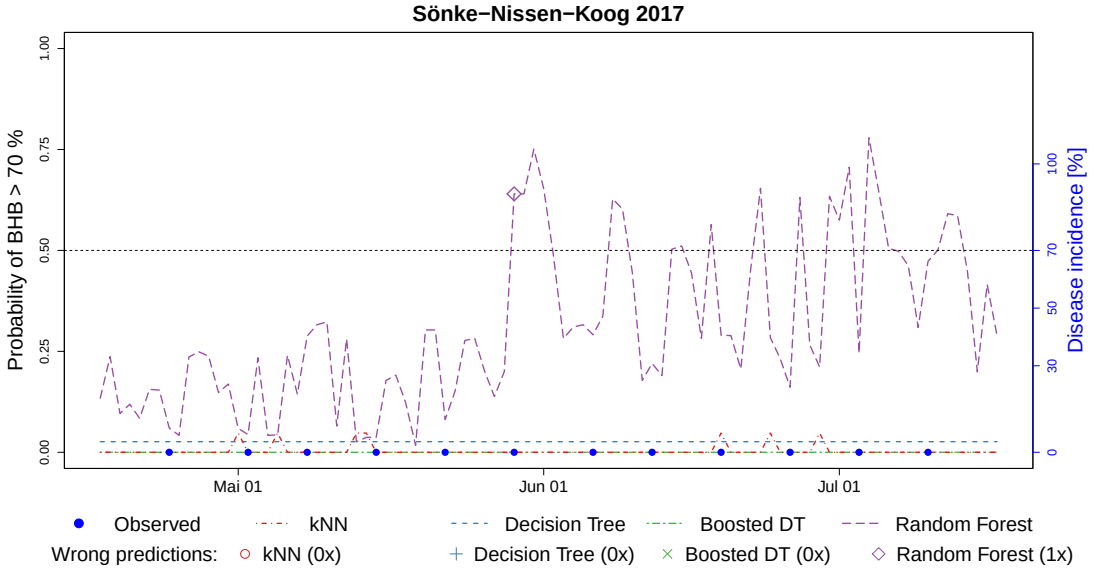


Figure 4.19: Temporal prediction of the probability of the exceedance of the powdery mildews disease threshold (70 %) in Sönke-Nissen-Koog

Summary of the prediction of powdery mildew events

Performance of powdery mildew models

- The evaluation of the outcome of the different machine learning methods resulted in the highest accuracy, specificity and precision using the Decision Tree and Boosted Decision Tree methods.

- By applying the *k*-Nearest Neighbor method, slightly worse results were achieved.

- Using the Random Forest machine learning technique, the overall approach

achieved a much lower accuracy and specificity, but also a much higher sensitivity than the other methods.

Spatial characteristics

- Considering the predictions in relation to the individual sampling sites resulted in a very low ROC AUC.
- The prediction of all models includes a sharp distinction between the eastern and western part of the study area (mainly due to the multi-annual parameter real evapotranspiration).
- The number of sites included in the forecast is very important for an improvement of the prognosis.

Temporal characteristics

- Considering the ROC AUC for the individual years, values around 0.7 with maxima for Boosted Decision Tree are obtained.
- An increased number of years taken into account for the prediction is less important for the quality of the forecasts.

4.2 Spatio-temporal prediction of brown rust events

In the following, the results of the prediction of yield relevant brown rust events are shown. First the results of the holdout method are summarised (Table 4.6 and figure 4.20). The comparison with the results of the powdery mildew prediction (table 4.1) shows higher values of the brown rusts accuracy, specificity and ROC AUC. The sensitivity and precision only differ slightly between the two pathogens.

Table 4.6: Statistical measures of the performance of the holdout prediction for brown rust

Method	Accuracy	Sensitivity	Specificity	Precision	ROC AUC
k-NN	0.82	0.38	0.90	0.42	0.69
DT	0.84	0.43	0.91	0.49	0.82
BDT	0.84	0.42	0.92	0.51	0.86
RF	0.78	0.75	0.79	0.40	0.84

Therewhile the order of the machine learning procedures also only differs a little. In the order of the accuracy values DTs and BDTs are leading with the highest values

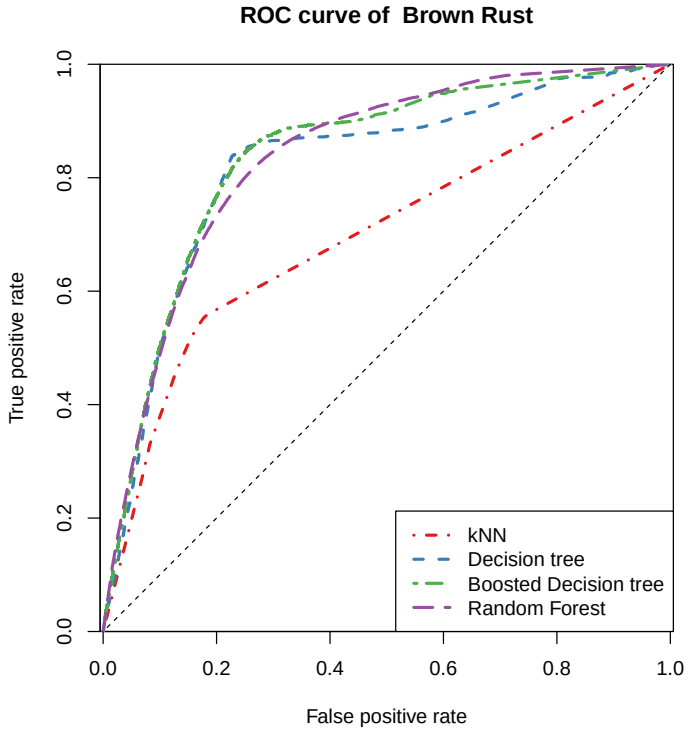


Figure 4.20: Receiver Operating Characteristic curve of the holdout validation for brown rust

for the brown rust prediction. The highest sensitivity again is reached by the RF prediction followed by the DTs, instead of the BDTs, as it has been for the powdery mildew prediction. Also the DTs and BDTs have changed rank regarding the specificity and the precision. For both variables BDTs reach the highest values using the holdout evaluation method during the prediction of brown rust. The highest ROC AUC value is received by the BDT method followed by the RF algorithm.

Also figure 4.20 gives an idea of the higher ROC AUC values received by the brown rust predictions in comparison with figure 4.1. The curves of the brown rust models are steeper than the ones of the powdery mildew forecast. That means, the true positive rate rises faster than the false positive rate. Nonetheless, the order of the curves seems to stay the same, besides the BDT curve, which overshadows the curves of DT and RF. Also at higher false positive rates the RF algorithm receives higher true positive rates than the DTs for the brown rust. The curve of k-NN also is considerably less pronounced in this case.

Secondly, the results of the LOOCV evaluation are described (Table 4.7 and figure 4.21). In comparison with the holdout method the statistical parameters only differ little. The k-NN values are reduced by 0.02 for each parameter except for the accuracy,

Table 4.7: Statistical measures of the performance of the LOOCV prediction for brown rust

Method	Accuracy	Sensitivity	Specificity	Precision	ROC AUC
k-NN	0.81	0.36	0.90	0.40	0.67
DT	0.83	0.40	0.91	0.47	0.81
BDT	0.83	0.37	0.92	0.47	0.84
RF	0.79	0.69	0.81	0.40	0.82

which is reduced by 0.01, and the specificity, where the value is unchanged. The DT values are 0.03 lower for sensitivity, 0.02 for precision and 0.01 for ROC AUC and accuracy respectively. The BDT values are reduced in sensitivity by 0.05, in precision by 0.04 to the same value of DTs, in ROC AUC by 0.02 and in accuracy by 0.01. The LOOCV evaluation shows different changes for the RF method. While sensitivity is reduced by 0.06 and ROC AUC by 0.02 similar to the BDT method, the accuracy rises by 0.01 and the specificity by 0.02. Also the ROC curve of the LOOCV prediction (figure 4.21) shows no shift of the best performing method. The relative location of the curves stays the same including the DT curve with high false positive rates.

Table 4.8: ROC AUC measures of the LOOCV prediction for brown rust separated according to the locations

Location	k-NN	DT	BDT	RF	Cases
Barlt	0.73	0.89	0.90	0.88	24338
Birkenmoor	0.74	0.84	0.89	0.88	20887
Elskop	0.70	0.83	0.87	0.82	21466
Fehmarn	*No Data*	*No Data*	*No Data*	*No Data*	*No Data*
Futterkamp	0.59	0.78	0.81	0.81	21393
Kastorf	0.66	0.81	0.82	0.81	14009
Kluvensiek	0.71	0.81	0.86	0.84	21221
Loit	0.70	0.81	0.84	0.84	21686
Niendorf	0.77	0.83	0.85	0.80	11228
Nienrade	0.65	0.68	0.78	0.78	11909
Schönberg	0.73	0.76	0.82	0.77	4975
Sönke-Nissen-Koog	0.56	0.84	0.87	0.81	22203

The output of the LOOCV evaluation for different years and locations is depicted in tables 4.8 and 4.9. In contradiction to the results of the powdery mildew prediction (Table 4.3) higher ROC AUC values are received for all methods and all locations (Table 4.8). The lowest ROC AUC values are achieved by k-NN for the location Sönke-Nissen-Koog, by the DT and BDT method for Nienrade and by RF for Schönberg. The highest values are reached for Barlt for all machine learning procedures except for the k-NN

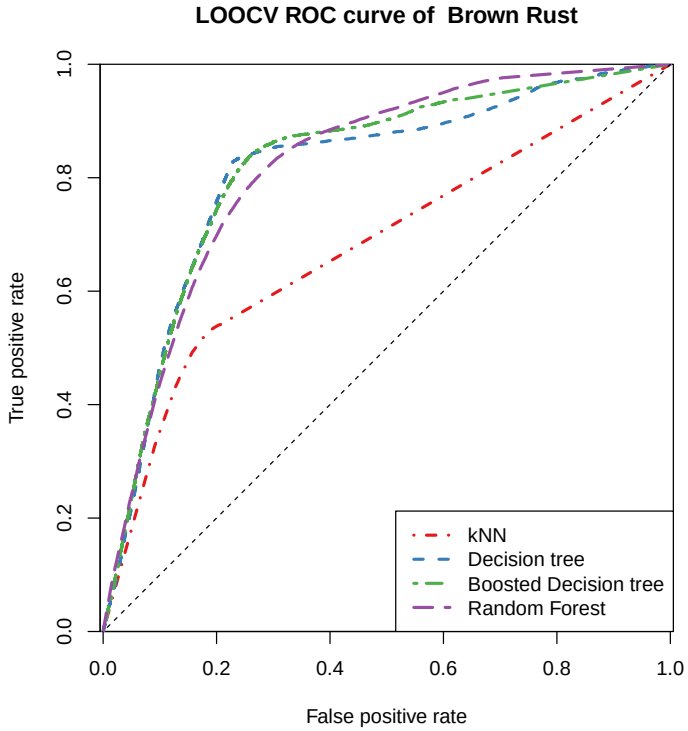


Figure 4.21: Receiver Operating Characteristic curve of the leave-one-out cross-validation for brown rust

approach where Niendorf is the best prediction. For most of the locations the BDT algorithm reaches the highest ROC AUC followed by the RF algorithm. The other methods never gain the highest value. Nevertheless the DTs also reach high values, which is also expressed in the average ROC AUC values. While the BDTs and RFs show the highest average values (0.846 / 0.822), DTs (0.807) closely follow in contrast to the k-NN approach (0.687). The Fehmarn site was thus omitted from these considerations. As the comparison with figure A.4 shows, there were no threshold exceedances observed during the monitoring procedure forbidding the calculation of ROC AUC.

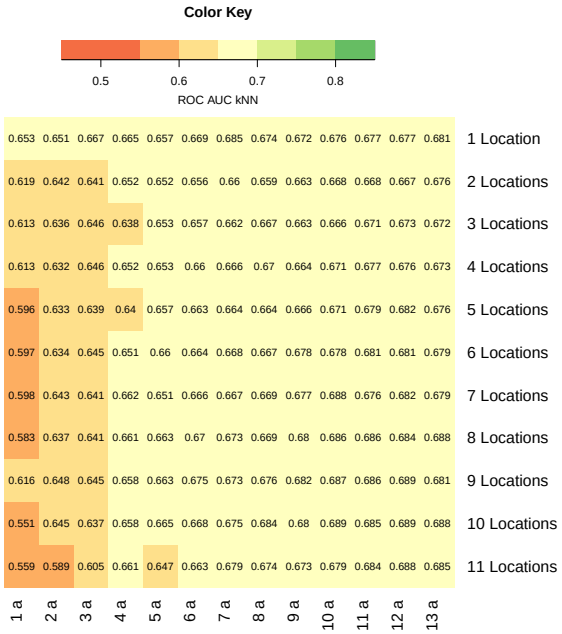
Table 4.9 shows the ROC AUC values of the different machine learning procedures calculated during the LOOCV prediction, separated by year. Like the year 1999 in the results of the powdery mildew prediction (Table 4.4) the year 2011 cannot be evaluated using the ROC AUC measure since the comparison with figure A.4 shows there were no exceedances of the brown rust damage threshold of 30 % disease incidence observed in this year. This procedure was possible for the year 1996 although figure A.4 shows no exceedances for this year either since the wheat variety Kanzler showed exceedances of the damage threshold in this year. All methods reach the lowest ROC AUC values besides the k-NN approach in the prediction for the year 2015. The lowest value of

the k-NN method is calculated for the year 2006. Altogether the results of k-NN again stand out since the methods ROC AUC values are substantially lower in comparison with the other methods. While the BDTs average ROC AUC over all years (0.869) is close to the RF (0.849) and DT values (0.847), k-NN is far behind (0.699).

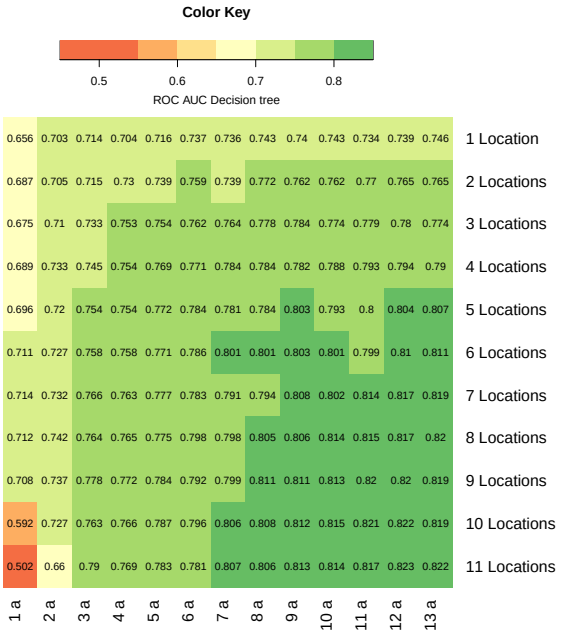
Table 4.9: ROC Area Under the Curve (ROC AUC) measures of the LOOCV prediction for brown rust separated according to the years

Year	k-NN	DT	BDT	RF	Cases
1996	0.61	0.80	0.80	0.78	19044
1997	0.82	0.77	0.87	0.78	14552
1998	0.71	0.85	0.85	0.86	15860
1999	0.65	0.82	0.89	0.84	17252
2000	0.61	0.85	0.88	0.78	15549
2001	0.79	0.93	0.95	0.94	9360
2002	0.65	0.84	0.82	0.83	8211
2003	0.71	0.91	0.91	0.91	7920
2005	0.86	0.96	0.97	0.95	8118
2006	0.59	0.88	0.86	0.84	6270
2007	0.64	0.78	0.84	0.83	7590
2008	0.64	0.78	0.78	0.78	7313
2009	0.62	0.81	0.81	0.83	7560
2010	0.77	0.91	0.95	0.92	8424
2011	*No Data*	*No Data*	*No Data*	*No Data*	*No Data*
2012	0.72	0.92	0.92	0.94	9177
2013	0.75	0.93	0.95	0.95	8833
2014	0.69	0.85	0.86	0.87	8510
2015	0.71	0.64	0.73	0.67	2800
2016	0.74	0.86	0.86	0.85	7452

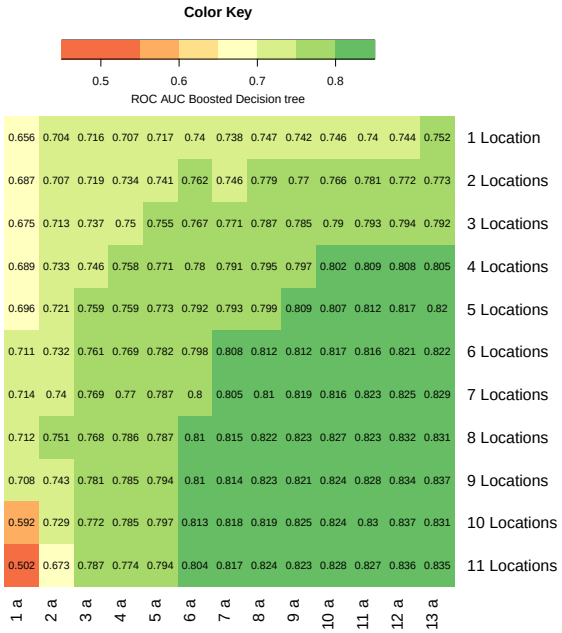
Figure 4.22 shows the results of the brown rusts holdout evaluation for different counts of years and locations. Contrary to the powdery mildew predictions (Figure 4.3) the rise of the ROC AUC values depends on both the number of locations and the number of years. The strong dependency on the number of locations for the powdery mildew prediction is mixed with the influence of the number of the years for the brown rust prediction. The k-NN prediction (Figure 4.22(a)) almost suggests a stronger influence of the number of years, without reaching ROC AUC values above 0.7. But with the other methods a consistent influence of both parameters is recognisable. Thereby the BDTs and RF predictions (Figures 4.22(c) and (d)) receiver higher ROC AUC with less years and locations in comparison to the DT predictions (Figure 4.22(b)). While DTs reach values of more than 0.8 if at least 8 years and locations are considered, BDTs require 5 years and locations and RFs at least 4 years and 5 locations.



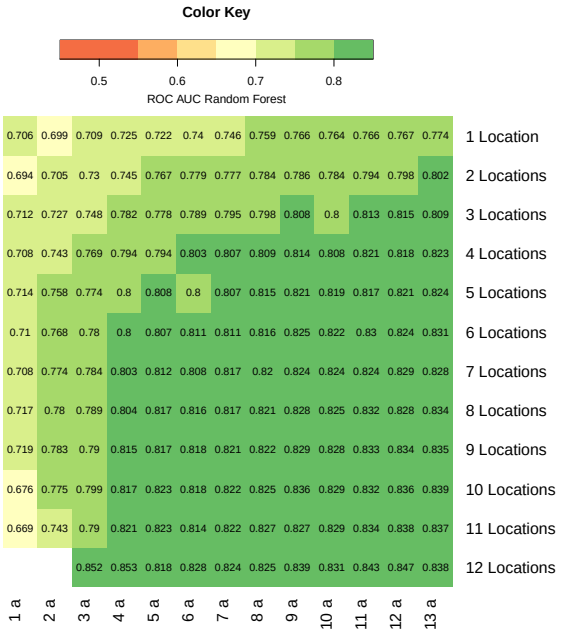
(a) *k*-Nearest Neighbor (k-NN)



(b) Decision Tree (DT)



(c) Boosted Decision Tree (BDT)



(d) Random Forest (RF)

Figure 4.22: Heatmaps of the Receiver Operating Characteristic Area under the curve for the machine learning procedure depending on number of years (a) and locations for brown rust

Figure 4.23 shows the logarithmic, average "Mean Decrease Accuracy" of all RF prediction models created during the holdout evaluation for the brown rust prediction. The primary influence on the RF models decisions is the CTU followed by the susceptibility class. Then all the weather variables follow, beginning with the mean humidity and the minimum temperature. Apart from the minimum precipitation on days defined as infection period the climatic parameters are less often considered for the RF decisions during the brown rust prediction procedure.

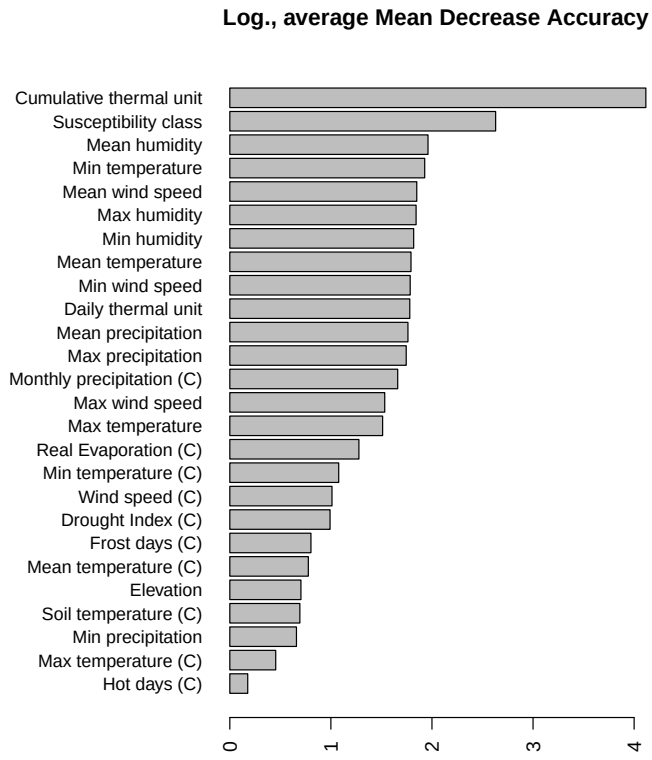


Figure 4.23: Logarithmic, average "Mean Decrease Accuracy" of the holdout Random Forest predictions for brown rust. Names according to table 3.4

Results of the real time modelling in 2017

In the following the results of the exemplary application of the different machine learning procedures as described in chapter 3.3.4 are shown. In preparation to predict the probability of an exceedance of the brown rusts disease threshold, the models were fitted to all known data of the years 1996 to 2016. Then they were applied to the new weather data of 2017 and the known climate data to create predictions for the three wheat varieties Ritmo, Dekan and Inspiration (brown rust susceptibility: 8, 8 and 5 as shown in chapter 3.2.2) which were observed at different locations in 2017. Table 4.10

depicts the statistical measures resulting from the comparison of the observed disease incidence with the predicted exceedances.

Table 4.10: Statistical measures of the performance of the prediction of yield relevant brown rust events in 2017

Method	Accuracy	Sensitivity	Specificity	Precision	ROC AUC
k-NN	0.74	0.19	0.97	0.73	0.80
DT	0.74	0.21	0.95	0.64	0.78
BDT	0.72	0.16	0.95	0.58	0.85
RF	0.82	0.58	0.92	0.74	0.91

The highest accuracy, sensitivity, precision and ROC AUC was reached by the RF technique. The specificity value of the RF is the lowest of the techniques, but it is quite large with more than 90 %. The highest specificity was achieved by the k-NN procedure which also received the second largest precision. Together with the DT and BDT method the k-NN technique got high values also in accuracy and ROC AUC, but very low values at sensitivity. Consequently, only around 20 % of all observed exceedances of the brown rust disease threshold were predicted as such by this methods. On the contrary the RF model predicted nearly 60 % of these events correctly.

Figure 4.24 depicts the DT model fitted on the brown rust observations and the weather and climate data associated with these infestations. The first split is done based on the CTU. A CTU lower or equal to 634.78 implies, according to the model, a very low probability of an exceedance of the brown rusts damage threshold. This prediction is based on 1903 observations. As the comparison with figure 4.25 shows this would refer to the average day of the year 172, which lies in the mid to the end of June. In a warm year, this day, of course, would be earlier and in a cold year later. The second split occurs on a CTU of 754.75, which would be the average day of year 186 (begin of July).

If the CTU is between 634.78 and 754.75, the next split is based on the climatic parameter minimum air temperature. An average minimum temperature higher than 5.8 °C would lead to the probability of exceedance of around 30 %. Figure 4.26 shows, that this would affect the most coastal areas, especially Fehmarn as the map 3.4 verifies. The next split separates based on the susceptibility class of the wheat variety. The classes 1, 2, 3, 5, 7 and 9 are assumed to undergo no exceedance of the threshold. The other classes are split according to the minimum air temperature of the assumed infection period. A temperature higher than 8.6 °C would lead to the prediction *Exceedance*, while a lower minimum temperature would predict *No exceedance*.

If at the beginning a CTU higher than 754.75 occurred the next split would have

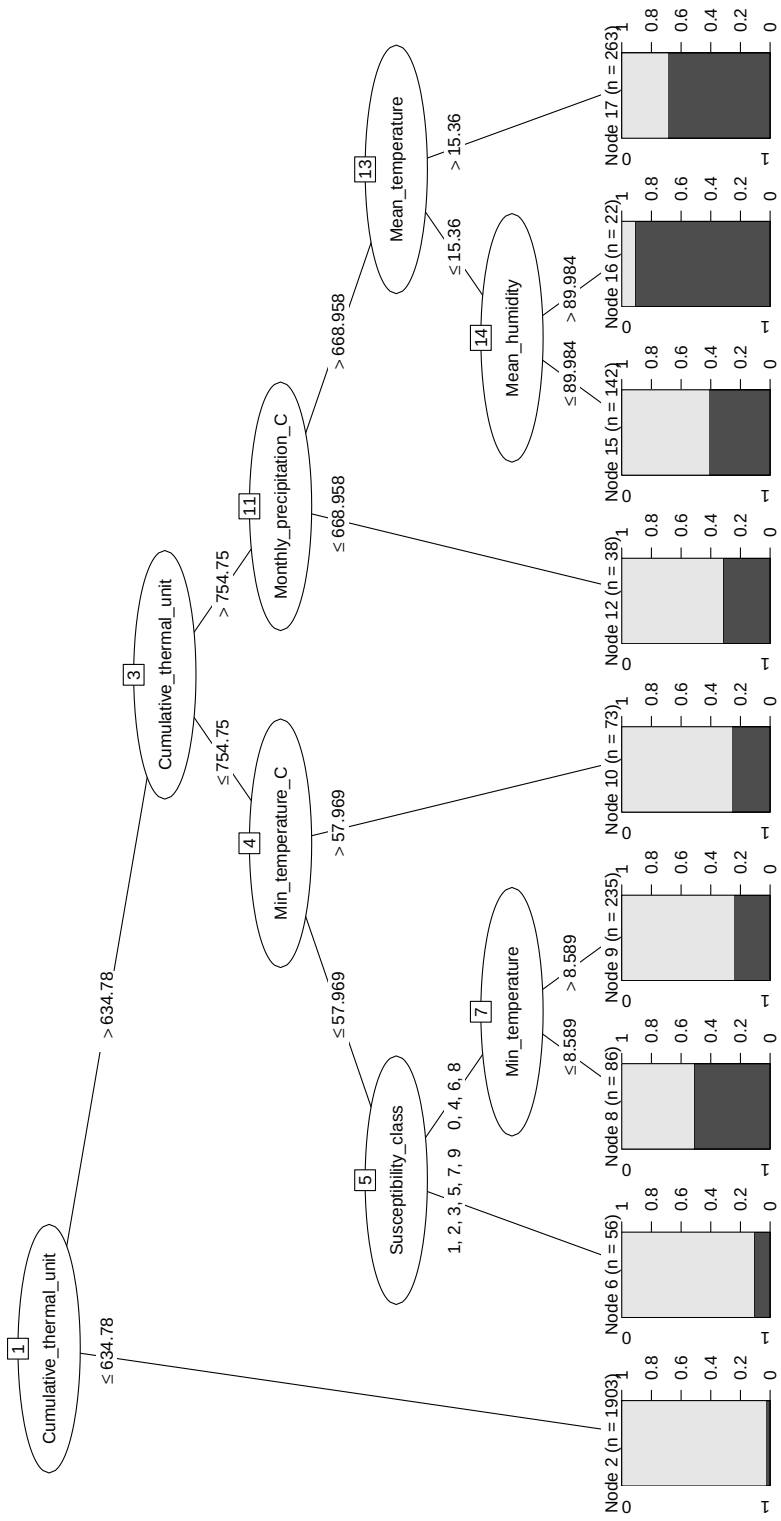


Figure 4.24: Decision Tree modelled to predict yield endangering brown rust events

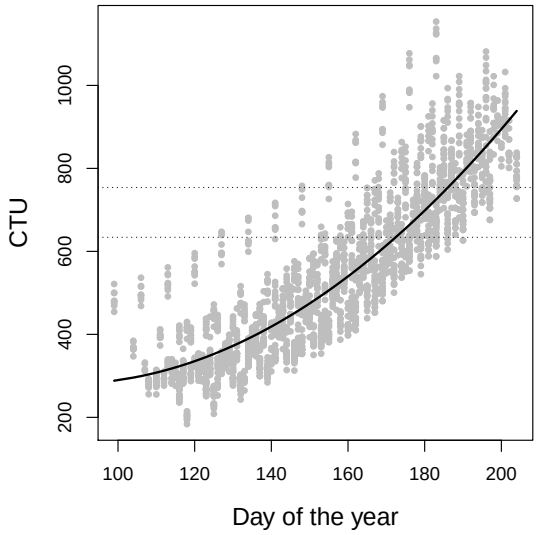


Figure 4.25: The Cumulative Thermal Unit (CTU) juxtaposed to the day of the year

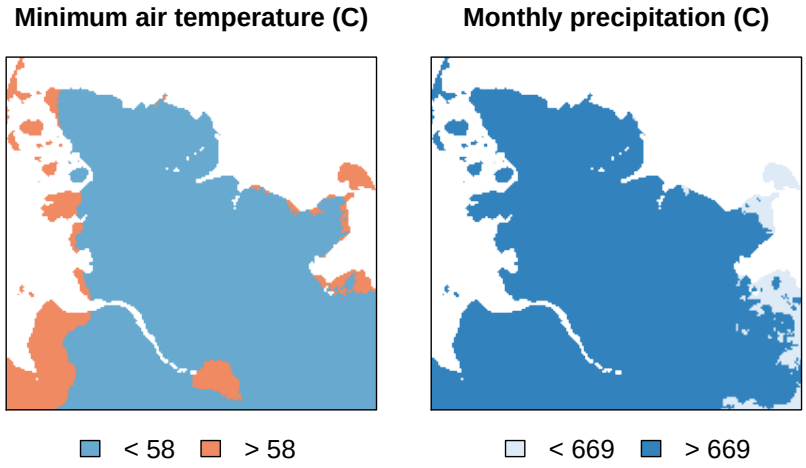


Figure 4.26: Multi-annual climatic minimum air temperature (°C*10) and monthly precipitation (mm) as classified by the Decision Tree model (figure 4.24)

been based on the climatic monthly precipitation. Average monthly precipitation lower than 669 mm would have led to the exceedance probability of around 30 %. As figure 4.26 shows, this split again affects the location Fehmarn. The larger part of the study area gets split based on the mean air temperature in the assumed infection period. If the temperature was above 15.4 °C, an *Exceedance* would be assumed. So if it is late in the season and not the easternmost part of the study area, a high temperature in the infection period is an indicator of brown rust spread. A lower temperature necessitates another split based on the relative humidity in the infection period. Humidity lower than 90 % would reduce the probability of the prediction *No exceedance*. A higher humidity would give a very high probability of *Exceedance*.

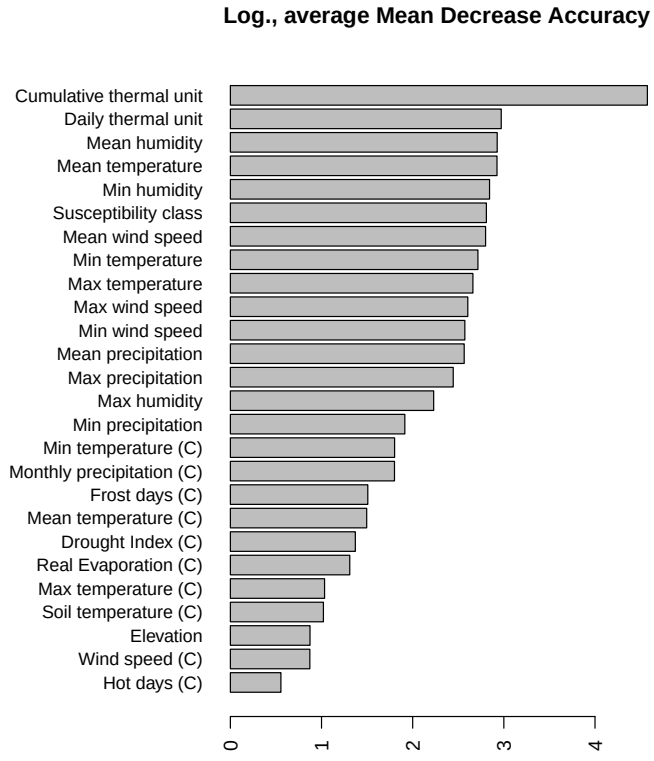


Figure 4.27: Logarithmic, average "Mean Decrease Accuracy" of the Random Forest predictions for brown rust applied on all data from 1996 to 2016. Names according to table 3.4

Figure 4.27 shows the logarithmic average "Mean Decrease Accuracy" of the RF model, fitted to the brown rust observations. The most important parameter is, similar to the DT model (figure 4.24), the CTU. The following parameters describe the temperature and humidity situation during the period of the infection followed by the susceptibility class. Then the wind speed and further temperature information fol-

low. The mean, min and max precipitation have a minor influence on the RF models predictions, although it is larger than the influence of the climatic variables.

Figures 4.28 to 4.31 depict the spatial prediction of the various machine learning models. All algorithms have in common that the maps for the first weeks show no predicted exceedances. The k-NN approach shows the first localised exceedance in the sixth week of observation. Going past the ninth week, a more distributed pattern of Exceedance predictions is displayed (figure 4.28).

The DT and BDT models (figures 4.29 and 4.30) show first predicted exceedances in the eighth week and the RF model in the sixth week (figure 4.31). The spatial predictions occur more heterogeneous within the models of the k-NN, BDT and RF methods. The DT predictions are more homogeneous, as the same models for the powdery mildew prediction. All models except the k-NN approach also have in common, that they produce *Exceedance* predictions for the whole study area during the last two to three weeks of the season. The k-NN approach showed higher probabilities during the last weeks only in the southern part of the study area and scattered near the eastern coast. Neither of the models predicted the first rise of the disease incidence in the seventh week. Also in the next week only the BDT model got one exceedance right. In the following week only the RF got more than half of the predictions correct. The high values in the tenth week were predicted correctly by the RF model but not by the other approaches. DT procedures followed up the predictions a week later but predicted incorrectly for the eastern locations, similar to RF models.

The temporal component of the prediction can also be seen in a higher temporal resolution of one day in figures 4.32 to 4.38. Here the predictions are shown for the seven locations used for the observation of the brown rusts disease incidence in 2017.

In Barlt (figure 4.32) the disease incidence began to rise in mid June, quickly exceeding the disease threshold of 30 %. The RF model falsely predicted one exceedance at the beginning of June but got the first observed exceedance wrong, like all the other methods. The second exceedance was predicted correctly by the RF, and incorrectly by the other procedures. By the time of the third exceedance, the DT and RF models were correct and the last exceedance was only mispredicted by the k-NN method.

Elskop (figure 4.33) experienced a disease incidence behaviour similar to Barlt. It began to rise in early June and exceeded the damage threshold in mid-June. The RF predicted an exceedance one week before the disease incidence had that high values. All following observations were predicted correctly by the RF model. The other procedures predicted the first two exceedances as underruns and the last two exceedances correctly.

In Futterkamp (figure 4.34) the disease incidence began to rise at the end of May

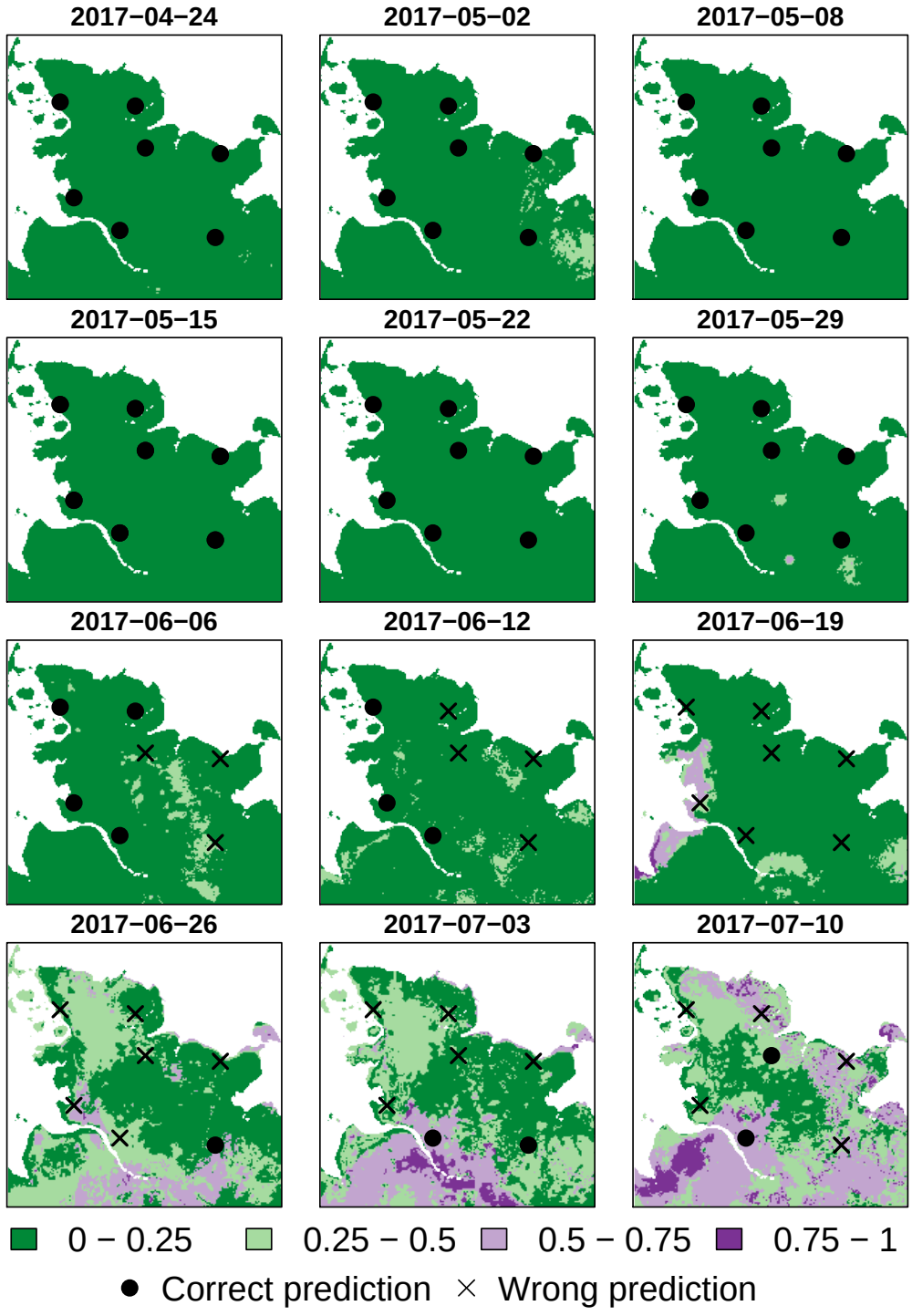


Figure 4.28: Spatial prediction of the probability of severe brown rust infections in 2017 using the *k*-Nearest Neighbor (k-NN) procedure

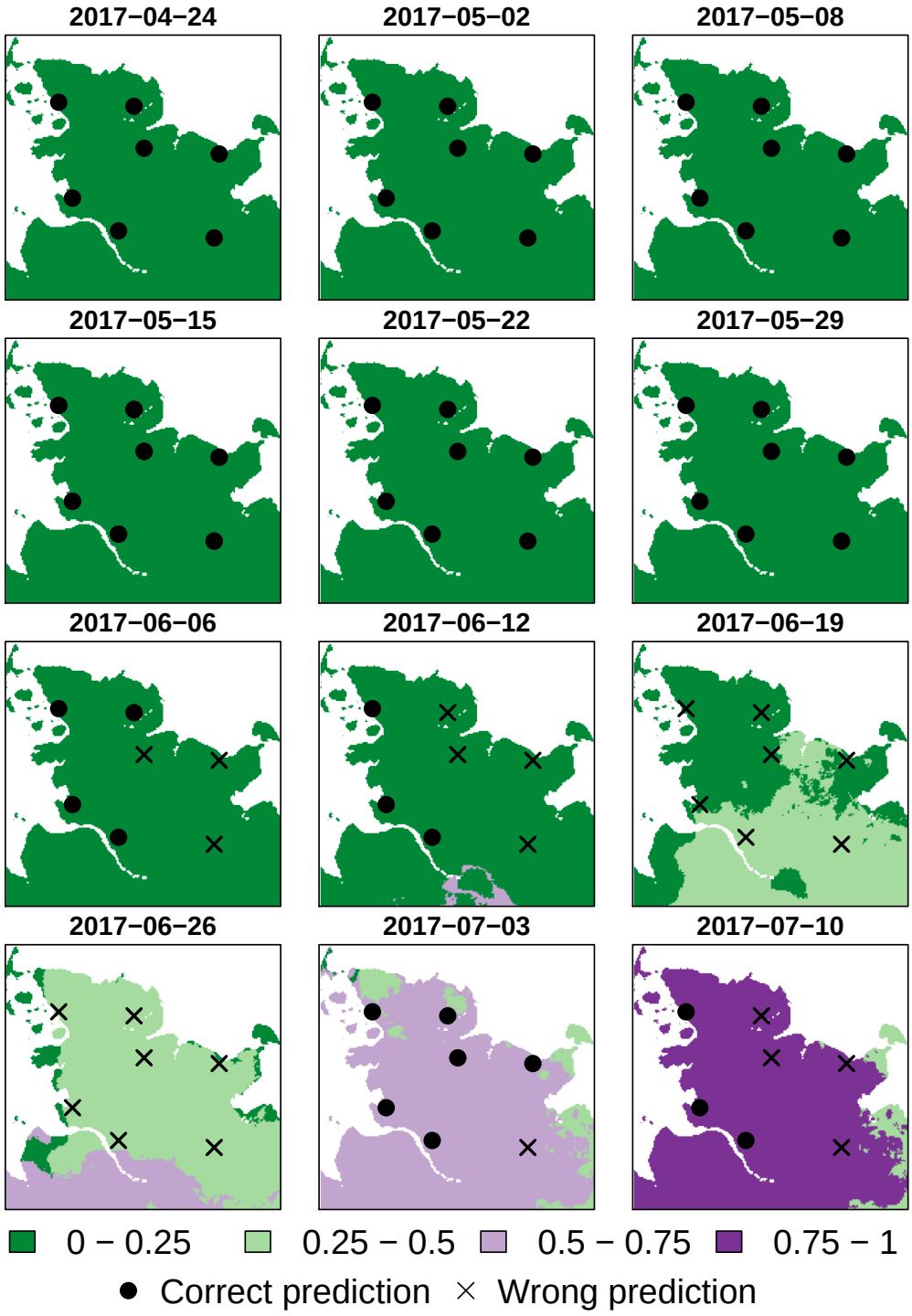


Figure 4.29: Spatial prediction of the probability of severe brown rust infections in 2017 using the Decision Tree (DT) procedure

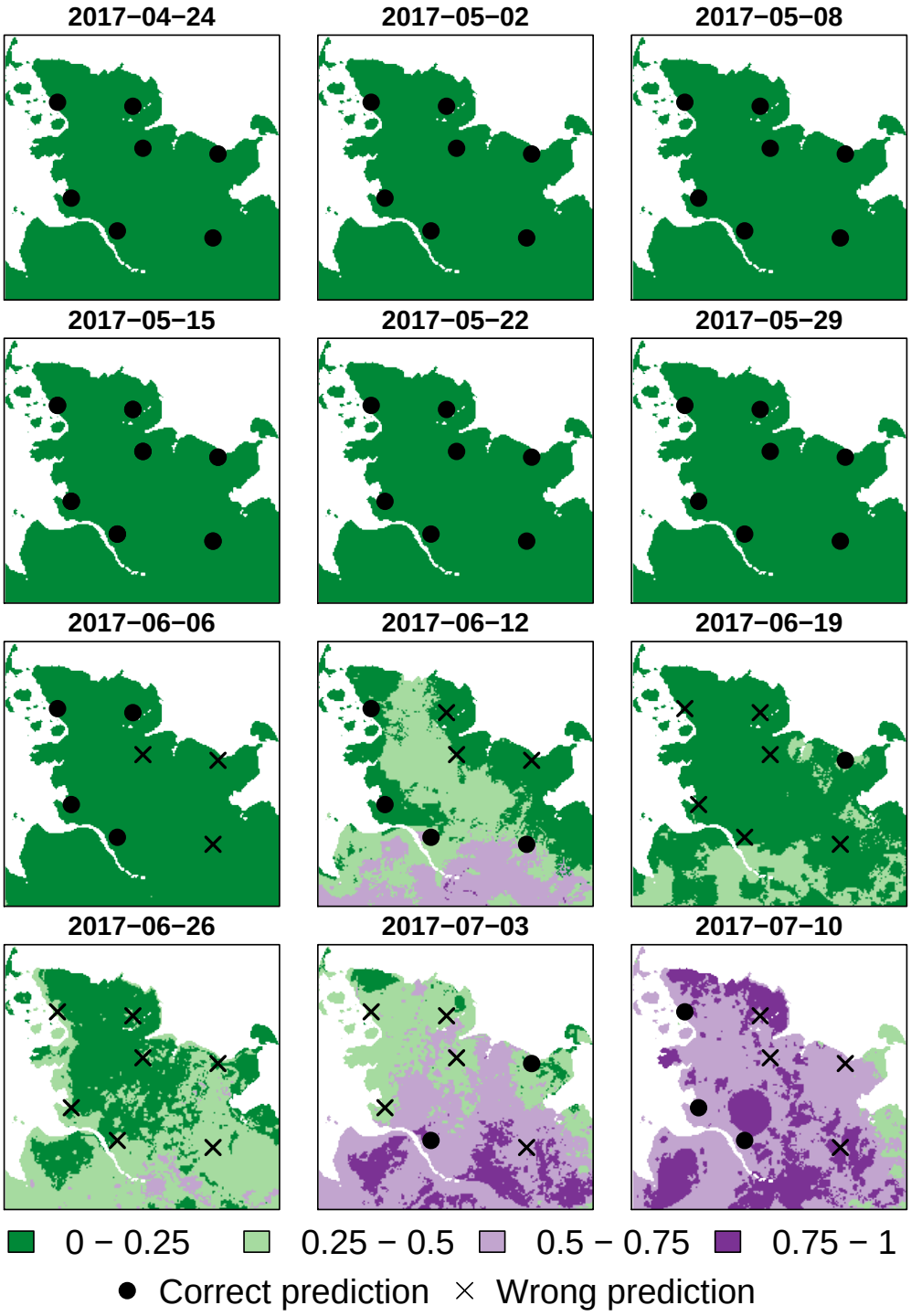


Figure 4.30: Spatial prediction of the probability of severe brown rust infections in 2017 using the Boosted Decision Tree (BDT) procedure

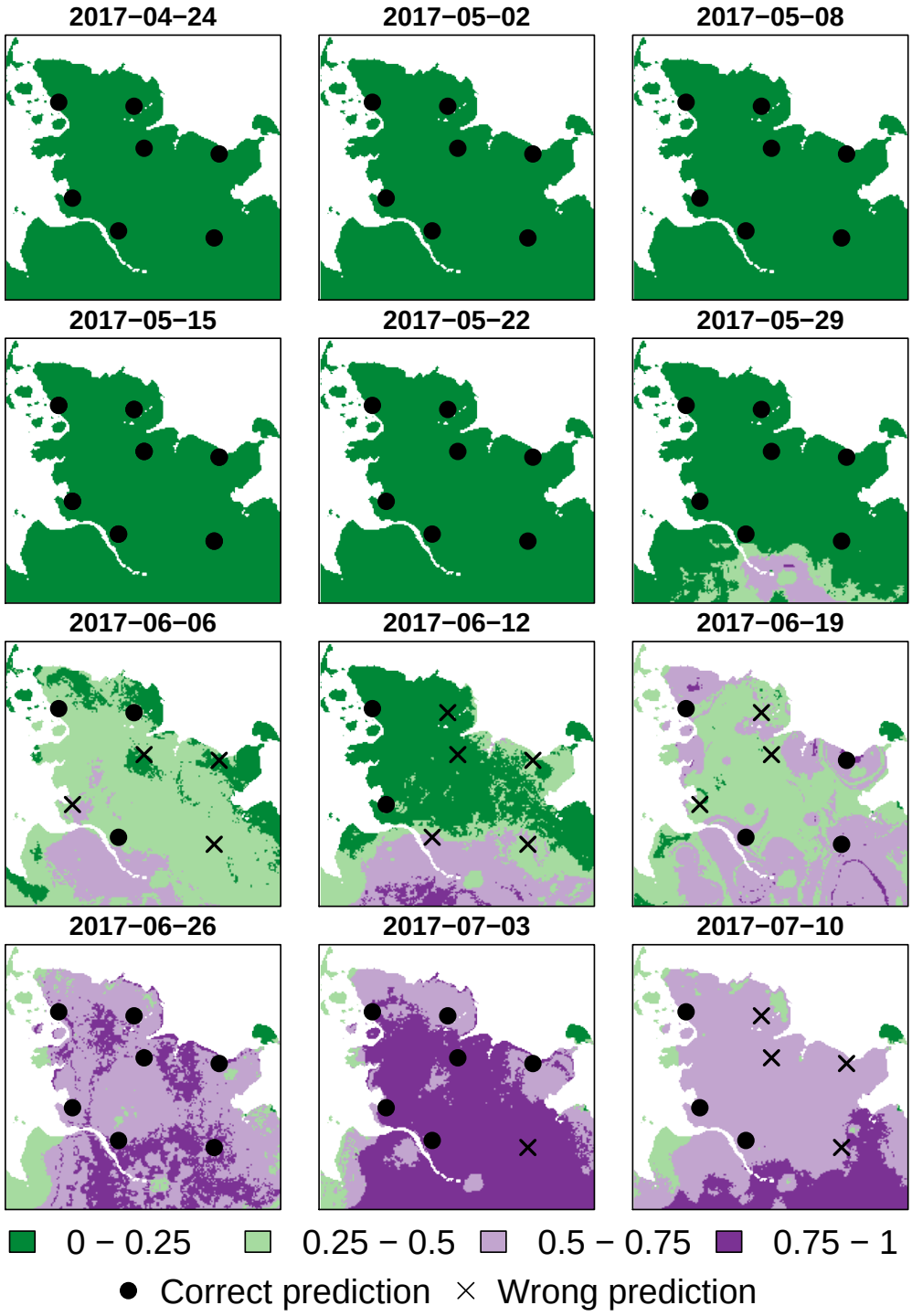


Figure 4.31: Spatial prediction of the probability of severe brown rust infections in 2017 using the Random Forest (RF) procedure

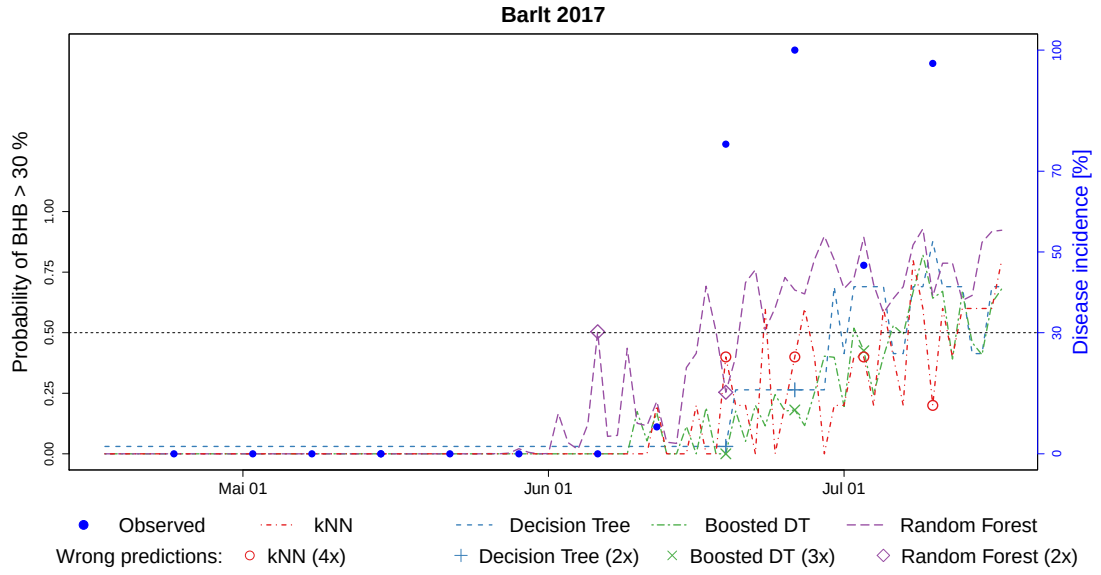


Figure 4.32: Temporal prediction of the probability of the exceedance of the brown rusts disease threshold (30 %) in Barlt

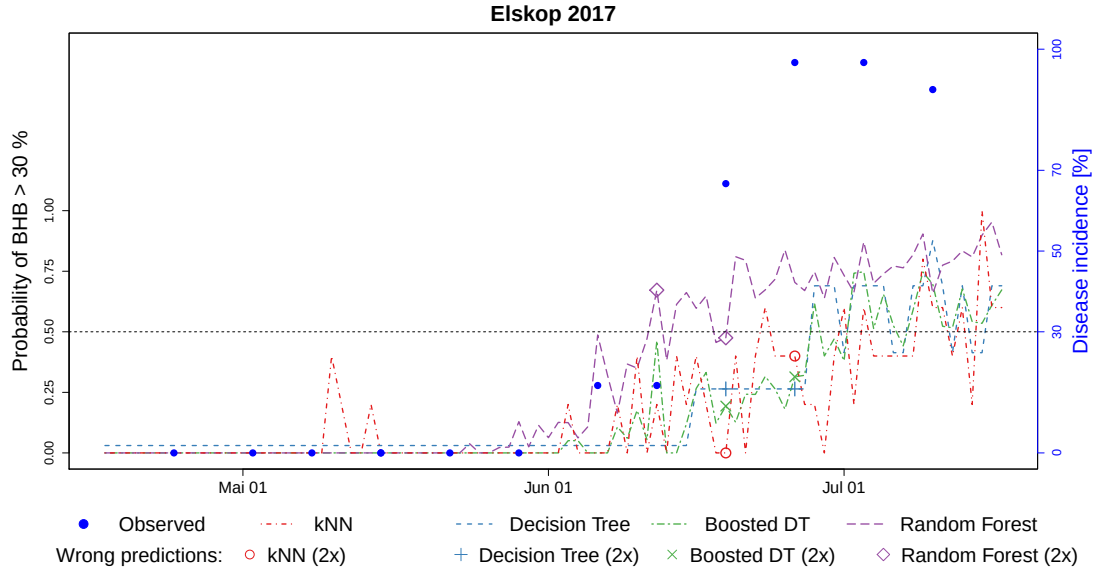


Figure 4.33: Temporal prediction of the probability of the exceedance of the brown rusts disease threshold (30 %) in Elskop

and began to exceed the threshold at the beginning of June. The probability predicted by the k-NN approach increased at the same time, resulting in a correct forecast of the first observation. The other methods predicted this event as *No exceedance*. Before the second exceedance was observed, the RF methods predicted probability exceeded the relevant 50 % value, but was reduced by the time of the observation. Therefore the second exceedance was not predicted as such by any method. The following three exceedances were predicted correctly by the RF procedure. Only the last exceedance was predicted correctly by all methods, while none was able to predict the decrease in the disease incidence at the end of the observation period.

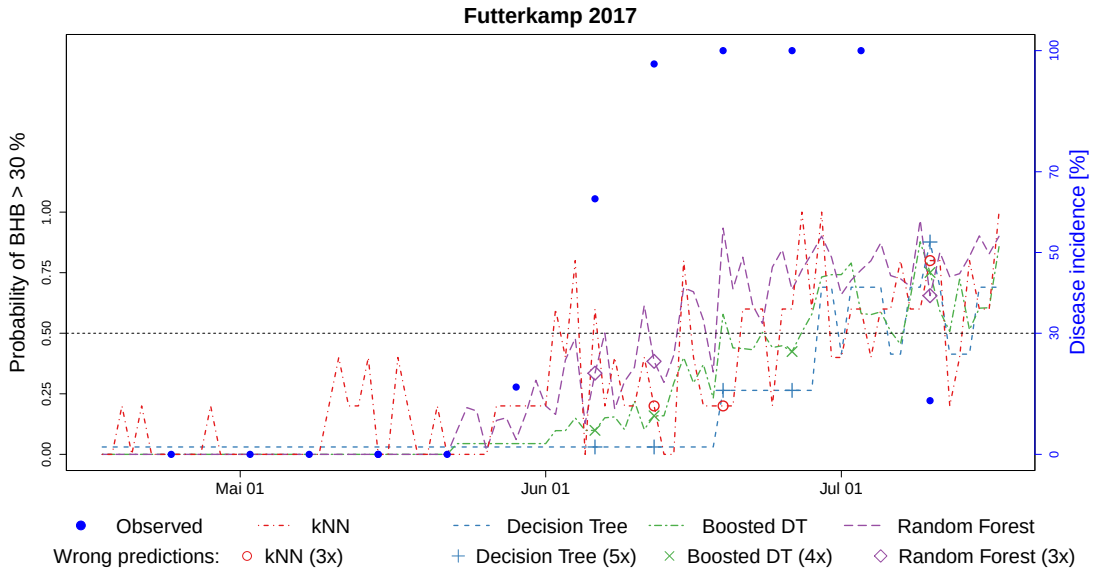


Figure 4.34: Temporal prediction of the probability of the exceedance of the brown rusts disease threshold (30 %) in Futterkamp

Figure 4.35 shows the results for Kastorf. Here the disease incidence exceeded the damage threshold of 30 % at the beginning of June. The first exceedance was not predicted correctly by any method, although the probability predicted by the RF model rose above the value of 50 % shortly after. Nevertheless, the only methods predicting the second exceedance correct were the BDTs. The next observation was only predicted correctly by the RF, followed by an observation only the RF and k-NN methods got right. The last two observations were underruns which were only predicted correctly by the k-NN approach.

Figure 4.36 shows the results of the prediction for Kluvensiek. As in Futterkamp and Kastorf the disease incidence exceeded the threshold in early June. The first three exceedances were not predicted as such by any method. Only the RF model predicted *Exceedance* before the third exceedance, but not at the same time. The

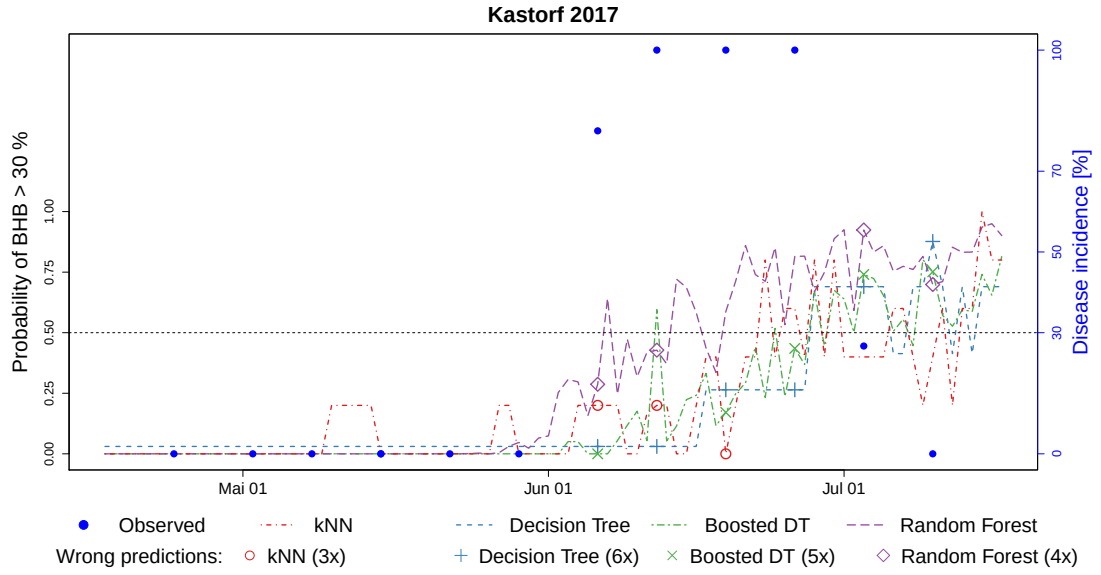


Figure 4.35: Temporal prediction of the probability of the exceedance of the brown rusts disease threshold (30 %) in Kastorf

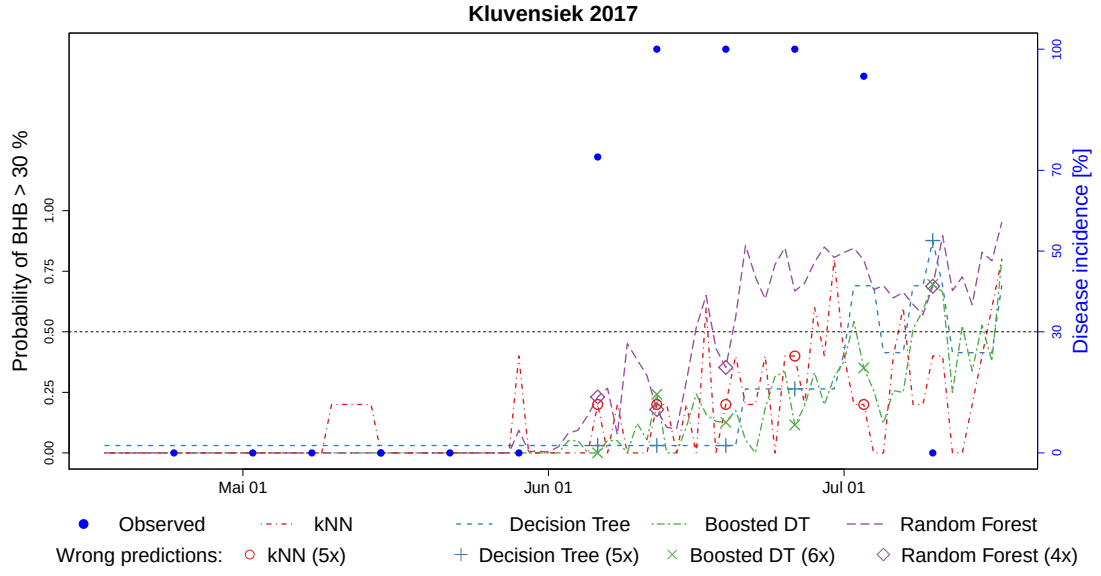


Figure 4.36: Temporal prediction of the probability of the exceedance of the brown rusts disease threshold (30 %) in Kluvensiek

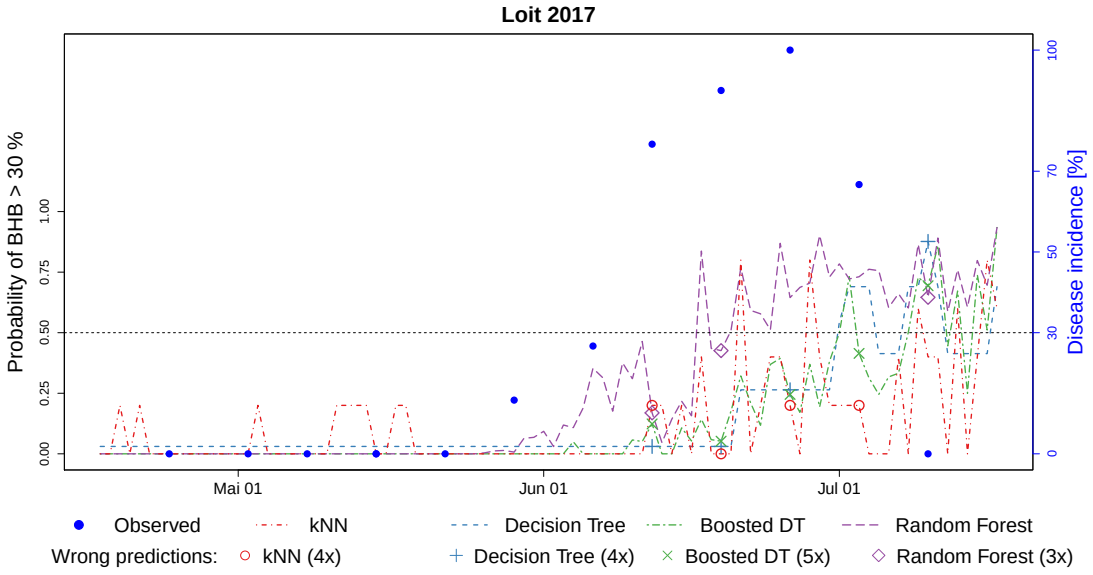


Figure 4.37: Temporal prediction of the probability of the exceedance of the brown rusts disease threshold (30 %) in Loit

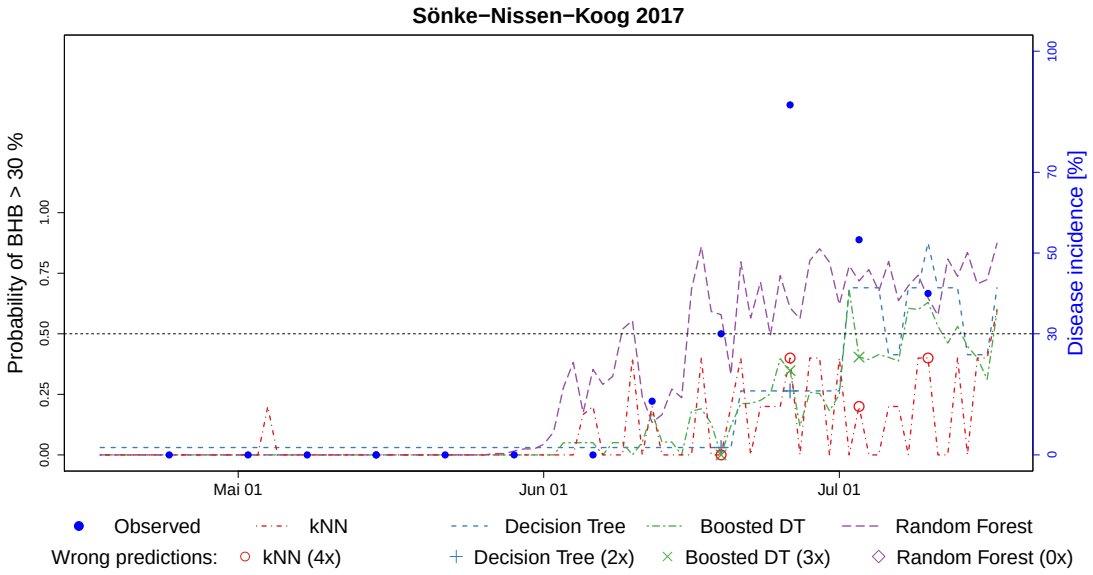


Figure 4.38: Temporal prediction of the probability of the exceedance of the brown rusts disease threshold (30 %) in Sönke-Nissen-Koog

fourth exceedance was only predicted correctly by the RF, and the last one by the RF and the DT model. The last underrun, however, was only predicted correctly by the k-NN method.

In Loit (figure 4.37) the disease incidence exceeded the damage threshold in mid June. The first two exceedances were again not predicted as such by any method, although the RF procedure predicted exceedances between those observations. The third exceedance was predicted correctly by the RF as well as the fourth one, which was also predicted as *Exceedance* by the DT model. The last underrun was, similar to Kluvensiek, only predicted correctly by the k-NN method.

In Sönke-Nissen-Koog (figure 4.38) the disease incidence exceeded the damage threshold quite late in the season by the beginning of July. The random Forest predicted every observation correctly at this location, while the k-NN method did not predict any exceedance at this location. Also the DT and BDT models missed the first two exceedances, with the BDT methods even missing the third one.

Summary of the prediction of brown rust events

Performance of brown rust models

- With the methods Decision Tree and Boosted Decision Tree it was possible to achieve the highest accuracy, specificity and precision in the prediction of yield-relevant brown rust events followed by *k*-Nearest Neighbor.
- Using the Random Forest method, far lower accuracy and specificity but also higher sensitivity were achieved in the prediction of brown rust events.

Spatial characteristics

- Taking into account the predictions of the individual sites, the ROC AUC is approximately 0.7 for *k*-Nearest Neighbor and 0.8 for the other methods.
- A larger number of sites included in the prediction increases the forecast quality.

Temporal characteristics

- Considering the results for the individual years, ROC AUC is 0.7 for *k*-Nearest Neighbor and 0.9 for the other methods.
- An improvement of the forecast quality occurs if a higher number of years are included in the prediction.
- The CTU, affected by time and temperature, has a strong influence on the forecast of the probability of yield endangering brown rust events.

Chapter 5

Discussion

The previous chapters introduced a novel approach to the spatial prediction of yield endangering infestations of winter wheat with different pathogens. The results showed differences in prediction accuracies, which are strongly dependant on the machine learning method used and the predicted pathogen. Due to the high influence of the pathogens studied on the models' performances, the predictions are first examined separately before the entire approach is discussed.

5.1 Powdery mildew prediction

Discussion and interpretation of results When predicting the exceedance of the damage threshold of powdery mildew, the results of the holdout validation and the real time modelling of 2017 showed the highest overall accuracy in the DTs (tables 4.1 (p. 69) and 4.5 (p. 77)). The LOOCV method (table 4.2 (p. 71)) showed the highest accuracy for the BDTs. Also the specificity, precision and ROC AUC values were highest for both procedures. However, the greatest sensitivity was achieved by RFs in all comparisons. Thus, the RF method was able to predict 58 % (table 4.2) to 89 % (table 4.5) of the damage threshold exceedances correctly. Nevertheless, the best values for sensitivity coincide with the worst precision and specificity. Only 30 % (table 4.5) to 40 % (table 4.1) of the exceedances predicted by the RFs also were observed. Similarly, only 55 % (table 4.5) to 60 % (table 4.1) of the underruns of the damage threshold could be predicted correctly. This can be attributed to the functions that have been introduced to optimise the prediction of exceedances. As described in chapter 3.3.3 and shown in appendix B.5 the functions *kv_fit*, *evtrials* and *evclasswt* were designed to modify the machine learnings algorithm to achieve the best ROC AUC values possible. In the k-NN method, however, only the number of neighbours could be

affected and in the BDTs the number of improvement attempts could be influenced. In the RF process, in addition to the number of splits at the nodes, the target elements were also weighted. With this weighting, the class imbalance problem visible in figure A.1 could be addressed much more directly than with other modifications made. The influence of the weighting functions is also apparent when comparing the previous works of Hamer et al. (2016b). There, the Random Forest model was not calibrated using the *evclasswt* function, which led to a high accuracy of 71 % but also to a low sensitivity of 59 %. The modified RF approach, therefore, predicts more frequent exceedances of the damage threshold, which improves the prediction of these exceedances at the expense of overall accuracy. These modifications were developed in response to discussions with business users who valued the prediction of exceedances more than the models accuracy.

Taking a closer look at the details of table 4.3, it is noticeable that with the LOOCV prediction of individual locations it is not possible to predict them satisfactorily with any of the tested machine learning methods. Table 4.4, on the other hand, shows that high ROC AUC values can be achieved concerning the respective year. This may be explained by the strong spatial dependency of the occurrence of the powdery mildew. As first illustrated in figure 3.6 higher powdery mildew infestations are recorded on the eastern coast, and only very low disease incidences were observed on the western coast. Figure A.5 shows that this does not change during the season. According to the observations of Klink (1997) and Verreet et al. (2000), the absence of powdery mildew is attributed to the topographical and climatic conditions of Schleswig-Holstein's west coast. For one year, the machine learning methods can easily predict this spatial difference by assigning a west coast location such as Barlt (p. 49) low probability of exceedance and high probabilities to an east coast location such as Futterkamp. This would lead to sufficient results in table 4.4. However, if the machine learning method always predicts low probabilities for a location like Barlt an exceedance like in 1996 (compare figure A.3 for Ritmo) produces a more severe weight and leads to a bad ROC AUC value. This strong spatial factor can also be seen in the models created by the RF and DT methods. Figures 4.4 and 4.8 reveal a high influence of the multi-annual climatic parameter real evapotranspiration on the created RF models. Also, figure 4.5 depicts an early split concerning many observations according to the real evapotranspiration, which separates Schleswig-Holstein into two halves, as figure 4.6 shows. The western part of the investigated area, which has higher evapotranspiration values, is classified as less endangered by mildew. Chapter 2.1.1 describes the environmental effects on the infection of mildew. Also, Yarwood (1957) recognised that powdery

mildew, unlike other airborne pathogens, was able to infect host plants even at low humidity levels. In combination with the model's split decision, this could suggest a long-term advantage of powdery mildew compared to other pathogens under slightly different climatic conditions, such as lower than average evapotranspiration. This spatial dependency is also expressed in the spatial predictions of the models (figures 4.9 to 4.12). The clearest spatial division is made by the DT and BDT algorithms (figures 4.10 and 4.11). The spatial prediction by the RF model (figure 4.12) also shows this division, but includes predictions above 50 % in the western part, probably caused by the weighting to the ROC AUC by the *evclasswt* function included in the model generation. The k-NN method also separates the westernmost part of the study area, but since no accessible model is created in the process of predicting using this technique, it was not possible to identify the parameter with the largest influence on this separation. Only the behaviour of the k-NN predictions can be used to conclude the parameters taken into account. The sharp and continuous differentiation of the westernmost area indicates the influence of climatic parameters. The curve progressions in figures 4.14 to 4.19 on the contrary show the influence of weather parameters, which are variable through the season, on the k-NN prediction. Contrary, the DT curve is very stable over time. Especially in Kluvensiek (figure 4.17), the only temporal influence of the DTU is clearly visible.

The quality of the forecast, taking into account different numbers of years and locations (figure 4.3), also points out the spatial influence emanating from the multi-annual climatic parameters. Five to six locations are required to receive appropriate results testing the models. Independent of the modelling technique the number of years considered for the calibration of the model is of minor importance. Consequently, the models assign greater importance to the location of the sites than to the weather conditions that vary over the years, which can be observed in the DT model (figure 4.5). Apart from the CTU, there is no variable parameter over the season. Although the CTU does not have such a high influence on the RF model (figure 4.8), the DTU and the mean air temperature are attested to have a great influence. Hence, the models have found the biggest influence on the behaviour of mildew in air temperature. This is also consistent with the known influences, described by Friedrich (1994), Adams et al. (1986) and Hau and de Vallavieille-Pope (2006), on the pathogen's life cycle as described in chapter 2.1.1.

Another striking feature in the DT model (figure 4.5) is the split based on the susceptibility class. Wheat varieties with a high susceptibility according to (Bundessortenamt, 2017) lead to the model's prediction of low infestation probabilities. As figure

A.1 shows, the plants of classes 6 and 8 account for only a tiny part of the data basis. In addition, only a few exceedances were observed in class 9 plants, contrary to the classification. Class 2 plants, on the other hand, showed more exceedances of the damage threshold than those of classes 3 and 4, which explains why this susceptibility class is identified as potentially more vulnerable. The split can, therefore, be traced back to the data basis.

Comparison with existing models Comparison with existing model approaches for the prediction of the powdery mildews behaviour as presented in chapter 2.1.3 is difficult because they often have different objectives or were not validated by the authors, like the models of Hau (1985) and Willocquet et al. (2008). The model of Friedrich (1994) was also only checked for the individual parts, not the probability of infection.

The MEVA-PLUS model of Bruns (1996), which includes the GEMETA model of Hau (1985), has been checked against predicted and actual disease severity. He used the first observed infestation to calibrate the model for the current season and achieved an R^2 larger than 0.2 until BBCH stage 49. A direct comparison of the results obtained in this work with the results of Bruns (1996) using the measure of determination is not possible since the R^2 is used for metric correlations and categorical variables are predicted in this work. Up to BBCH level 49, however, there was at least a weak correlation between the forecast and the actual situation. In order to continue to achieve such results, Bruns (1996) started a further calibration of the model at this point in time. This shows a fundamental difference between Bruns (1996) approach and the one presented here. Many popular models such as MEVA-PLUS (Bruns, 1996), WHEATPEST (Willocquet et al., 2008) and InfoCrop (Aggarwal et al., 2006) require up-to-date infection data to start the prognosis. However, since the aim of this work is a spatial prediction, it is only possible to use spatially distributed data, which exclude the current infestation situation, although this, as shown by Bruns (1996) work, simplifies a reliable prediction considerably. The same applies to the predictions of the WHEGROSIM model approach by Rossi and Giosuè (2003). WHEGROSIM uses the model approach of Friedrich (1994) to predict the disease incidence of powdery mildew on wheat plants. This approach achieves a very high coefficient of determination of 0.89, but like the other approaches, it requires an initial infestation value to achieve this accuracy (Rossi and Giosuè, 2003).

5.2 Brown rust prediction

Discussion and interpretation of results The validation of the models fitted to the brown rust pathogen shows only slight differences between the holdout and the LOOCV evaluation procedure (tables 4.6 (p. 91) and 4.7 (p. 93)). Both techniques showed the highest accuracy, specificity, precision and ROC AUC for the BDT method. The highest accuracy was also achieved by the DT method as well as the precision using the LOOCV evaluation procedure. The highest sensitivity again was achieved by the RF method. While the other methods only reached a sensitivity around 40 % the RF approach reached a sensitivity around 70 % in the holdout and LOOCV evaluation. In contrast to the prediction of the powdery mildew, all statistical measures of performance, except sensitivity were rather close together. Thereby the values generated by the LOOCV method were marginally lower than those determined by the holdout method. This also can be seen in the ROC curves (figures 4.20 and 4.21). Both figures show similar results. The curves of the machine learning procedures are close together with the exception of the k-NN approach. While the true positive rate for the other procedures still shows a low false positive rate of 20 % for 80 %, the false positive rate for the k-NN procedure for a similar true positive rate is 70 %. This clearly speaks against the use of the k-NN approach to predict the brown rust infestations.

In the forecast for 2017 (table 4.10 (p. 98)) the RF approach achieved the highest value for each statistical measure, with the exception of specificity. The accuracy and sensitivity were lower for all procedures than for the iterative evaluations while specificity, precision and ROC AUC were higher. The RF approach correctly predicted 58 % of the exceedances that occurred, and BDTs succeeded in just 16 % of the cases (table 4.10). In the iterative LOOCV test, these values were 69 % and 37 %, respectively (table 4.7). This would indicate a year with a rather high number of brown rust infestations. An assumption confirmed by the visual impression of figure A.4 (p. 149). This also explains the high precision values. If more exceedances occur but are no longer predicted, the proportion of correct forecasts of exceedances increases. The specificity is also increased, as with more predictions of an underrun in the case of less observed disease incidences below 30 %, the proportion of the correct predictions of the underruns in all occurred increases. This shows that all statistical measures must be taken into account when evaluating the machine learning methods.

However, the spatial and temporal behaviour of the predictions must also be examined. Considering the ROC AUC values of LOOCV in relation to the individual locations (table 4.8) and years (table 4.9), all procedures with the exception of k-NN show consistently high values, which corresponds to the ROC curves for all locations

and years. Most often the BDTs show the highest values. Particularly in the case of values based on years, the RFs and DTs often have equal ROC AUC values. Additionally, there are differences in the spatial predictions during a season. Particularly noticeable is the homogeneity with which the DT executes the spatial predictions (figure 4.29). In the first few weeks, no procedure says that an exceedance is expected (figures 4.28 to 4.31). However, when the predicted exceedances begin the forecast made by the other methods is considerably more spatially differentiated in comparison to the DT approach. In contrast to mildew, the most important influencing variables on the DTs decisions (figure 4.24) are not climatic but weather-influenced variables. The first decision assigns a very low probability of exceeding the damage threshold to all observations with a CTU of less than 634.78, which on average corresponds to the 172nd day of the year (figure 4.25). This day would be June 21, 2017 in the current season. The location data (figures 4.32 to 4.38) show a slight increase in the probability of DTs beforehand, but the actual exceedance is not forecast for all locations until two weeks later. Like the DTs, no other system was able to predict the first transgression at all locations correctly. This was best achieved with brown rust using the RF method, which correctly predicted the first exceedance of the damage threshold at the three locations Barlt (figure 4.32), Elskop (figure 4.33) and Sönke-Nissen-Koog (figure 4.38), but in Barlt and Elskop also incorrectly classified one underrun before. The only other algorithm that correctly predicted the first exceedance was the k-NN approach which succeeded in Futterkamp (figure 4.34).

Besides the major split of the DT model (figure 4.24) by the CTU, further divisions show regularities in the behaviour of the brown rust. After the next split by the CTU, the following splits of both sites occur based on climatic parameters. As the comparison with figure 4.26 shows, these splits address the climatic situation of Fehmarn, which has on average a higher minimum temperature and lower monthly precipitation. The comparison with figures A.6 and A.4 shows, that the region of Fehmarn does not show any exceedances at all, but this observation is only based on three years of which two years did not show exceedances at the other locations but Futterkamp as well. There is no evidence in the literature that a high minimum air temperature can lead to a reduction of the brown rust infestation. In contrast, Roelfs et al. (1992) point out that the duration of the infection depends on the temperature and that a higher value means faster infection. The split of the model (figure 4.24) at node 13 also shows a higher probability at a temperature above 15 degrees Celsius, which means that it is within the ideal range between 15 and 25 degrees Celsius, as indicated by Simkin and Wheeler (1974), and thus confirms the literature. However, there is proof of the

influence of reduced precipitation. Sache (2000) has already found that precipitation can increase sporulation. Also, Hau and de Vallavieille-Pope (2006) and Simkin and Wheeler (1974) noted that brown rust requires high humidity for infection. Since high humidity is required to form precipitation as Gouache et al. (2015) pointed out, node 11 also addresses this aspect. Of course, this aspect is also described in node 14, where a humidity of more than 90 % causes the prediction of a high probability of exceeding the damage threshold. Furthermore, when looking at the DT model (figure 4.24), it is noticeable that the susceptibility classes 1,2,3,5,7 and 9 are assumed to undergo no exceedance of the threshold (node 5). The comparison with figure A.2 and table 3.2 shows that classes 1, 2 and 7 do not appear in the calibration data set at all. Since they are stored as possible value attributes of the factor variable, they were considered as underrun of the threshold by the DT algorithm. This is to prevent the output of error values when applying the models to new data records. At first glance, it may come as a surprise that wheat of the high susceptibility class 9 should not be expected to exceed the damage threshold, while wheat class 4 is more at risk. Figure A.2 shows that wheat of class 9 is proportionally less likely to experience exceedances than wheat of class 4. The data for class 4 are based on insufficient observations of the years 2000 and 2011, as shown in table 3.2.

Comparison with existing models As with the powdery mildew, no existing model is known for brown rust that predicts the probability of the pathogen-specific damage threshold being exceeded. However, as described in chapter 2.2.3, some models try to predict the behaviour of the brown rust. A comparison with the WHEATPEST model of Willocquet et al. (2008) turns out to be problematic because it models the influence of observed disease severity on plant-specific parameters rather than the behaviour of the pathogen itself.

The RUSTDEP model of Rossi et al. (1997), on the other hand, was aimed at predicting disease severity of brown rust on winter wheat. It uses comparable temperature, precipitation and humidity data, but also requires the first observed disease severity, which is, of course, a non-spatial variable. Using a holdout validation method, the model predicted 80 % of the metric values in such a way it falls into the confidence interval of the observed data. If this is compared with the accuracy of the holdout evaluation of the method presented here (table 4.6), it is noticeable that similar, if not slightly better values have been achieved especially by the DT based approaches.

Gouache et al. (2015) used the observations of 400 field tests over 30 years to model the influence of different climate parameters on the brown rusts disease severity with regression equations. The climate variables are derived from temperature, precipitation

and evapotranspiration. Thereby the model received a ROC AUC value of 0.85, which is comparable to the ROC AUC values achieved by the machine learning methods described here, except for the k-NN approach. These are in the range between 0.81 (table 4.21) and 0.91 (table 4.10). However, the figures 4.22(b) to 4.22(d) show that the ROC AUC value increases with more incoming locations and seasons. This suggests that with a data record with a number of years, similar to Gouache et al. (2015), a higher value could be possible.

5.3 Synthesis

The work shows that it is possible to predict both pathogens using the model approach (figure 3.7), with brown rust producing consistently better results than powdery mildew. The machine learning processes discovered a stronger spatial component in the prediction of powdery mildew and a stronger temporal component in the prediction of brown rust, as the parameters identified by DT models (figure 4.5 and 4.24) demonstrate.

General assessment of considered variables For both pathogens, a clear correlation with the CTU was determined by various machine learning methods. The CTU value represents, as described in chapter 3.3.2, the development of the host plant based on the weighted cumulative temperature. The importance of the CTU value can thus imply a connection with the growth state of the host plant on the one hand and a connection with the temperature conditions themselves on the other hand, which are also of importance for the pathogens (Paulus, 1990; Beest et al., 2008; Eckhardt et al., 1984; Friedrich, 1995b; Simkin and Wheeler, 1974). Also, a connection with the purely temporal character of this covariate cannot be excluded.

Of the climatic parameters, the real evapotranspiration stands out, as discussed in chapter 5.1. However, the multi-annual climatic temperature as well as the daily aggregated weather air temperature values also occur in the DT and RF models and influence the decisions made as a result of these. This confirms the previously made statement about the importance of air temperature for the prediction of pathogens.

The susceptibility class was also identified as an important parameter for both pathogens. As has already been explained, a higher susceptibility class did not necessarily mean a higher probability of exceeding the pathogen-specific damage threshold. Thus the models recognised what can already be seen in the figures A.1 and A.2. Correspondingly, the question is whether this parameter, for which the expectation does not coincide with the observations, should be taken into account in modelling.

Alternatively, a classification according to variety name could be used. However, this work did include the use of the susceptibility class, since the number of samples per class was increased and the comparability of the results was improved. An additional problem could be the different number of samples. If there is only a small amount of data for individual varieties or classes from afflicted years, less susceptible varieties can also appear to be more susceptible if they are compared with varieties for which more data from more years are available. If necessary, under-represented wheat varieties should be excluded.

In addition to the wheat varieties, this also applies to the sampling sites. As described in chapter 5.2, Fehmarn has a major influence on the brown rust prediction, although data is only available for a few years. Exclusion of the data of this site for the benefit of the model should also be considered, but removing site observations is critical given the number of locations. Particularly when looking at the forecast quality of the powdery mildew prognosis concerning the sites considered (figure 4.3), an improvement in the prediction can be seen with an increasing number of locations. It can be concluded that a further forecast optimization can be achieved with a higher number of sampling sites. This improvement also occurs when predicting brown rust events, but also with a strong influence of the available number of years. The available locations can, therefore, be assumed to represent the entire infestation process of this pathogen better than the behaviour of powdery mildew.

The geographical arrangement of the sampling sites in the study area should also be taken into account. Map 3.4 demonstrates that the IPS monitoring stations are irregularly distributed in Schleswig-Holstein. Due to the characteristics of Schleswig-Holsteins natural regions (described in chapter 3.1) the study sites are located in the eastern uplands and the marshes. In the high and outwash plains, wheat is cultivated to a smaller amount (map 3.2), so IPS monitoring fields do not represent these areas. With more regular sampling, the central region of the study area could also be represented in the data record. Especially for powdery mildew, this could contribute to a more differentiated prediction instead of the sharp division of the investigated area by machine learning techniques, as can be seen in figures 4.10 and 4.11.

A solution to this problem could be innovative surveying techniques. For example, Franke and Menz (2007), Zhang et al. (2014) and Cao et al. (2015) investigated the use of multi and hyperspectral data sets obtained by remote sensing to illustrate the current course of infestation. If reliable infestation rates could be determined using these methods in the future, this would have several advantages. On the one hand, larger areas could be sampled. A higher temporal frequency, with intervals of less than

one week, would also be conceivable. In the framework of this work, an attempt was also made to close the gaps in time between the weekly sampling dates by adopting linear and polynomial progressions. Since no approach could improve the model quality and the literature could not support the choice of the temporal interpolation method, a calculation of the values between weekly samplings was discarded. On the other hand, non-invasive sampling would also have the advantage that the sampler exerts no influence on the course of the disease. Buttner and Stetzenbach (1993) have already found out that human activity in a field stock, as it occurs during sampling, leads to a higher spore concentration in the air and the pathogen can thus spread better, which could be avoided by airborne procedures.

When looking at the parameters, influencing the predictions results, it is striking that the wind speed is not taken into account in the DT models. In RFs, wind speed is also considered to be at most the fourth most important parameter. Considering that both fungi are wind-borne pathogens, this supposedly small role of wind speed can irritate. One possible explanation could be that the wind conditions in Schleswig-Holstein are such that wind speed is not a limiting factor for the spread of pathogens. Figure 3.5 shows that wind speeds between 3 and $6\,\mathrm{m\,s^{-1}}$ are achieved on average in the entire study area. The wind speed increases to the north and towards to the coasts. Cao et al. (2012) stated that a maximum of 0.6 to $2\,\mathrm{m\,s^{-1}}$ must be achieved to spread mildew. Geagea et al. (1997) determined a minimum wind speed of 1.3 to $1.8\,\mathrm{m\,s^{-1}}$ for the distribution of the brown rust. These minimum speeds are thus below the climatic average wind speeds in the study area. The wind speed required for spore release could thus be reached so often that it does not limit the spread of pathogens. However, this line of thought conceals two facts. On the one hand, the values measured by the weather stations do not correspond to the wind speeds in the field canopy, even if they were calculated to this height by the logarithmic wind function (chapter B.1). On the other hand, Cao et al. (2012) already described that the spread of the conidia is most likely to be triggered by short, strong gusts, which would be most likely to be represented by the variable of the maximum wind speed. This variable does not occur in the DT models at all and in the RF models it is only under the ten most important variables (figure 4.27). If the problem should be a wrong representation of the wind gusts in the population, it would hardly be possible to solve it in the context of a large-scaled prognosis, since it is not possible to measure the gusts over a large area because of the size of the area.

Comparison of machine learning approaches It is difficult to make a clear decision in favour of a machine learning process since no method is clearly superior to

another. However, the k-NN approach has emerged as the weakest method for the applications presented here. Apart from the specificity in table 4.10, this method has never achieved the highest statistical measure of performance and while in the ROC curves of the powdery mildew (figures 4.1 and 4.2) it's performance appeared to be only slightly worse, this is clearly the case with the curves of the brown rust (figures 4.20 and 4.21). Besides, the modelling behaviour of the k-NN approach is not as transparent as it is for the other machine learning techniques. The ensemble DTs, such as the BDTs and the RFs, are not entirely open either. However, the most influential parameters can be identified and discussed via the "Mean Decrease Accuracy". Access is even better via the DTs, which allow following the decisions made by the trees. Such transparency is crucial because the algorithms cannot search for causal connections. The evaluation and interpretation of the rules provided by the DT models is the responsibility of the user.

Looking at the statistical measures of the machine learning methods, except k-NN, it is noticeable that the values of the DTs and BDTs always lie next to each other, whereas the DTs for powdery mildew prediction slightly perform better (tables 4.1, 4.2 and 4.5). Except for sensitivity, the RF approach delivers considerably worse results predicting the probability of powdery mildew events. For both pathogens the RFs achieve clearly better sensitivity values than the other methods and for the brown rust predictions (tables 4.6, 4.7 and 4.10) only the specificity is noticeably lower. This is of course, as already described above, the consequence of the training of the RFs to weight exceedances more strongly than underruns. Accordingly, it is not possible to merely recommend a machine learning approach, as it depends on the needs of the user. If new knowledge about the connections between pathogens and weather parameters should emerge, it is advisable to choose a system that is as open as possible and which provides an understanding of the predictions made. The transparency of the DT approach provides this understanding of the connections. Besides, in most cases it offers the highest overall accuracy of prediction. However, if the requirement for the model is to classify as many exceedances as possible correct, also at the risk of misclassifying underruns, then the modified RF approach presented here should be used.

The work presented here aims to focus the attention of farmers on potential events that could endanger yields. The final decision as to whether and how the wheat is treated depends on the farmer himself. Therefore, a lower accuracy is accepted in favour of a higher sensitivity.

General assessment of the methodology All in all, the approach presented in figure 3.7 makes it possible to use interpolation and expertly supported summarising of weather data and the combination with climate data to generate machine learning models to generate pathogen-specific predictions. In this work, it is shown that based on a corresponding data basis for such a large area as Schleswig-Holstein predictions could be achieved with an accuracy of over 70 %.

As a possible criticism of the approach, it could now be justified that such a comprehensive database does not exist for many areas. On the contrary, the approach is highly adaptable. Even with a smaller number of weather stations, the interpolation methods can be used to generate spatial input data. Figures 4.3(b) to 4.3(d) and 4.22(b) to 4.22(d) also show that with a corresponding number of sampling stations, the infestation data from three years can be sufficient to calibrate reliable models. The approach can also be used in such a way that new models can be calibrated by using current prediction checks to improve the forecast. This has not been done within the scope of this work only in order not to influence the evaluation.

Chapter 6

Conclusions

This thesis presents a modeling approach that combines expert knowledge of pathogens with geostatistical and machine learning techniques (figure 3.7). The results show that the approach is well transferable to different pathogens and achieves an accuracy of more than 80 % for brown rust and more than 70 % for powdery mildew using iterative evaluations. By uploading the spatial and temporal forecasts generated by the model to the website, developed as part of this work, interested farmers are enabled to access them and include the predictions in their planning. Thereby, these forecasts should not be interpreted as direct treatment recommendations. Instead, it should encourage the farmer to monitor his stocks with greater attention to the pathogens in question. However, it is not possible to recommend a specific machine learning algorithm. Rather, a comparison of the performance of the prediction studied in the course of the evaluation showed that the selection should be based on the respective objective. A white box procedure such as the DTs is suitable for investigating the connections between the dangerous pathogen infestation and the weather and climate conditions. The RFs, on the other hand, have shown that they are an excellent counterpart to the class imbalance problem, which is common in these predictions, although this can lead to a reduction in overall accuracy.

The transferability of the approach to other pathogens results from its high adaptability. For example, different meteorological data could also be relevant for other pathogens. Other parameters could be interpolated if necessary and included into the approach. In another area, which has a more pronounced relief, different interpolation methods could, of course, lead to better interpolation results, which could easily be taken into account. Depending on the user's knowledge of the pathogen, the infection and incubation periods summarised in the course of the "Combination of regionalised weather data" (chapter 3.3.2) have to be adjusted, which is possible with a few in-

puts in script B.4. In addition to winter wheat, other host plants could of course also be included. The growth of these plants could be considered as a model parameter by using the DTU function (described in script B.1) with suitable information. Last but not least, the choice of a suitable machine learning process must be made. If knowledge from the application is to be gained, an open system such as the DT should be used, but other systems are also recommended which give an idea of the influence of the variable on the prediction. When selecting and adapting the machine learning method, it is important to consider any imbalanced classification problems and to carry out appropriate weightings. The newly developed adaptation functions presented here (chapter 3.3.3 and script B.5) are suitable for this purpose. However, the approach's purpose must also be considered, and the statistical measures of the model performance must be weighed against each other.

In future work, the transferability of the approach presented here could be further tested. On the one hand, the application of the models generated by this work in another area would be interesting. Since it is assumed that the relationships determined by the learning methods are pathogen-specific and not site-specific, the models should also be successful in other areas such as southern Germany. Since there is a clear difference between the West and East Coast locations, especially in the case of mildew infestation, it would be particularly interesting to see whether the correlations with the meteorological variables determined by the models are causal, or whether it is a pseudo-correlation. On the other hand, the application to other pathogens is also interesting. The application of the approach on two widespread wind-borne pathogens has already shown the transferability of the approach, but application to other pathogens, perhaps in other areas as well, could further explore the possibilities of the approach. In this context, it could be possible to compare the proven and established infestation surveys procedures with innovative, airborne methods.

Bibliography

Adams, G. C., Gottwald, T. R., and Leach, C. M. (1986). Environmental factors initiating liberation of conidia of powdery mildew. *Phytopathology*, 76(11):1239–1245.

Addiscott, T. (1977). A simple computer model for leaching in structured soils. *Journal of Soil Science*, 28(4):554–563.

Aggarwal, P., Kalra, N., Chander, S., and Pathak, H. (2006). InfoCrop: A dynamic simulation model for the assessment of crop yields, losses due to pests, and environmental impact of agro-ecosystems in tropical environments. I. Model description. *Agricultural Systems*, 89(1):1 – 25.

Agrios, G. (2005). *Plant Pathology*. Elsevier Science.

Altman, D. G. and Bland, J. M. (1994). Diagnostic tests. 1: Sensitivity and specificity. *BMJ: British Medical Journal*, 308(6943):1552.

Armstrong, J. and Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*, 8(1):69 – 80.

Audsley, E., Milne, A., and Paveley, N. (2005). A foliar disease model for use in wheat disease management decision support systems. *Annals of Applied Biology*, 147:161–172.

Aust, H.-J., Hau, B., and Kranz, J. (1983). Epigram - a simulator of barley powdery mildew / Epigram - ein Simulator des Gerstenmehltaus. *Zeitschrift für Pflanzenkrankheiten und Pflanzenschutz / Journal of Plant Diseases and Protection*, 90(3):244–250.

Babak, O. and Deutsch, C. V. (2009). Statistical approach to inverse distance interpolation. *Stochastic Environmental Research and Risk Assessment*, 23(5):543–553.

Bayes, T. (1763). LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions*, 53:370–418.

Beest, D. E. T., Paveley, N. D., Shaw, M. W., and van den Bosch, F. (2008). Disease-weather relationships for powdery mildew and yellow rust on winter wheat. *Phytopathology*, 98(5):609–617.

Benavides, R., Montes, F., Rubio, A., and Osoro, K. (2007). Geostatistical modelling of air temperature in a mountainous region of northern spain. *Agricultural and Forest Meteorology*, 146(34):173 – 188.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Advanced Texts in Econometrics. Clarendon Press.

Borges, P. d. A., Franke, J., da Anunciação, Y. M. T., Weiss, H., and Bernhofer, C. (2016). Comparison of spatial interpolation methods for the estimation of precipitation distribution in Distrito Federal, Brazil. *Theoretical and Applied Climatology*, 123(1):335–348.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis.

Bruns, J. B. (1996). *Untersuchungen zur wetterbasierten Befallssimulation und Verlustprognose von echtem Mehltau (Erysiphe graminis D.C. f. sp. tritici Marchal) an Winterweizen*. PhD thesis, Georg-August-Universität Göttingen.

Bundessortenamt (2017). Beschreibende Sortenliste. https://goo.gl/vk8xHX. Last accessed on March 3, 2018.

Buttner, M. P. and Stetzenbach, L. D. (1993). Monitoring airborne fungal spores in an experimental indoor environment to evaluate sampling methods and the effects of human activity on air sampling. *Applied and Environmental Microbiology*, 59(1):219–226.

Cao, X., Duan, X., Zhou, Y., and Luo, Y. (2012). Dynamics in concentrations of blumeria graminis f. sp tritici conidia and its relationship to local weather conditions and disease index in wheat. *European Journal of Plant Pathology*, 132(4):525–535.

Cao, X., Luo, Y., Zhou, Y., Fan, J., Xu, X., West, J. S., Duan, X., and Cheng, D. (2015). Detection of powdery mildew in two winter wheat plant densities and prediction of grain yield using canopy hyperspectral reflectance. *PLOS ONE*, 10(3):1–14.

Chemura, A., Mutanga, O., and Dube, T. (2017). Separability of coffee leaf rust infection levels with machine learning methods at sentinel-2 msi spectral resolutions. *Precision Agriculture*, 18(5):859–881.

Coakley, S. M., Line, R. F., and McDaniel, L. R. (1988). Predicting stripe rust severity on winter wheat using an improved method for analyzing meteorological and rust data. *Phytopathology*, 78(5):543–550.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, 22(3):239–252.

Cressie, N. (1991). *Statistics for Spatial Data*. Wiley Series in Probability and Statistics. Wiley.

Curtis, B., Rajaram, S., and Macpherson, H. G., editors (2002). *Bread Wheat - Improvement and Production*. FAO Plant Production and Protection Series. Rome.

Delwiche, S. R., Yang, I.-C., and Graybosch, R. A. (2013). Multiple view image analysis of freefalling u.s. wheat grains for damage assessment. *Computers and Electronics in Agriculture*, 98:62 – 73.

Diepenbrock, W., Ellmer, F., and Léon, J. (2005). *Ackerbau, Pflanzenbau und Pflanzenzüchtung: Grundwissen Bachelor*. UTB Grundwissen Bachelor. Ulmer.

DWD (2015). Vorschriften und Betriebsunterlagen (VuB 3) Beobachterhandbuch (BHB) für Wettermeldestellen des synoptisch-klimatologischen Mess- und Beobachtungsnetzes. https://goo.gl/hNkcZP. Last accessed on March 3, 2018.

DWD (2016a). DATA SET DESCRIPTION - Multi-annual grids of potential evapotranspiration over grass. https://goo.gl/iJaaXg. Last accessed on March 3, 2018.

DWD (2016b). DATA SET DESCRIPTION - Multi-annual station means for the climate normal reference period 1981-2010, for current station location and for reference station location. https://goo.gl/wc98Im. Last accessed on March 3, 2018.

Dybowski, R., Gransden, W. R., and Phillips, I. (1993). Towards a statistically oriented decision support system for the management of septicaemia. *Artificial Intelligence in Medicine*, 5(6):489 – 502.

Eckhardt, H., Steubing, L., and Kranz, J. (1984). Untersuchungen zur Infektionseffizienz, Inkubations- und Latenzzeit beim Gerstenmehltau Erysiphe graminis f. sp. hordei. *Journal of plant diseases and protection*, 91(6):590–600.

Eguía, P., Granada, E., Alonso, J., Arce, E., and Saavedra, A. (2016). Weather datasets generated using kriging techniques to calibrate building thermal simulations with {TRNSYS}. *Journal of Building Engineering*, 7:78 – 91.

Engel, C. (2015). *Analysis of weather based dynamics of yield losses of wheat pathogens and their threshold-oriented control on the physiology of plants and yield for winter wheat based on the over regional IPM-Monitoring Schleswig-Holstein (1995 - 2014)*. PhD thesis, Kiel University.

Ewert, F., Porter, J., and Honermeier, B. (1996). Use of AFRCWHEAT2 to predict the development of main stem and tillers in winter triticale and winter wheat in North East Germany. *European Journal of Agronomy*, 5:89 – 103.

Fahrmeir, L., Kneib, T., and Lang, S. (2009). *Regression: Modelle, Methoden und Anwendungen*. Statistik und ihre Anwendungen. Springer Berlin Heidelberg.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874. ROC Analysis in Pattern Recognition.

Fix, E. and Hodges Jr, J. L. (1951). Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, California Univ Berkeley.

Fogarty, J., Baker, R. S., and Hudson, S. E. (2005). Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005*, GI '05, pages 129–136, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada. Canadian Human-Computer Communications Society.

Food and Agriculture Organization of the United Nations (2017a). Faostat statistics database - landuse. https://goo.gl/AbmELV. Last accessed on March 5, 2018.

Food and Agriculture Organization of the United Nations (2017b). Faostat statistics database - pesticides use. https://goo.gl/beLNMi. Last accessed on March 5, 2018.

Food and Agriculture Organization of the United Nations (2017c). Faostat statistics database - production crops. https://goo.gl/SZRBAx. Last accessed on March 5, 2018.

Food and Agriculture Organization of the United Nations (2017d). Plant pests and diseases. http://www.fao.org/emergencies/emergency-types/plant-pests-and-diseases/en/. Last accessed on March 5, 2018.

Forte, R. M. (2015). *Mastering Predictive Analytics with R*. Packt Publishing Ltd.

Franke, J. and Menz, G. (2007). Multi-temporal wheat disease detection by multi-spectral remote sensing. *Precision Agriculture*, 8(3):161–172.

Fränzle, O. (2004). *Streifzug durch 6000 Jahre Landnutzungs- und Landschaftswandel in Schleswig-Holstein*, volume 41 of *EcoSys*, chapter Reliefentwicklung und Bodenbildung in Schleswig-Holstein, pages 11 – 35. Verein zur Förderung der Ökosystemforschung zu Kiel e.V.

Friedrich, S. (1994). *Prognose der Infektionswahrscheinlichkeit durch Echten Mehltau an Winterweizen (Erysiphe graminis DC. f. sp. tritici) anhand meteorologischer Eingangsparameter*. Mainz.

Friedrich, S. (1995a). Calculation of conidial dispersal of Erysiphe graminis within naturally infected plant canopies using hourly meteorological input prarameters. *Journal of Plant Diseases and Protection*, 102(4):337–347.

Friedrich, S. (1995b). Calculation of the incubation period of powdery mildew under field conditions. *Journal of Plant Diseases and Protection*, 102(4):348–353.

Friedrich, S. (1995c). Modelling infection probability of powdery mildew in winter wheat by meteorological input variables. *Journal of Plant Diseases and Protection*, 102(4):354–365.

Fürnkranz, J. (1997). Pruning algorithms for rule learning. *Machine learning*, 27(2):139–172.

Gajdar, J. and Paqué, K. (2015). *Der Untergang eines Imperiums*. Springer Fachmedien Wiesbaden.

Geagea, L., Huber, L., and Sache, I. (1997). Removal of urediniospores of brown (puccinia recondita f.sp. tritici) and yellow (p. striiformis) rusts of wheat from infected leaves submitted to a mechanical stress. *European Journal of Plant Pathology*, 103(9):785–793.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*. Applied geostatistics series. Oxford University Press.

Gouache, D., Lon, M. S., Duyme, F., and Braun, P. (2015). A novel solution to the variable selection problem in window pane approaches of plant pathogen climate models: Development, evaluation and application of a climatological model for brown rust of wheat. *Agricultural and Forest Meteorology*, 205:51 – 59.

Goudriaan, J. and Van Laar, H. (1994). *Modelling Potential Crop Growth Processes: Textbook with Exercises*. Current Issues in Production Ecology. Springer Netherlands.

Hack, H., Bleiholder, H., Buhr, L., Meier, U., Schnock-Fricke, U., Weber, E., and Witzenberger, A. (1992). Einheitliche Codierung der phänologischen Entwicklungsstadien mono-und dikotyler Pflanzen–Erweiterte BBCH-Skala, Allgemein. *Nachrichtenblatt des deutschen Pflanzenschutzdienstes*, 44(12):265–270.

Hallmann, J., Hallmann, A., and von Tiedemann, A. (2009). *Phytomedizin*. Grundwissen Bachelor. UTB GmbH.

Hamer, W., Klink, H., Duttmann, R., and Verreet, J.-A. (2016a). Anwendung eines maschinellen Lernverfahrens zur Vorhersage. *Bauernblatt (May 28, 2016)*, pages 38–39.

Hamer, W., Klink, H., Duttmann, R., and Verreet, J.-A. (2017). Echter Mehltau und Braunrost nun sicher vorhersagbar? *Bauernblatt (April 29, 2017)*, page 29.

Hamer, W. B., Verreet, J.-A., and Duttmann, R. (2016b). Räumliche und zeitliche Vorhersage der Eintrittswahrscheinlichkeit eines ertragsgefährdenden Mehltauereignisses an Winterweizen mit der Random-Forest-Methode. *AGIT - Journal für Angewandte Geoinformatik*, 2:342–352.

Hammett, K. and Manners, J. (1971). Conidium liberation in Erysiphe graminis. I. Visual and statistical analysis of spore trap records. *Transactions of the British Mycological Society*, 56(3):387–401.

Hanus, H., Heyland, K., and Keller, E. (2008). *Handbuch des Pflanzenbaues: Getreide und Futtergräser*. Handbuch des Pflanzenbaues. Ulmer.

Harikrishnan, R. and Ro, L. E. d. (2008). A logistic regression model for predicting risk of white mold incidence on dry bean in north dakota. *Plant Disease*, 92(1):42–46.

Harley, T., Bryan, J., Kanal, L., Taylor, D., and Grayum, J. (1963). Semi-automatic imagery screening research study and experimental investigation. volume I. Technical report, PHILCO CORP BLUE BELL PA.

Harteveld, D. O. C., Grant, M. R., Pscheidt, J. W., and Peever, T. L. (2017). Predicting ascospore release of monilinia vaccinii-corymbosi of blueberry with machine learning. *Phytopathology*, 107(11):1364–1371.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer.

Hau, B. (1985). *Epidemiologische Simulatoren als Instrumente der Systemanalyse mit besonderer Berücksichtigung eines Modells des Gerstenmehltaus*. Acta phytomedica. P. Parey.

Hau, B. and de Vallavieille-Pope, C. (2006). Wind-dispersed diseases. In Cooke, B., Jones, D. G., and Kaye, B., editors, *The Epidemiology of Plant Diseases*, pages 387–416. Springer Netherlands.

Hengl, T. (2011). *A Practical Guide to Geostatistical Mapping*. BPR Publishers.

Hengl, T., Heuvelink, G., and Stein, A. (2003). Comparison of kriging with external drift and regression-kriging. Technical report, ITC.

Herrera-Foessel, S. A., Singh, R. P., Huerta-Espino, J., Crossa, J., Yuen, J., and Djurle, A. (2006). Effect of leaf rust on grain yield and yield traits of durum wheats with race-specific and slow-rusting resistance to leaf rust. *Plant Disease*, 90(8):1065–1072.

Hiemstra, P., Pebesma, E., Twenhöfel, C., and Heuvelink, G. (2008). Real-time automatic interpolation of ambient gamma dose rates from the dutch radioactivity monitoring network. *Computers & Geosciences*. DOI: http://dx.doi.org/10.1016/j.cageo.2008.10.011.

Hodges, T. and Ritchie, J. (1990). The CERES-wheat phenology model. In Hodges, T., editor, *Predicting Crop Phenology*, chapter 12, pages 133–142. Taylor & Francis.

Holzworth, D. P., Huth, N. I., deVoil, P. G., Zurcher, E. J., Herrmann, N. I., McLean, G., Chenu, K., van Oosterom, E. J., Snow, V., Murphy, C., Moore, A. D., Brown, H., Whish, J. P., Verrall, S., Fainges, J., Bell, L. W., Peake, A. S., Poulton, P. L., Hochman, Z., Thorburn, P. J., Gaydon, D. S., Dalgliesh, N. P., Rodriguez, D., Cox, H., Chapman, S., Doherty, A., Teixeira, E., Sharp, J., Cichota, R., Vogeler, I., Li, F. Y., Wang, E., Hammer, G. L., Robertson, M. J., Dimes, J. P., Whitbread, A. M., Hunt, J., van Rees, H., McClelland, T., Carberry, P. S., Hargreaves, J. N., MacLeod, N., McDonald, C., Harsdorf, J., Wedgwood, S., and Keating, B. A. (2014). APSIM - Evolution towards a new generation of agricultural systems simulation. *Environmental Modelling & Software*, 62:327 – 350.

Horn, R., Fleige, H., and Peth, S. (2006). *Soils and Landuse Management Systems in Schleswig-Holstein (Germany): Guide of ISTRO Excursion 2006*. Schriftenreihe. Inst. für Pflanzenernährung und Bodenkunde.

Hudson, G. and Wackernagel, H. (1994). Mapping temperature using kriging with external drift: Theory and an example from scotland. *International Journal of Climatology*, 14(1):77–91.

Huerta-Espino, J., Singh, R. P., Germán, S., McCallum, B. D., Park, R. F., Chen, W. Q., Bhardwaj, S. C., and Goyeau, H. (2011). Global status of wheat leaf rust caused by puccinia triticina. *Euphytica*, 179(1):143–160.

Jamieson, P., Porter, J., and Wilson, D. (1991). A test of the computer simulation model ARCWHEAT1 on wheat crops grown in New Zealand. *Field Crops Research*, 27(4):337 – 350.

Jensen, A. L. and Jensen, F. V. (1996). Midas - an influence diagram for management of mildew in winter wheat. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, UAI'96, pages 349–356, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jones, J., Hoogenboom, G., Porter, C., Boote, K., Batchelor, W., Hunt, L., Wilkens, P., Singh, U., Gijsman, A., and Ritchie, J. (2003). The DSSAT cropping system model. *European Journal of Agronomy*, 18(34):235 – 265. Modelling Cropping Systems: Science, Software and Applications.

Käsbohrer, M., Hoffmann, G., and Fischbeck, G. (1988). Disease frequency as a threshold for control of powdery mildew (Erysiphe graminis f. sp. tritici) on wheat. *Journal of Plant Diseases and Protection*, 95(1):1–15.

Keating, B., Carberry, P., Hammer, G., Probert, M., Robertson, M., Holzworth, D., Huth, N., Hargreaves, J., Meinke, H., Hochman, Z., McLean, G., Verburg, K., Snow, V., Dimes, J., Silburn, M., Wang, E., Brown, S., Bristow, K., Asseng, S., Chapman, S., McCown, R., Freebairn, D., and Smith, C. (2003). An overview of APSIM, a model designed for farming systems simulation. *European Journal of Agronomy*, 18(34):267 – 288. Modelling Cropping Systems: Science, Software and Applications.

Khoshgoftaar, T. M., Fazelpour, A., Dittman, D. J., and Napolitano, A. (2015). Ensemble vs. data sampling: Which option is best suited to improve classification performance of imbalanced bioinformatics data? In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 705–712.

Kim, K. S., Beresford, R. M., and Walter, M. (2014). Development of a disease risk prediction model for downy mildew (peronospora sparsa) in boysenberry. *Phytopathology*, 104(1):50–56.

Klink, H. (1997). *Geoepidemiologische Erhebungen von Weizenpathogenen in Schleswig-Holstein unter Anwendung und Entwicklung des Integrierten Pflanzenschutzsystems (IPS-Modell Weizen) für einen minimierten, bedarfsgerechten Fungizideinsatz (1993- 1996).* PhD thesis, Christian-Albrechts-Universität zu Kiel.

Kocourek, F. and Vǎchet, L. (1984). Über ein temperaturabhängiges Modell zur Vorhersage der Entwicklungsgeschwindigkeit bei Erysiphe graminis f. sp.tritici. *Anzeiger für Schädlingskunde, Pflanzenschutz, Umweltschutz*, 57(1):15–18.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283.

Krige, D. G. (1951). *A Statistical Approach to Some Mine Valuation and Allied Problems on the Witwatersrand*.

Kruse, R., Borgelt, C., Braune, C., Klawonn, F., Moewes, C., and Steinbrecher, M. (2015). *Computational Intelligence*. Springer Vieweg, second edition.

Kuhn, M., Weston, S., Coulter, N., and Culp., M. (2015). *C50: C5.0 Decision Trees and Rule-Based Models*. R package version 0.1.0-24.

Langensiepen, M., Hanus, H., Schoop, P., and Gräsle, W. (2008). Validating CERES-wheat under North-German environmental conditions. *Agricultural Systems*, 97:34 – 47.

Langley, N. R., Dudzik, B., and Cloutier, A. (2017). A decision tree for nonmetric sex assessment from the skull. *Journal of Forensic Sciences*.

Langley, P. (1986). Editorial: On machine learning. *Machine Learning*, 1(1):5–10.

Lantz, B. (2015). *Machine Learning with R*. Packt Publishing Ltd., second edition.

Larose, D. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley.

Last, F. (1955). The spore content of air within and above mildew-infected cereal crops. *Transactions of the British Mycological Society*, 38(4):453 – 464.

Li, J. and Heap, A. D. (2014). Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, 53(0):173 – 189.

Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. *R News*, 2(3):18–22.

Littleboy, M., Silburn, D., Freebairn, D., Woodruff, D., Hammer, G., and Leslie, J. (1992). Impact of soil erosion on production in cropping systems. i. development and validation of a simulation model. *Soil Research*, 30(5):757–774.

Liu, N., Gong, G., Zhang, M., Zhou, Y., Chen, Z., Yang, J., Chen, H., Wang, X., Lei, Y., and Liu, K. (2012). Over-summering of wheat powdery mildew in sichuan province, china. *Crop Protection*, 34(0):112 – 118.

LKSH (2015). Sortenempfehlung Winterweizen 2015 Schleswig-Holstein Naturraum Östliches Hügelland. http://goo.gl/FHZfCu. Last accessed on March 5, 2018.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

Loh, W.-Y. (2014). Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348.

Löpmeier, F.-J. (2014). Agrarmeteorologisches Modell zur Berechnung der aktuellen Verdunstung (AMBAV). https://goo.gl/5rMMxQ. Last accessed on March 5, 2018.

Lu, J., Ehsani, R., Shi, Y., Abdulridha, J., de Castro, A. I., and Xu, Y. (2017). Field detection of anthracnose crown rot in strawberry using spectroscopy technology. *Computers and Electronics in Agriculture*, 135:289 – 299.

Luo, W., Taylor, M. C., and Parker, S. R. (2008). A comparison of spatial interpolation methods to estimate continuous wind speed surfaces using irregularly distributed data from england and wales. *International Journal of Climatology*, 28(7):947–959.

Madden, L. V. and Hughes, G. (1999). Sampling for plant disease incidence. *Phytopathology*, 89(11):1088–1103.

Matheron, G. (1963). Principles of geostatistics. *Economic geology*, 58(8):1246–1266.

Matheron, G. (1969). Le krigeage universel. *Cahiers du Centre de Morphologie Mathmatique, Ecole des Mines de Paris, Fontainebleau*, no 1.

McBratney, A. B., Odeh, I. O., Bishop, T. F., Dunbar, M. S., and Shatar, T. M. (2000). An overview of pedometric techniques for use in soil survey. *Geoderma*, 97(34):293 – 327.

McCown, R., Cox, P., Keating, B., Hammer, G., Carberry, P., Probert, M., and Freebairn, D. (1994). The development of strategies for improved agricultural systems and land-use management. In *Opportunities, use, and transfer of systems research methods in agriculture to developing countries*, pages 81–96. Springer.

McCown, R., Hammer, G., Hargreaves, J., Holzworth, D., and Freebairn, D. (1996). Apsim: a novel software system for model development, model testing and simulation in agricultural systems research. *Agricultural Systems*, 50(3):255 – 271.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mehra, L. K., Cowger, C., and Ojiambo, P. S. (2017). A model for predicting onset of stagonospora nodorum blotch in winter wheat based on preplanting and weather factors. *Phytopathology*, 107(6):635–644.

Merchán, V. and Kranz, J. (1986). Studies on the effect of rain on the infection of wheat by Erysiphe graminis DC. f. sp. tritici Marchal. *Journal of Plant Diseases and Protection*, 93(3):255–261.

Meynen, E. and Schmithüsen, J. (1962). *Handbuch der naturräumlichen Gliederung Deutschlands: 1953-1962*. Number Bd. 2. Bundesanst. für Landeskunde u. Raumforschung.

Miedaner, T. (2014). *Kulturpflanzen: Botanik - Geschichte - Perspektiven*. Springer Berlin Heidelberg.

Miedaner, T. and Flath, K. (2007). Effectiveness and environmental stability of quantitative powdery mildew (blumeria graminis) resistance among winter wheat cultivars. *Plant Breeding*, 126(6):553–558.

Monestiez, P., Courault, D., Allard, D., and Ruget, F. (2001). Spatial interpolation of air temperature using environmental context: Application to a crop model. *Environmental and Ecological Statistics*, 8(4):297–309.

Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 58(302):415–434.

Mwebaze, E. and Biehl, M. (2016). *Prototype-Based Classification for Image Analysis and Its Application to Crop Disease Diagnosis*, pages 329–339. Springer International Publishing, Cham.

Nguyen, X. T., Nguyen, B. T., Do, K. P., Bui, Q. H., Nguyen, T. N. T., Vuong, V. Q., and Le, T. H. (2015). Spatial interpolation of meteorologic variables in vietnam using the kriging method. *Journal of Information Processing Systems*, 11(1):134–147.

Nicodemus, K. K. (2011). Letter to the editor: On the stability and ranking of predictors from random forest variable importance measures. *Briefings in Bioinformatics*, 12(4):369.

Olatinwo, R. O., Paz, J. O., Brown, S. L., Kemerait, R. C., Culbreath, A. K., and Hoogenboom, G. (2009). Impact of early spring weather factors on the risk of tomato spotted wilt in peanut. *Plant Disease*, 93(8):783–788.

Patrick, E. and Fischer, F. (1970). A generalized k-nearest neighbor rule. *Information and Control*, 16(2):128 – 152.

Paul, P. A. and Munkvold, G. P. (2005). Regression and artificial neural network modeling for the prediction of gray leaf spot of maize. *Phytopathology*, 95(4):388–396.

Paulus, A. O. (1990). Fungal diseases of strawberry. *HortScience*, 25(8):885–889.

Piazza, A. D., Conti, F. L., Noto, L., Viola, F., and Loggia, G. L. (2011). Comparative analysis of different techniques for spatial interpolation of rainfall data to create a serially complete monthly time series of precipitation for sicily, italy. *International Journal of Applied Earth Observation and Geoinformation*, 13(3):396 – 408.

Pietzsch, S. and Bissolli, P. (2011). A modified drought index for wmo ra vi. *Advances in Science and Research*, 6(1):275–279.

Porter, J. (1984). A model of canopy development in winter wheat. *The Journal of Agricultural Science*, 102(02):383–392.

Porter, J. R. (1993). Afrcwheat2: a model of the growth and development of wheat incorporating responses to water and nitrogen. *European Journal of Agronomy*, 2(2):69–82.

Quinlan, J. R. et al. (1996). Bagging, boosting, and c4. 5. In *AAAI/IAAI, Vol. 1*, pages 725–730.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rabbinge, R., Jorritsma, I., and Schans, J. (1985). Damage components of powdery mildew in winter wheat. *Netherlands Journal of Plant Pathology*, 91(5):235–247.

Revolution Analytics and Weston, S. (2015a). *doSNOW: Foreach Parallel Adaptor for the 'snow' Package*. R package version 1.0.14.

Revolution Analytics and Weston, S. (2015b). *foreach: Provides Foreach Looping Construct for R*. R package version 1.4.3.

Rimbach, G., Nagursky, J., and Erbersdobler, H. (2015). *Lebensmittel-Warenkunde für Einsteiger*. Springer-Lehrbuch. Springer Berlin Heidelberg.

Rish, I. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York.

Ritchie, J., Godwin, D., and Otter-Nacke, S. (1988). CERES-Wheat. A simulation model of wheat growth and development. *Univ. of Tex. Press, Austin, Tex.*

Roelfs, A., Singh, R., Saari, E., Maize, I., and Center, W. I. (1992). *Rust Diseases of Wheat: Concepts and Methods of Disease Management*. Concepts and Methods of Disease Management. CIMMYT.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Reviews*, 65:386–408.

Roßberg, D., Jorg, E., and Falke, K. (2005). SIMONTO–ein neues Ontogenesemodell fur Wintergetreide und Winterraps. *Nachrichtenblatt des Deutschen Pflanzenschutzdienstes*, 57(4):74–80.

Rossi, V. and Giosuè, S. (2003). A dynamic simulation model for powdery mildew epidemics on winter wheat. *EPPO Bulletin*, 33(3):389–396.

Rossi, V., Racca, P., Giosue', S., Pancaldi, D., and Alberti, I. (1997). A simulation model for the development of brown rust epidemics in winter wheat. *European Journal of Plant Pathology*, 103(5):453–465.

Sache, I. (2000). Short-distance dispersal of wheat rust spores. *Agronomie*, 20(7):757–767.

Sankaran, S. and Ehsani, R. (2013). Detection of huanglongbing-infected citrus leaves using statistical models with a fluorescence sensor. *Applied spectroscopy*, 67(4):463–469.

Sankaran, S., Ehsani, R., Inch, S. A., and Ploetz, R. C. (2012). Evaluation of visible-near infrared reflectance spectra of avocado leaves as a non-destructive sensing tool for detection of laurel wilt. *Plant Disease*, 96(11):1683–1689.

Schaeben, H., Akin, H., and Siemes, H. (2013). *Praktische Geostatistik: Eine Einführung für den Bergbau und die Geowissenschaften*. Hochschultext. Springer Berlin Heidelberg.

Schapire, R. and Freund, Y. (2012). *Boosting: Foundations and Algorithms*. Adaptive computation and machine learning. MIT Press.

Schlenger, H., Paffen, K., and Stewig, R. (1969). *Schleswig-Holstein: ein geographisch-landeskundlicher Exkursionführer*. Schriften des Geographischen Instituts der Universität Kiel. Hirt.

Schlüter, K., Börner, H., and Aumann, J. (2009). *Pflanzenkrankheiten und Pflanzenschutz*. Springer-Lehrbuch. Springer Berlin Heidelberg.

Schmidt, J. (2010). Weltbank warnt vor Ernährungskrise. Süddeutsche Zeitung - https://goo.gl/3BJiWf. Last accessed on March 5, 2018.

Secher, B. J. M., Jørgensen, L. N., Murali, N. S., and Boll, P. S. (1995). Field validation of a decision support system for the control of pests and diseases in cereals in denmark. *Pesticide Science*, 45(2):195–199.

Shah, D. A., De Wolf, E. D., Paul, P. A., and Madden, L. V. (2014). Predicting fusarium head blight epidemics with boosted regression trees. *Phytopathology*, 104(7):702–714.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, pages 517–524, New York, NY, USA. ACM.

Simkin, M. B. and Wheeler, B. E. J. (1974). The development of Puccinia hordei on barley cv. Zephyr. *Annals of Applied Biology*, 78(3):225–235.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). ROCR: visualizing classifier performance in R. *Bioinformatics*, 21(20):7881.

Sinha, P., Prajneshui, and Varma, A. (2004). Statistical Modelling and Forecasting of Powdery Mildews Affecting Agricultural Crops: An Overview . *Journal of the Indian Society of Agricultural Statistics*, 57.

Smith, D. L., Hollowell, J. E., Isleib, T. G., and Shew, B. B. (2007). A site-specific, weather-based disease regression model for sclerotinia blight of peanut. *Plant Disease*, 91(11):1436–1444.

Soltani, A. and Sinclair, T. (2012). *Modeling Physiology of Crop Development, Growth and Yield*. CAB books. CABI.

Sreeramulu, T. (1964). Incidence of conidia of erysiphe graminis in the air over a mildew-infected barley field. *Transactions of the British Mycological Society*, 47(1):31 – 38.

Statistik-Nord (2017a). *Die Bodennutzung in Schleswig-Holstein 2017*. Statistische Berichte. Statistisches Amt für Hamburg und Schleswig-Holstein Hamburg.

Statistik-Nord (2017b). *Ernteberichterstattung über Feldfrüchte und Grünland in Schleswig-Holstein September 2017*. Statistische Berichte. Statistisches Amt für Hamburg und Schleswig-Holstein Hamburg.

Steger, S., Brenning, A., Bell, R., Petschko, H., and Glade, T. (2016). Exploring discrepancies between quantitative validation results and the geomorphic plausibility of statistical landslide susceptibility maps. *Geomorphology*, 262:8 – 23.

Sutton, R. S. (1992). Introduction: The challenge of reinforcement learning. *Machine Learning*, 8(3-4):225–227.

Thépot, S. and Gouache, D. (2009). Étude d'un modéle climatique de prévision de la nuisibilité de la rouille brune sur blé dur en France. AFPP 9 Conférence internationale sur les maladies des plantes.

Thiessen, A. H. (1911). Precipitation averages for large areas. *Monthly weather review*, 39(7):1082–1089.

Timm, B. C. and McGarigal, K. (2012). Fine-scale remotely-sensed cover mapping of coastal dune and salt marsh ecosystems at cape cod national seashore using random forests. *Remote Sensing of Environment*, 127:106–117.

Vechet, L., Burketova, L., and Sindelarova, M. (2009). A comparative study of the efficiency of several sources of induced resistance to powdery mildew (blumeria graminis f. sp. tritici) in wheat under field conditions. *Crop Protection*, 28(2):151 – 154.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Verreet, J. A., Klink, H., and Hoffmann, G. M. (2000). Regional Monitoring for Disease Prediction and Optimization of Plant Protection Measuares: The IPM Wheat Model. *Plant Disease*, 84(8):816–826.

Wagner, P. D., Fiener, P., Wilken, F., Kumar, S., and Schneider, K. (2012). Comparison and evaluation of spatial interpolation schemes for daily rainfall in data scarce regions. *Journal of Hydrology*, 464465:388 – 400.

Webster, R. (1994). The development of pedometrics. *Geoderma*, 62(1):1 – 15.

Weir, A., Bragg, P., Porter, J., and Rayner, J. (1984). A winter wheat crop simulation model without water or nutrient limitations. *The Journal of Agricultural Science*, 102(02):371–382.

Wen, L., Bowen, C. R., and Hartman, G. L. (2017). Prediction of short-distance aerial movement of phakopsora pachyrhizi urediniospores using machine learning. *Phytopathology*, 107(10):1187–1198.

Wernecke, P. and Claus, S. (1996). Modelle der Ontogenese für die Kulturarten Winterweizen, Wintergerste und Winterraps. In Mühle, H. and Claus, S., editors, *Reaktionsverhalten von agrarischen Ökosystemen homogener Areale: Methoden der Beschreibung, Messung und Quantifizierung*, chapter 2.5, pages 105–120. Vieweg+Teubner Verlag.

Willocquet, L., Aubertot, J., Lebard, S., Robert, C., Lannou, C., and Savary, S. (2008). Simulating multiple pest damage in varying winter wheat production situations. *Field Crops Research*, 107(1):12 – 28.

Wolf, D., D., E., and Francl, L. J. (2000). Neural network classification of tan spot and stagonospora blotch infection periods in a wheat field environment. *Phytopathology*, 90(2):108–113.

Wooldridge, J. (2015). *Introductory Econometrics: A Modern Approach*. Cengage Learning.

Wyner, A. J., Olson, M., Bleich, J., and Mease, D. (2017). Explaining the success of adaboost and random forests as interpolating classifiers. *Journal of Machine Learning Research*, 18(48):1–33.

Yarwood, C. (1957). Powdery mildews. *The Botanical Review*, 23(4):235–301.

Zhang, J., Pu, R., Yuan, L., Huang, W., Nie, C., and Yang, G. (2014). Integrating remotely sensed and meteorological observations to forecast wheat powdery mildew at a regional scale. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(11):4328–4339.

Zhang, P. (1993). Model selection via multifold cross validation. *The Annals of Statistics*, 21(1):299–313.

Zhang, S., Shang, Y., Wang, L., et al. (2015). Plant disease recognition based on plant leaf image. *Journal of Animal & Plant Sciences*, 25(3):42–45.

Appendix A

Additional figures

Appendix A contains additional figures representing the infestation data collected by IPS monitoring (Verreet et al., 2000).

Figures A.1 and A.2 display the proportion of observed exceedances of the disease incidence threshold by Klink (1997) for powdery mildew and brown rust depending on the pathogens susceptibility classes.

Figure A.3 and A.4 present the locations where IPS sampling was carried out in which years and the proportion of exceedances of the damage threshold for the wheat variety Ritmo.

Figures A.5 and A.6 show the average spatial distribution of powdery mildew and brown rust in Schleswig-Holstein over the season.

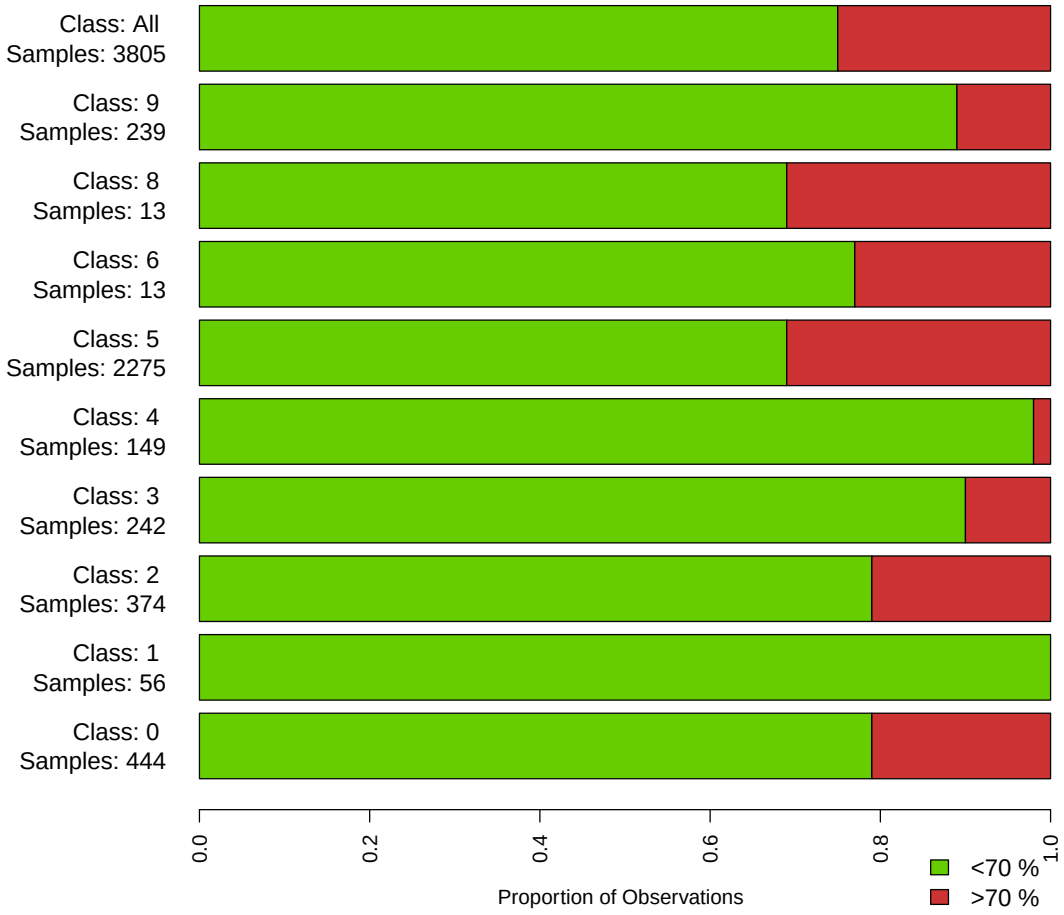


Figure A.1: Summary of observed disease incidence exceedances of the 70 % threshold for all available powdery mildew susceptibility classes. The 0 class implies no known vulnerability for the species

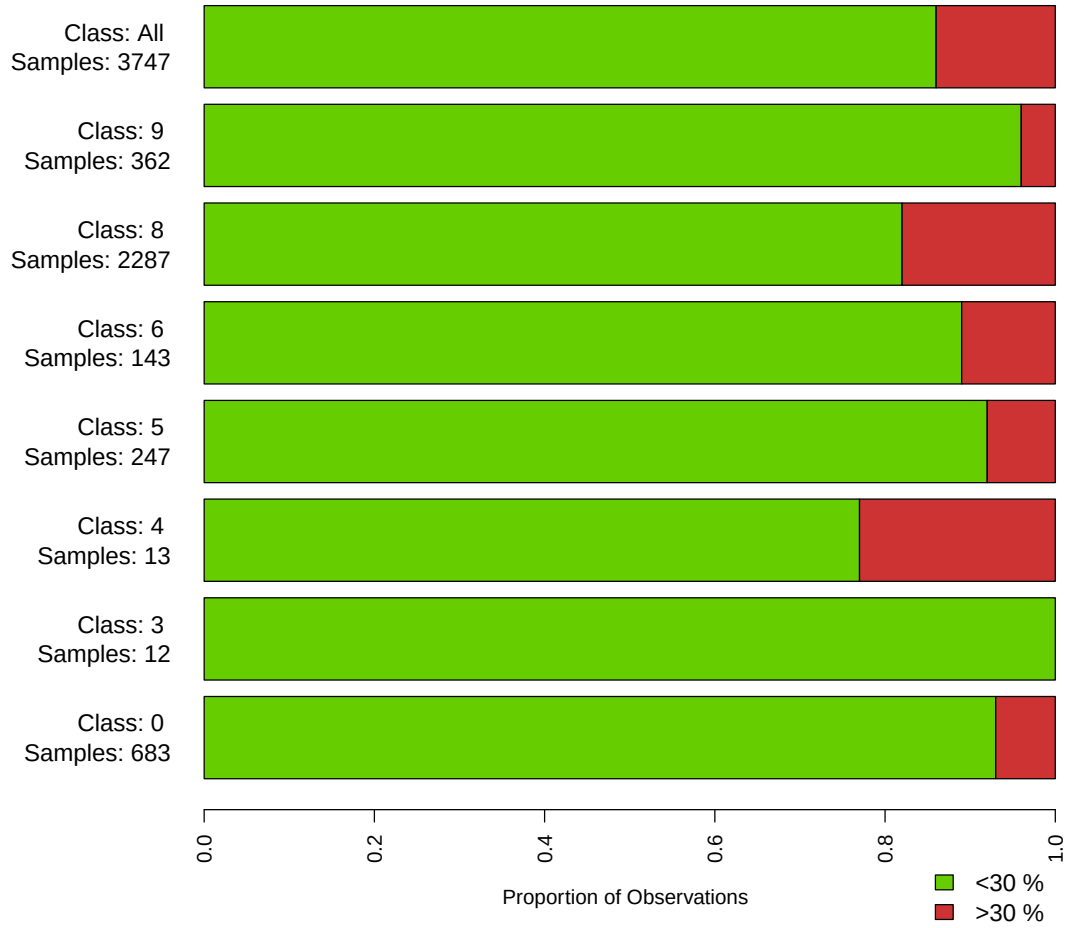


Figure A.2: Summary of observed disease incidence exceedances of the 30 % threshold for all available brown rust susceptibility classes. The 0 class implies no known vulnerability for the species

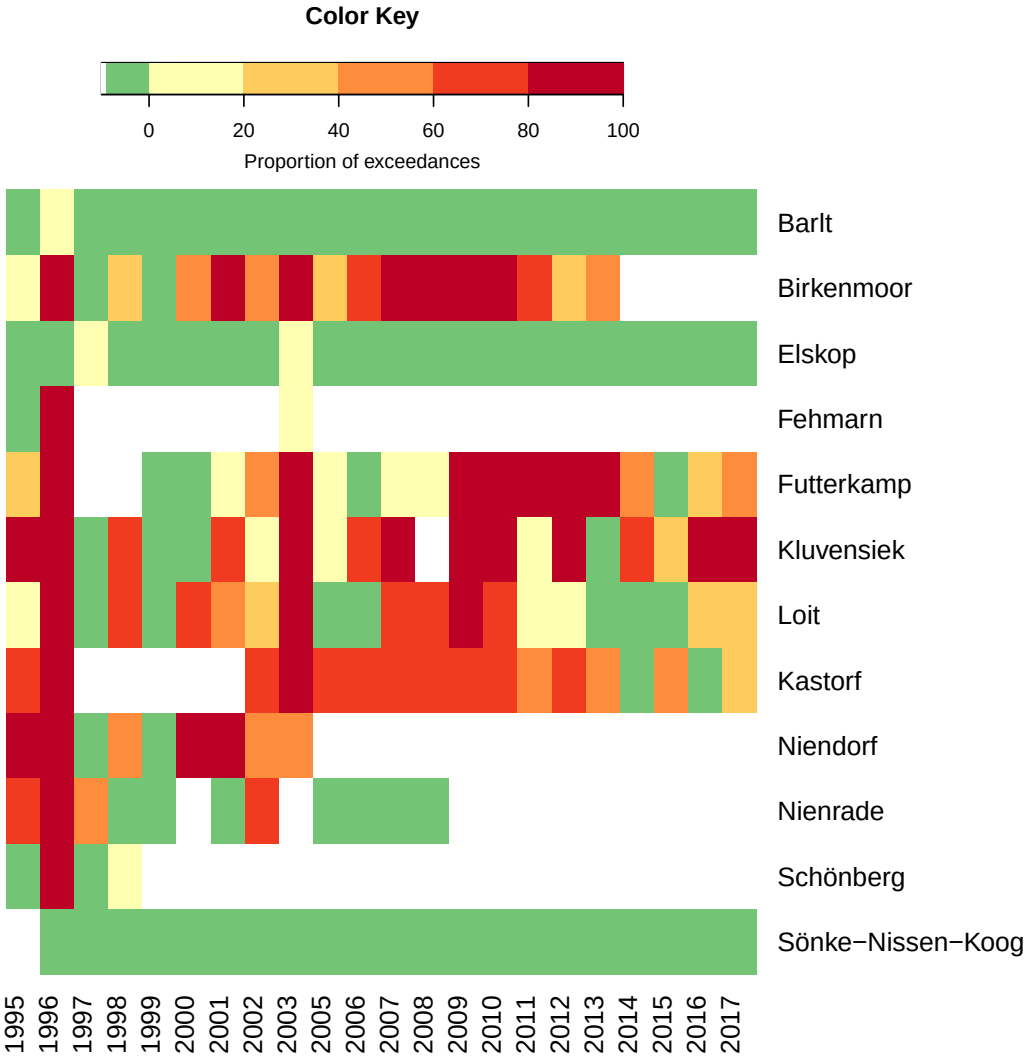


Figure A.3: Proportion of observed disease incidence exceedances of the 70 % threshold of all powdery mildew observations on Ritmo in Schleswig-Holstein. White spaces imply no measurements at that location in the year.

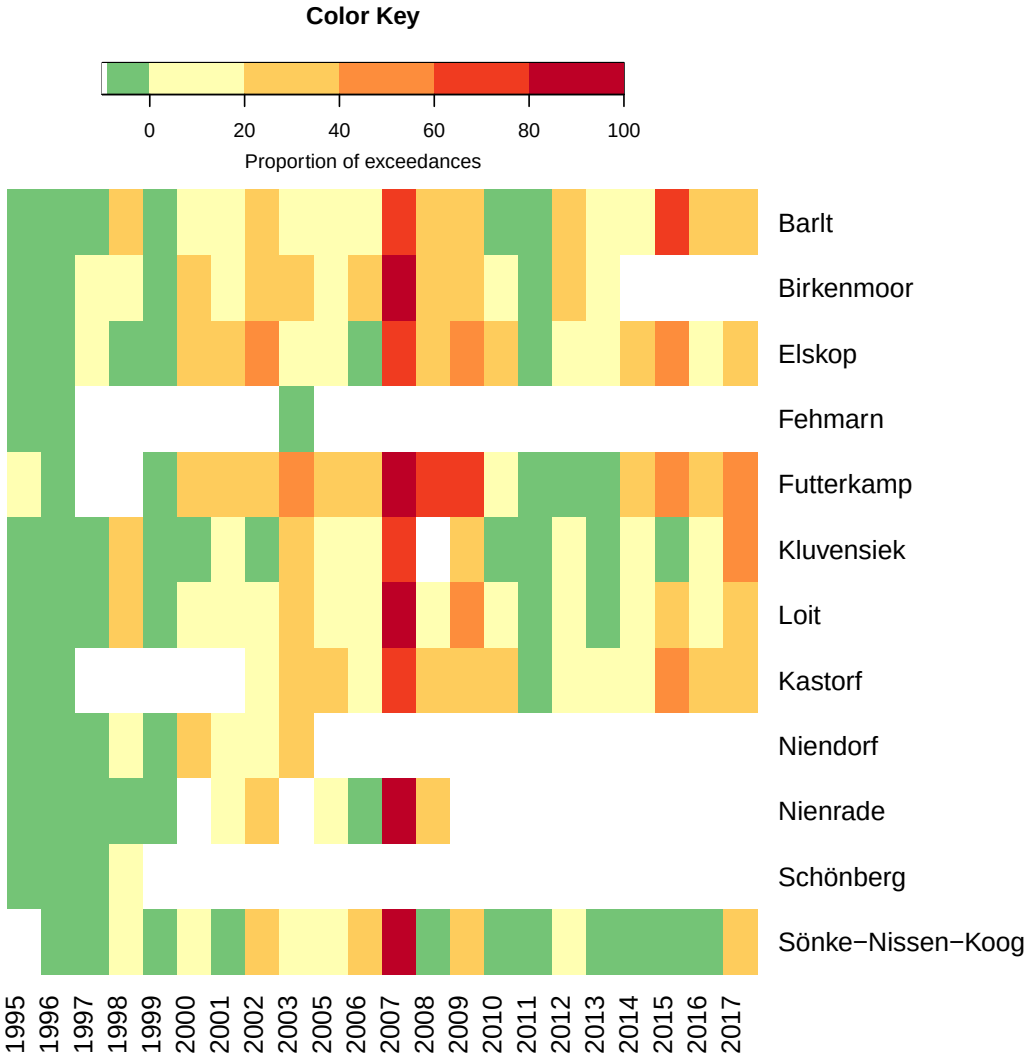


Figure A.4: Proportion of observed disease incidence exceedances of the 30 % threshold of all brown rust observations on Ritmo in Schleswig-Holstein. White spaces imply no measurements at that location in the year.

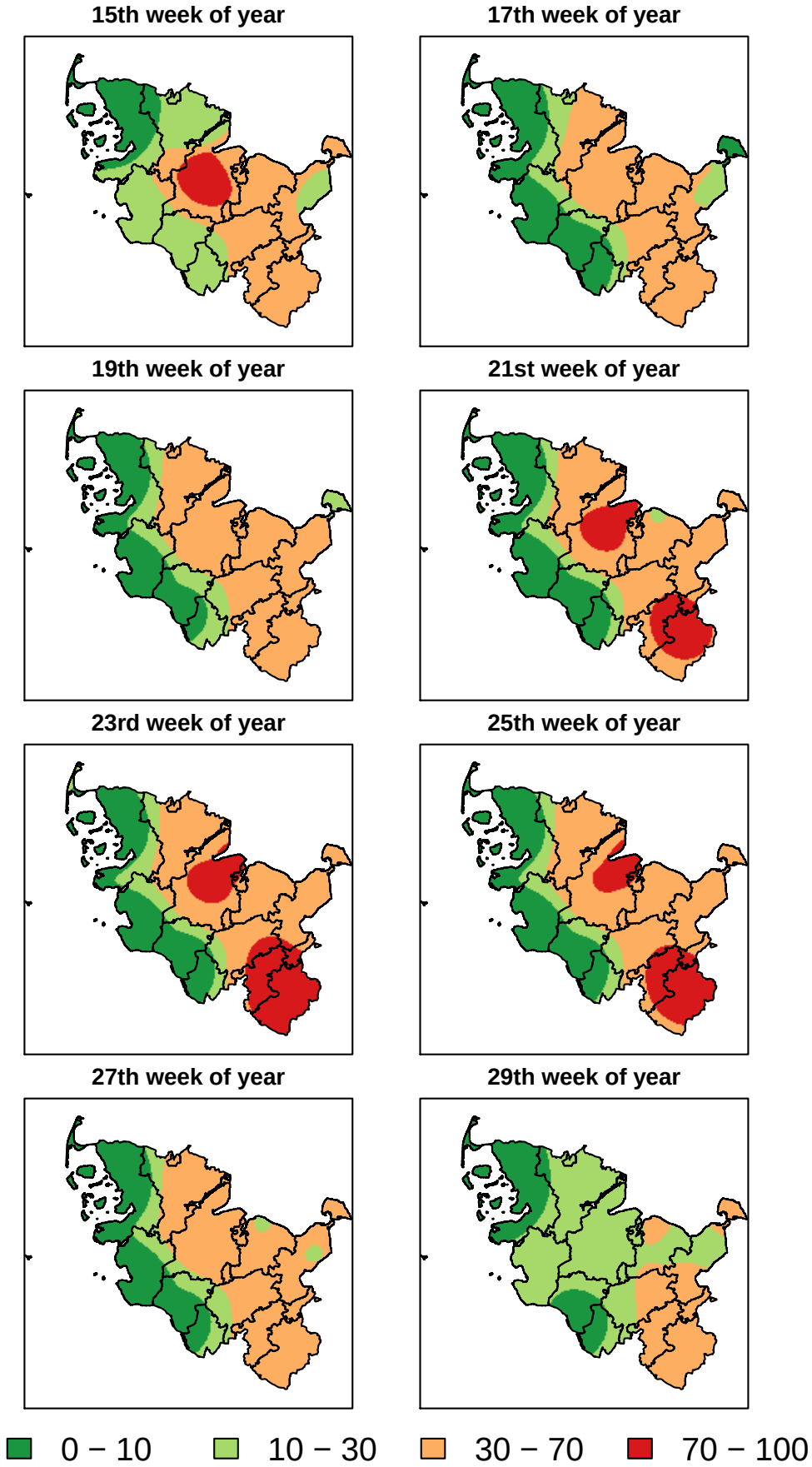


Figure A.5: Interpolated averaged observed disease incidences of powdery mildew in Schleswig-Holstein

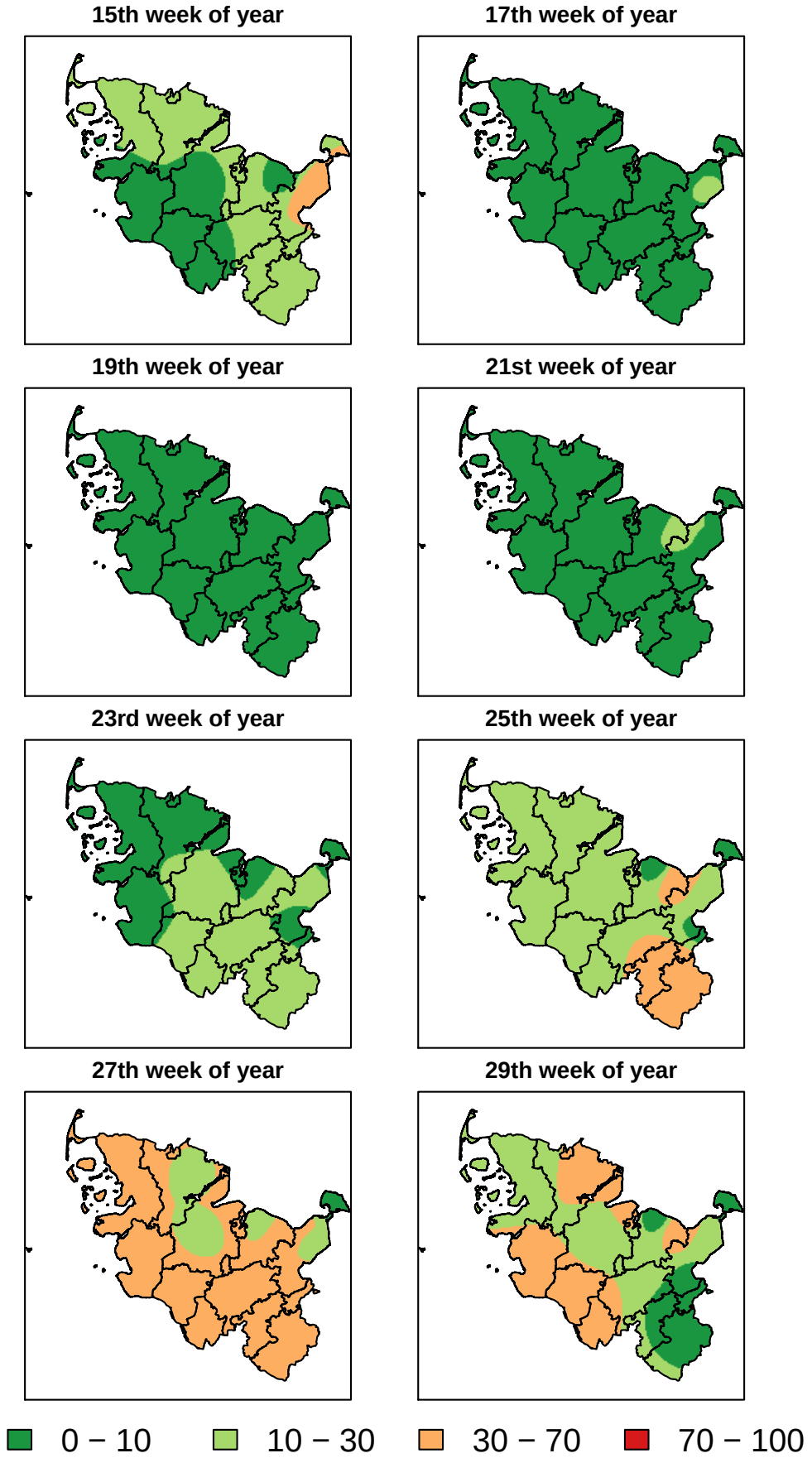


Figure A.6: Interpolated averaged observed disease incidences of brown rust in Schleswig-Holstein

Appendix B

Code

Appendix B contains the R-code of the thesis. It is structured:

B.1 Additional functions

```
# This script holds some useful functions
# Created 2017-01-13
# Created by: Wolfgang B. Hamer

##########################################
# Calculation of the logarithmic wind function
logwi=function(windkl,mesh){
k=0.41                          #constant
z=mesh                          #elevation (recent)
zo=2*10^-2                      #roughness length of wheat
d=0                             #suppression height
u=(windkl*k)/(log((z-d)/zo))#shear stress speed
z=2                             #elevation (aim)
wind=(u/k)*log((z-d)/zo)        # Final Windspeed
return(wind)}

##########################################
# Creation of Voronoi Polygons (created by Carson Farmer)
# http://www.carsonfarmer.com/2009/09/voronoi-polygons-with
  -r/
voronoipolygons=function(x,poly) {
  require(deldir)
  if (.hasSlot(x, 'coords')) {
    crds=x@coords
  } else crds=x
  bb = bbox(poly)
  rw = as.numeric(t(bbox(poly)))
  z=deldir(crds[,1], crds[,2],rw=rw)
  w=tile.list(z)
  polys=vector(mode='list', length=length(w))
  require(sp)
  for (i in seq(along=polys)) {
    pcrds=cbind(w[[i]]$x, w[[i]]$y)
    pcrds=rbind(pcrds, pcrds[1,])
    polys[[i]]=Polygons(list(Polygon(pcrds)), ID=as.
       character(i))
  }
  SP=SpatialPolygons(polys)
  voronoi=SpatialPolygonsDataFrame(SP, data=data.frame(x=
```

```
      crds[,1],
  y=crds[,2], row.names=sapply(slot(SP, 'polygons'),
  function(x) slot(x, 'ID'))))
  return(voronoi)
}

##############################################
# Create and rasterize a voronoi polygon of the points "
   jewe" with the values of "gevari" and the mask of "
   shmapo"
vop=function(jewe,gevari,shmapo){
tempint=jewe
# The poygon is created and the coordinate system is set
vo=voronoipolygons(tempint,shmapo)
vo@proj4string=CRS("+init=epsg:25832")

# The voronoi polygons (row "werte") get the information ("
   gevari") of the shape ("jewe")
tab=over(vo,tempint)
vo$namen=tab
vo$werte=tempint[paste(gevari)]@data[,1]

# The polygon gets rasterized with a specific extent of
   1000 meters
vora=raster(extent(vo))
res(vora)=1000
vora=rasterize(vo, vora,"werte")
return(vora)}

##############################################
# Create a function to calculate the modal value of a
   raster stack
modus=function(stac){
Modu=function(x){
y=unique(x)
y=y[!is.na(y)]
y[which.max(tabulate(match(x, y)))]
}
return(calc(stac, fun=Modu))
}
```

```
##########################################
# Create a function to calculate the temperature sum
# Created after Soltani p. 60 ff
tempfun=function(TMP,TBD=0,TP1D=25,TP2D=28,TCD=40){
#TBD=0 # base temperature
#TP1D=25 # lower optimum temperature
#TP2D=28 # upper optimum temperature
#TCD=40 #ceiling temperature
tempfun=TMP
tempfun[TMP<=TBD]=0
tempfun[TMP>TBD&TMP<TP1D]=(TMP[TMP>TBD&TMP<TP1D]-TBD)/(TP1D
    -TBD)
tempfun[TMP>=TP1D&TMP<=TP2D]=1
tempfun[TMP>TP2D&TMP<TCD]=(TCD-TMP[TMP>TP2D&TMP<TCD])/(TCD-
    TP2D)
tempfun[TMP>=TCD]=0
return(tempfun)}

##############################################
# Create a function to calculate daily thermal unit using
    the temperature sum
dtu=function(TMP,TBD=0,TP1D=25,TP2D=28,TCD=40){
DTU=(TMP-TBD)*tempfun(TMP)
return(DTU)}

##############################################
# Creating a function to test different k-values for the
    kNN-prediction
kv_fit=function(dataset_kn,maxk=50){
   dat_years=sort(as.numeric(as.character(unique(years(
       dataset_kn$Datum)))))
   trainyears=sort(sample(dat_years,length(dat_years)/2))
   testyears=dat_years[!is.element(dat_years,trainyears)]

   n_dataset=as.data.frame(lapply(dataset_kn[3:(dim(dataset_
       kn)[2]-2)], normalize))
   for(xy in 1:(dim(n_dataset)[2])){
    if(any(is.nan(n_dataset[,xy]))){n_dataset[,xy]=rep(0,
        length(n_dataset[,xy]))}
   }
   knn_train=n_dataset[is.element(years(dataset_kn$Datum),
```

```
    trainyears), ]
  knn_test=n_dataset[is.element(years(dataset_kn$Datum),
    testyears), ]

  # Selecting the classified disease incidence values as
    labels
  knn_train_labels=dataset_kn[is.element(years(dataset_kn$
    Datum),trainyears),(dim(dataset_kn)[2])]
  knn_test_labels=dataset_kn[is.element(years(dataset_kn$
    Datum),testyears),(dim(dataset_kn)[2])]

  # What if the labels are not different?
  if(sum(length(unique((knn_train_labels))),length(unique((
    knn_test_labels))))!=4){
   print("Not enough cases! Sqrt selected")
   return(round(sqrt(dim(dataset_kn)[1]),0))
  }else{
  # Defining an variable representing the worst fit
  maxv=c(0,0)

  # Trying out the values 1-maxk as k-values and
    calculating the Area under the ROC as result
  for(i in 1:maxk){
   tempres=data.frame(Observed=as.factor(knn_test_labels),
     Predicted=knn(train = knn_train, test = knn_test, cl
     = knn_train_labels, k = i))

   #Evaluating the Area under the ROC curve.
   kn_pre=prediction(predictions=as.numeric(tempres$
     Predicted),labels=tempres$Observed)
   kn_perfa=performance(kn_pre,measure="auc")
   kn_auc=unlist(kn_perfa@y.values)

   # If the actual area under the roc is larger than the
     best fit before the variable maxv should be
     overwritten with the used k value and the new auc
   if(kn_auc>maxv[2]){
    maxv=c(i,kn_auc)
   }
   }
 return(maxv[1])
```

```
 }
}

##########################################
# Creating a function to evaluate the best trials of
   Decision trees based on the Area under the ROC
evtrials=function(evdataset){
  # Separating in training and test years
  dat_years=sort(as.numeric(as.character(unique(years(
     evdataset$Datum)))))
  trainyears=sort(sample(dat_years,length(dat_years)/2))
  testyears=dat_years[!is.element(dat_years,trainyears)]
  traindat=evdataset[is.element(as.numeric(as.character(
     years(evdataset$Datum))),trainyears),]
  testdat=evdataset[is.element(as.numeric(as.character(
     years(evdataset$Datum))),testyears),]

  # If there are not at last two resulting classes do not
     weight
  if(length(unique(as.factor(traindat[,(dim(traindat)[2])]
     ))<2|length(unique(as.factor(testdat[,(dim(testdat)
     [2])])))<2){
  print("Not enough cases! No trials used")
  return(1)
  }else{

  maxv=c(0,0)

  for(o in c(1, 5, 10, 15, 20, 25, 30, 35)){
   model=C5.0(traindat[,c(3:(dim(traindat)[2]-2))], as.
      factor(traindat[,(dim(traindat)[2])]),control = C5.0
      Control(minCases =round((dim(traindat)[1]/100)*.7,0))
      ,trials=o)
   tempres=data.frame(Observed=as.factor(testdat$IBHBF),
      Predicted=predict(model, testdat))

   # Evaluating the Area under the ROC curve.
   dt_pre=prediction(predictions=as.numeric(tempres$
      Predicted),labels=tempres$Observed)
   dt_perfa=performance(dt_pre,measure="auc")
   dt_auc=unlist(dt_perfa@y.values)
```

```
    if(dt_auc>maxv[2]){
     maxv=c(o,dt_auc)
    }
   }
   return(maxv[1])
   }
}

##########################################
# Creating a function to evaluate the best classwt weight
   and the best mtry value of the aim variable for the
   Random Forests based on the Area under the ROC
evclassmtry=function(evdataset){
  # Separating in training and test years
  dat_years=sort(as.numeric(as.character(unique(years(
     evdataset$Datum)))))
  trainyears=sort(sample(dat_years,length(dat_years)/2))
  testyears=dat_years[!is.element(dat_years,trainyears)]
  traindat=evdataset[is.element(as.numeric(as.character(
     years(evdataset$Datum))),trainyears),]
  testdat=evdataset[is.element(as.numeric(as.character(
     years(evdataset$Datum))),testyears),]

  #Creating a standard mtry value
  mtry1=round(dim((traindat[,c(3:(dim(traindat)[2]-2))]))
     [2]/3,0)

  # If there are not at last two resulting classes do not
     weight
  if(length(unique(as.factor(traindat[,(dim(traindat)[2])])
     ))<2|length(unique(as.factor(testdat[,(dim(testdat)
     [2])])))<2){
  print("Not enough cases! No weighting used")
  return(c(mtry1,1))
  }else{

  # If the tuneRF function doesnt work proper
  vers=evalWithTimeout({try(randomForest(as.formula(paste("
     as.factor(",names(traindat)[(dim(traindat)[2])],")~",
     paste(names(traindat)[c(3:(dim(traindat)[2]-2))],
```

```
      collapse="+"))),
      data=traindat, importance=TRUE, ntree=1000), silent=
         TRUE)},timeout=600,onTimeout="warning")
   if (any(class(vers) == "try-error")){
   return(c(mtry1,1))
   }else{

   maxv=c(mtry1,1,0)

   for(j in c(mtry1,2,4,8,16)){
    for(o in seq(1,2E6,.25E6)){
     tryrf=evalWithTimeout({try(randomForest(as.formula(
        paste("as.factor(",names(traindat)[(dim(traindat)
        [2])],")~",paste(names(traindat)[c(3:(dim(traindat)
        [2]-2))],collapse="+"))),
      data=traindat, importance=TRUE, ntree=1000,mtry=j,
         classwt = c(1,o)), silent=TRUE)},timeout=6000,
         onTimeout="warning")
     if (any(class(tryrf) == "try-error")){
      rf_auc=0
     }else{
      tempres=data.frame(Observed=as.factor(testdat$IBHBF),
         Predicted=predict(tryrf, testdat))

      # Evaluating the Area under the ROC curve.
      rf_pre=prediction(predictions=as.numeric(tempres$
         Predicted),labels=tempres$Observed)
      rf_perfa=performance(rf_pre,measure="auc")
      rf_auc=unlist(rf_perfa@y.values)
     }
     if(rf_auc>maxv[3]){
      maxv=c(j,o,rf_auc)
     }
    }
   }
   return(maxv[c(1,2)])
   }
   }
}
```

B.2 Download of weather data

Data used First, a brief overview of the data used in this script is given:

The file *"Beschreibung_Stationen.csv"* is a table, containing information about all weather observation stations of the DWD in Germany. It is used since the data of the CDC are stored according to the station ID. Using the table the data of NG can be separated.

Table B.1: Beschreibung_Stationen.csv

Stations_id	von_datum	bis_datum	Stationshoehe	geoBreite	geoLaenge	Stationsname	Bundesland
3	19500401	20110331	202	50.7827	6.0941	Aachen	Nordrhein-Westfalen
44	20070401	20160112	44	52.9335	8.2370	Groenkneten	Niedersachsen
52	19760101	19880101	46	53.6623	10.1990	Ahrensburg-Wulfsdorf	Schleswig-Holstein
71	20091201	20160112	759	48.2155	8.9784	Albstadt-Badkap	Baden-Württemberg
73	20070401	20160112	340	48.6159	13.0506	Aldersbach-Kriestorf	Bayern
78	20041101	20160112	65	52.4850	7.9119	Alfhausen	Niedersachsen

```
# This script aims at the automated download of the recent
    weather data of the German Meteorological Survey  and
    the combination with the weather data downloaded before
# Created 2016-06-12
# Created by: Wolfgang B. Hamer

# First the required packages need to be installed and the
    working directory needs to be defined
library(chron) # Manage time
library(RCurl) # Allow general HTTP requests
setwd("D:\\Doktorarbeit\\R_Scripte\\PaPros")

# Then a table is imported and formated with the
    information of all locations of the DWD
# Later one it will be used to filter the locations of
    northern germany
allstar=read.csv2("1. FTP Download\\Beschreibung_Stationen.
    csv")
for(k in 1:dim(allstar)[1]){allstar$Statnam[k]=strsplit(as.
    character(allstar$Stationsname[k]), " ")[[1]][2]}
for(k in 1:dim(allstar)[1]){allstar$Buland[k]=strsplit(as.
    character(allstar$Bundesland[k]), " ")[[1]][1]}
allstar=allstar[,-c(7,8)]
allstar$geoBreite=as.numeric(as.character(allstar$geoBreite
    ))
allstar$geoLaenge=as.numeric(as.character(allstar$geoLaenge
    ))
```

```
# Here the federal states of northern germany are filtered
    from the table
gestar=allstar[is.element(allstar$Buland, c("Niedersachsen"
    ,"Schleswig-Holstein","Mecklenburg-Vorpommern","Bremen",
    "Hamburg")),]

##################################################################
# Following the temperature and humidity are downloaded #
##################################################################

# First an empty dataframe is created; later one it will be
     filled with the weather data
temper=as.data.frame(matrix(ncol=9,nrow=1,dimnames=list(1,c
    ("Station","STATIONS_ID","Latitude","Longitude","Zeit","
    Datum","Stunde","Temperatur","Luftfeuchte"))))

# The following part looks into the folder of the
    meteorological surveys recent air temperature ftp server
     and sorts the information
# In the end it receives a vector (star) with the
    information of the available observation stations in
    form of the available zip files
filenames=getURL("ftp://ftp-cdc.dwd.de/pub/CDC/observations
    _germany/climate/hourly/air_temperature/recent/",ftp.use
    .epsv = FALSE,dirlistonly = TRUE)
filenames=gsub("zip","zipqwert", filenames)
filenames=strsplit(filenames, "qwert")
filenames=substring(filenames[[1]],3)
filenames[1]=paste("st",filenames[1],sep="")
star=filenames[which(nchar(filenames)==29)]

# Here the for-loop begins which includes the download and
    structuration of the temperature and humidity data
pb=txtProgressBar(min = 1, max = length(star), style = 3)
for(j in 1:length(star)){

# Only go on when the location code (as part of the
    zipfiles name) is equal to one of the locations in
    northern germany
if(is.element(as.numeric(strsplit(as.character(star[j]), "_
```

```
    ")[[1]][3]),gestar$Stations_id)){

# Download the location specific zipfile and store it as
    temporal file
pfad=paste("ftp://ftp-cdc.dwd.de/pub/CDC/observations_
    germany/climate/hourly/air_temperature/recent/",star[j],
    sep="")
download.file(pfad,"1. FTP Download\\tempdat.zip")

# Look into the zipfile and import the table containing "
    product_" in its name
# This table contains the weather information
statdat=read.csv2(unz("1. FTP Download\\tempdat.zip", unzip
    ("1. FTP Download\\tempdat.zip", list = T)$Name[grep("
    produkt_",unzip("1. FTP Download\\tempdat.zip", list = T
    )$Name)]))
# Look into the zipfile and import the table containing "
    Geographie" in its name
# This table contains the geographic information about the
    weather observation station
statinf=read.csv2(unz("1. FTP Download\\tempdat.zip",unzip(
    "1. FTP Download\\tempdat.zip", list = T)$Name[grep("
    Geographie",unzip("1. FTP Download\\tempdat.zip", list =
     T)$Name)]))

# The table "statinf" contains the information about the
    locations geographic location in different times
# Since the location is still active the information till
    when it is active is filled with the actual date
statinf$bis_datum[is.na(statinf$bis_datum)]=as.numeric(
    paste(substr(Sys.Date(),1,4),substr(Sys.Date(),6,7),
    substr(Sys.Date(),9,10),sep=""))
statinf$Geogr.Breite=as.numeric(as.character(statinf$Geogr.
    Breite))
statinf$Geogr.Laenge=as.numeric(as.character(statinf$Geogr.
    Laenge))

# The table containing the weather information gets the
    geographic location information and the name of the
    location
statdat$Latitude=NA
```

```
statdat$Longitude=NA
statdat$Station=paste(strsplit(as.character(statinf$
    Stationsname[1]), " ")[[1]][nchar(strsplit(as.character(
    statinf$Stationsname[1]), " ")[[1]])>0], collapse=" ")
# Since the location could change over the time, this must
    be considered giving the weather table the geographic
    location information
for(i in 1:dim(statinf)[1]){
statdat$Latitude[which(statdat$MESS_DATUM>=statinf$von_
    datum[i]*100 & statdat$MESS_DATUM<=statinf$bis_datum[i]*
    100)]=statinf$Geogr.Breite[i]
statdat$Longitude[which(statdat$MESS_DATUM>=statinf$von_
    datum[i]*100 & statdat$MESS_DATUM<=statinf$bis_datum[i]*
    100)]=statinf$Geogr.Laenge[i]
}

# The date and time information of the table is transformed
     to characters in form of the "chron" library
statdat$Zeit=as.character(chron(dates=paste(substr(statdat$
    MESS_DATUM,1,4),"-",substr(statdat$MESS_DATUM,5,6),"-",
    substr(statdat$MESS_DATUM,7,8),sep=""),times=paste(
    substr(statdat$MESS_DATUM,9,10),":00:00",sep=""),format=
    c('Y-m-d','h:m:s')))
if(dim(statdat[is.na(statdat$Zeit),])[1]>1){print("WARNUNG:
    Zu viele NAs")}else{statdat=statdat[!is.na(statdat$Zeit
    ),]}
statdat$LUFTTEMPERATUR=as.numeric(as.character(statdat$TT_
    TU))
statdat$REL_FEUCHTE=as.numeric(as.character(statdat$RF_TU))
statdat$Datum=as.Date(substr(statdat$MESS_DATUM,1,8),"%Y%m%
    d")
statdat$Stunde=as.numeric(substr(statdat$MESS_DATUM,9,10))

# The final dataset is extracted and renamed; only data
    newer than 2015 are considered further on, since older
    data are already stored locally
ausgabe=statdat[,c("Station","STATIONS_ID","Latitude","
    Longitude","Zeit","Datum","Stunde","LUFTTEMPERATUR","REL
    _FEUCHTE")]
names(ausgabe)[c(8,9)]=c("Temperatur","Luftfeuchte")
ausgabe=ausgabe[which(as.integer(as.character(years(ausgabe
```

```
    $Datum)))>2015),]

# If the created output is not empty it is combined with
    the information of the other weather measuring stations
if(dim(ausgabe)[1]>1){
 temper=rbind(temper,ausgabe)
 }
}
setTxtProgressBar(pb, j)
}
close(pb)

# All information are combined; now empty rows are deleted
    and data are brought to the correct format
temper=temper[-1,]
temper$Zeit=chron(dates=substr(temper$Zeit,2,9),times=paste
    (substr(temper$Zeit,11,12),":00:00",sep=""),format=c('Y-
    m-d','h:m:s'))
temper$STATIONS_ID=as.numeric(temper$STATIONS_ID)
temper=temper[!is.na(temper$STATIONS_ID),]

# Should some temperature and humidity data have no
    coordinates assigned, they get the coordinate of the
    actual location table with the fitting location
fixvar=unique(temper$STATIONS_ID[is.na(temper$Latitude)])
for(k in 1:length(fixvar)){
temper$Latitude[is.na(temper$Latitude)][which(temper$
    STATIONS_ID[is.na(temper$Latitude)]==fixvar[k])]=allstar
    $geoBreite[which(allstar$Stations_id==fixvar[k])]
temper$Longitude[is.na(temper$Latitude)][which(temper$
    STATIONS_ID[is.na(temper$Latitude)]==fixvar[k])]=allstar
    $geoLaenge[which(allstar$Stations_id==fixvar[k])]
}
fixvar=unique(temper$STATIONS_ID[is.na(temper$Longitude)])
for(k in 1:length(fixvar)){
temper$Longitude[is.na(temper$Longitude)][which(temper$
    STATIONS_ID[is.na(temper$Longitude)]==fixvar[k])]=
    allstar$geoLaenge[which(allstar$Stations_id==fixvar[k])]
}

##############################################################
```

```
# Following the precipitation data are downloaded #
##################################################

# First an empty dataframe is created; later one it will be
    filled with the precipitation data
nieder=as.data.frame(matrix(ncol=8,nrow=1,dimnames=list(1,c
    ("Station","STATIONS_ID","Latitude","Longitude","Zeit","
    Datum","Stunde","Niederschlag"))))

# The following part looks into the folder of the
    meteorological surveys recent precipitation ftp server
    and sorts the information
# In the end it receives a vector (star) with the
    information of the available observation stations in
    form of the available zip files
filenames=getURL("ftp://ftp-cdc.dwd.de/pub/CDC/observations
    _germany/climate/hourly/precipitation/recent/",ftp.use.
    epsv = FALSE,dirlistonly = TRUE)
filenames=gsub("zip","zipqwert", filenames)
filenames=strsplit(filenames, "qwert")
filenames=substring(filenames[[1]],3)
filenames[1]=paste("st",filenames[1],sep="")
star=filenames[which(nchar(filenames)==29)]

# Here the for-loop begins which includes the download and
    structuration of the precipitation data
pb=txtProgressBar(min = 1, max = length(star), style = 3)
for(j in 1:length(star)){

# Only go on when the location code (as part of the
    zipfiles name) is equal to one of the locations in
    northern germany
if(is.element(as.numeric(strsplit(as.character(star[j]), "_
    ")[[1]][3]),gestar$Stations_id)){

# Download the location specific zipfile and store it as
    temporal file
pfad=paste("ftp://ftp-cdc.dwd.de/pub/CDC/observations_
    germany/climate/hourly/precipitation/recent/",star[j],
    sep="")
download.file(pfad,"1. FTP Download\\tempdat.zip")
```

```
# Look into the zipfile and import the table containing "
    product_" in its name
# This table contains the weather information
statdat=read.csv2(unz("1. FTP Download\\tempdat.zip", unzip
    ("1. FTP Download\\tempdat.zip", list = T)$Name[grep("
    produkt_",unzip("1. FTP Download\\tempdat.zip", list = T
    )$Name)]))
# Look into the zipfile and import the table containing "
    Geographie" in its name
# This table contains the geographic information about the
    weather observation station
statinf=read.csv2(unz("1. FTP Download\\tempdat.zip",unzip(
    "1. FTP Download\\tempdat.zip", list = T)$Name[grep("
    Geographie",unzip("1. FTP Download\\tempdat.zip", list =
     T)$Name)]))

# The table "statinf" contains the information about the
    locations geographic location in different times
# Since the location is still active the information till
    when it is active is filled with the actual date
statinf$bis_datum[is.na(statinf$bis_datum)]=as.numeric(
    paste(substr(Sys.Date(),1,4),substr(Sys.Date(),6,7),
    substr(Sys.Date(),9,10),sep=""))
statinf$Geogr.Breite=as.numeric(as.character(statinf$Geogr.
    Breite))
statinf$Geogr.Laenge=as.numeric(as.character(statinf$Geogr.
    Laenge))

# The table containing the precipitation information gets
    the geographic location information and the name of the
    location
statdat$Latitude=NA
statdat$Longitude=NA
statdat$Station=paste(strsplit(as.character(statinf$
    Stationsname[1]), " ")[[1]][nchar(strsplit(as.character(
    statinf$Stationsname[1]), " ")[[1]])>0], collapse=" ")
# Since the location could change over the time, this must
    be considered giving the weather table the geographic
    location information
for(i in 1:dim(statinf)[1]){
```

```
statdat$Latitude[which(statdat$MESS_DATUM>=statinf$von_
    datum[i]*100 & statdat$MESS_DATUM<=statinf$bis_datum[i]*
    100)]=statinf$Geogr.Breite[i]
statdat$Longitude[which(statdat$MESS_DATUM>=statinf$von_
    datum[i]*100 & statdat$MESS_DATUM<=statinf$bis_datum[i]*
    100)]=statinf$Geogr.Laenge[i]
}

# The date and time information of the table is transformed
    to characters in form of the "chron" library
statdat$Zeit=as.character(chron(dates=paste(substr(statdat$
    MESS_DATUM,1,4),"-",substr(statdat$MESS_DATUM,5,6),"-",
    substr(statdat$MESS_DATUM,7,8),sep=""),times=paste(
    substr(statdat$MESS_DATUM,9,10),":00:00",sep=""),format=
    c('Y-m-d','h:m:s')))
if(dim(statdat[is.na(statdat$Zeit),])[1]>1){print("WARNUNG:
    Zu viele NAs")}else{statdat=statdat[!is.na(statdat$Zeit
    ),]}
statdat$NIEDERSCHLAGSHOEHE=as.numeric(as.character(statdat$
    R1))
statdat$Datum=as.Date(substr(statdat$MESS_DATUM,1,8),"%Y%m%
    d")
statdat$Stunde=as.numeric(substr(statdat$MESS_DATUM,9,10))

# The final dataset is extracted and renamed; only data
    newer than 2015 are considered further on, since older
    data are already stored locally
ausgabe=statdat[,c("Station","STATIONS_ID","Latitude","
    Longitude","Zeit","Datum","Stunde","NIEDERSCHLAGSHOEHE")
    ]
names(ausgabe)[c(8)]=c("Niederschlag")
ausgabe=ausgabe[which(as.integer(as.character(years(ausgabe
    $Datum)))>2015),]

# If the created output is not empty it is combined with
    the information of the other weather measuring stations
if(dim(ausgabe)[1]>1){
nieder=rbind(nieder,ausgabe)
}
}
setTxtProgressBar(pb, j)
```

```
}
close(pb)

# All precipitation information are combined; now empty
   rows are deleted and data are brought to the correct
   format
nieder=nieder[-1,]
nieder$Zeit=chron(dates=substr(nieder$Zeit,2,9),times=paste
   (substr(nieder$Zeit,11,12),":00:00",sep=""),format=c('Y-
   m-d','h:m:s'))
nieder$STATIONS_ID=as.numeric(nieder$STATIONS_ID)
nieder=nieder[!is.na(nieder$STATIONS_ID),]

# Should some precipitation data have no coordinates
   assigned, they get the coordinate of the actual location
    table with the fitting location
fixvar=unique(nieder$STATIONS_ID[is.na(nieder$Latitude)])
for(k in 1:length(fixvar)){
nieder$Latitude[is.na(nieder$Latitude)][which(nieder$
   STATIONS_ID[is.na(nieder$Latitude)]==fixvar[k])]=allstar
   $geoBreite[which(allstar$Stations_id==fixvar[k])]
nieder$Longitude[is.na(nieder$Latitude)][which(nieder$
   STATIONS_ID[is.na(nieder$Latitude)]==fixvar[k])]=allstar
   $geoLaenge[which(allstar$Stations_id==fixvar[k])]
}
fixvar=unique(nieder$STATIONS_ID[is.na(nieder$Longitude)])
for(k in 1:length(fixvar)){
nieder$Longitude[is.na(nieder$Longitude)][which(nieder$
   STATIONS_ID[is.na(nieder$Longitude)]==fixvar[k])]=
   allstar$geoLaenge[which(allstar$Stations_id==fixvar[k])]
}

#################################################################
# Following the temperature and humdidity data are #
#  merged with the precipitation data             #
#################################################################

# The dataframes are merged by the locations ID and the
   date time (chron format)
wetda=merge(temper,nieder,by=c("STATIONS_ID","Zeit"),all=T)
```

```
# If there were NA data in the temperatures dataframes
    location information, these are replaced with the
    precipitations data information
wetda$Station.x[is.na(wetda$Station.x)]=wetda$Station.y[is.
    na(wetda$Station.x)]
wetda$Latitude.x[is.na(wetda$Latitude.x)]=wetda$Latitude.y[
    is.na(wetda$Latitude.x)]
wetda$Longitude.x[is.na(wetda$Longitude.x)]=wetda$Longitude
    .y[is.na(wetda$Longitude.x)]
wetda$Datum.x[is.na(wetda$Datum.x)]=wetda$Datum.y[is.na(
    wetda$Datum.x)]
wetda$Stunde.x[is.na(wetda$Stunde.x)]=wetda$Stunde.y[is.na(
    wetda$Stunde.x)]

# Afterwards the redundant data are deleted and the rows
    named correctly
wetda$Station.y=NULL
wetda$Latitude.y=NULL
wetda$Longitude.y=NULL
wetda$Datum.y=NULL
wetda$Stunde.y=NULL
names(wetda)=c("STATIONS_ID","Zeit","Station","Latitude","
    Longitude","Datum","Stunde","Temperatur","Luftfeuchte","
    Niederschlag")

# The old, expendable tables are deleted
rm(temper,nieder)

#####################################################
# Following the wind data are downloaded #
#####################################################

# First an empty dataframe is created; later one it will be
     filled with the precipitation data
windr=as.data.frame(matrix(ncol=9,nrow=1,dimnames=list(1,c(
    "Station","STATIONS_ID","Latitude","Longitude","Zeit","
    Datum","Stunde","Windgesch","Windrichtung"))))

# The following part looks into the folder of the
    meteorological surveys recent wind ftp server and sorts
    the information
```

```
# In the end it receives a vector (star) with the
    information of the available observation stations in
    form of the available zip files
filenames=getURL("ftp://ftp-cdc.dwd.de/pub/CDC/observations
    _germany/climate/hourly/wind/recent/",ftp.use.epsv =
    FALSE,dirlistonly = TRUE)
filenames=gsub("zip","zipqwert", filenames)
filenames=strsplit(filenames, "qwert")
filenames=substring(filenames[[1]],3)
filenames[1]=paste("st",filenames[1],sep="")
star=filenames[which(nchar(filenames)==29)]

# Here the for-loop begins which includes the download and
    structuration of the wind data
pb=txtProgressBar(min = 1, max = length(star), style = 3)
for(j in 2:length(star)){

# Only go on when the location code (as part of the
    zipfiles name) is equal to one of the locations in
    northern germany
if(is.element(as.numeric(strsplit(as.character(star[j]), "_
    ")[[1]][3]),gestar$Stations_id)){

# Download the location specific zipfile and store it as
    temporal file
pfad=paste("ftp://ftp-cdc.dwd.de/pub/CDC/observations_
    germany/climate/hourly/wind/recent/",star[j],sep="")
download.file(pfad,"1. FTP Download\\tempdat.zip")

# Look into the zipfile and import the table containing "
    product_" in its name
# This table contains the weather information
statdat=read.csv2(unz("1. FTP Download\\tempdat.zip", unzip
    ("1. FTP Download\\tempdat.zip", list = T)$Name[grep("
    produkt_",unzip("1. FTP Download\\tempdat.zip", list = T
    )$Name)]))
# Look into the zipfile and import the table containing "
    Geographie" in its name
# This table contains the geographic information about the
    weather observation station
statinf=read.csv2(unz("1. FTP Download\\tempdat.zip",unzip(
```

```
    "1. FTP Download\\tempdat.zip", list = T)$Name[grep("
    Geographie",unzip("1. FTP Download\\tempdat.zip", list =
     T)$Name)]))

# The table "statinf" contains the information about the
    locations geographic location in different times
# Since the location is still active the information till
    when it is active is filled with the actual date
statinf$bis_datum[is.na(statinf$bis_datum)]=as.numeric(
    paste(substr(Sys.Date(),1,4),substr(Sys.Date(),6,7),
    substr(Sys.Date(),9,10),sep=""))
statinf$Geogr.Breite=as.numeric(as.character(statinf$Geogr.
    Breite))
statinf$Geogr.Laenge=as.numeric(as.character(statinf$Geogr.
    Laenge))

# The table containing the precipitation information gets
    the geographic location information and the name of the
    location
statdat$Latitude=NA
statdat$Longitude=NA
statdat$Station=paste(strsplit(as.character(statinf$
    Stationsname[1]), " ")[[1]][nchar(strsplit(as.character(
    statinf$Stationsname[1]), " ")[[1]])>0], collapse=" ")
# Since the location could change over the time, this must
    be considered giving the weather table the geographic
    location information
for(i in 1:dim(statinf)[1]){
statdat$Latitude[which(statdat$MESS_DATUM>=statinf$von_
    datum[i]*100 & statdat$MESS_DATUM<=statinf$bis_datum[i]*
    100)]=statinf$Geogr.Breite[i]
statdat$Longitude[which(statdat$MESS_DATUM>=statinf$von_
    datum[i]*100 & statdat$MESS_DATUM<=statinf$bis_datum[i]*
    100)]=statinf$Geogr.Laenge[i]
}

# The date and time information of the table is transformed
     to characters in form of the "chron" library
statdat$Zeit=as.character(chron(dates=paste(substr(statdat$
    MESS_DATUM,1,4),"-",substr(statdat$MESS_DATUM,5,6),"-",
    substr(statdat$MESS_DATUM,7,8),sep=""),times=paste(
```

```
    substr(statdat$MESS_DATUM,9,10),":00:00",sep=""),format=
    c('Y-m-d','h:m:s')))
if(dim(statdat[is.na(statdat$Zeit),])[1]>1){print("WARNUNG:
    Zu viele NAs")}else{statdat=statdat[!is.na(statdat$Zeit
    ),]}
statdat$WINDGESCHWINDIGKEIT=as.numeric(as.character(statdat
    $F))
statdat$WINDRICHTUNG=as.numeric(as.character(statdat$D))
statdat$Datum=as.Date(substr(statdat$MESS_DATUM,1,8),"%Y%m%
    d")
statdat$Stunde=as.numeric(substr(statdat$MESS_DATUM,9,10))

# The final dataset is extracted and renamed; only data
    newer than 2015 are considered further on, since older
    data are already stored locally
ausgabe=statdat[,c("Station","STATIONS_ID","Latitude","
    Longitude","Zeit","Datum","Stunde","WINDGESCHWINDIGKEIT"
    ,"WINDRICHTUNG")]
names(ausgabe)[c(8,9)]=c("Windgesch","Windrichtung")
ausgabe=ausgabe[which(as.integer(as.character(years(ausgabe
    $Datum)))>2015),]

# If the created output is not empty it is combined with
    the information of the other weather measuring stations
if(dim(ausgabe)[1]>1){
windr=rbind(windr,ausgabe)
}
setTxtProgressBar(pb, j)
}
}
close(pb)

# All wind data are combined; now empty rows are deleted
    and data are brought to the correct format
windr=windr[-1,]
windr$Zeit=chron(dates=substr(windr$Zeit,2,9),times=paste(
    substr(windr$Zeit,11,12),":00:00",sep=""),format=c('Y-m-
    d','h:m:s'))
windr$STATIONS_ID=as.numeric(windr$STATIONS_ID)
windr=windr[!is.na(windr$STATIONS_ID),]
```

```
# Should some wind data have no coordinates assigned, they
    get the coordinate of the actual location table with the
     fitting location
fixvar=unique(windr$STATIONS_ID[is.na(windr$Latitude)])
for(k in 1:length(fixvar)){
windr$Latitude[is.na(windr$Latitude)][which(windr$STATIONS_
    ID[is.na(windr$Latitude)]==fixvar[k])]=allstar$geoBreite
    [which(allstar$Stations_id==fixvar[k])]
windr$Longitude[is.na(windr$Latitude)][which(windr$STATIONS
    _ID[is.na(windr$Latitude)]==fixvar[k])]=allstar$
    geoLaenge[which(allstar$Stations_id==fixvar[k])]
}
fixvar=unique(windr$STATIONS_ID[is.na(windr$Longitude)])
for(k in 1:length(fixvar)){
windr$Longitude[is.na(windr$Longitude)][which(windr$
    STATIONS_ID[is.na(windr$Longitude)]==fixvar[k])]=allstar
    $geoLaenge[which(allstar$Stations_id==fixvar[k])]
}

##############################################################
# Following the already merged data are combined #
#  with the wind data                                        #
##############################################################

# First old, expendable variables are deleted
rm("allstar","ausgabe","gestar","i","j","k","pb","pfad","
    star","statdat","statinf")

# Then the dataframes are merged by the locations ID and
    the date time (chron format)
wetda2=merge(wetda,windr,by=c("STATIONS_ID","Zeit"),all=T)

# If there were NA data in the merged before dataframes
    location information, these are replaced with the wind
    data information
wetda2$Station.x[is.na(wetda2$Station.x)]=wetda2$Station.y[
    is.na(wetda2$Station.x)]
wetda2$Latitude.x[is.na(wetda2$Latitude.x)]=wetda2$Latitude
    .y[is.na(wetda2$Latitude.x)]
wetda2$Longitude.x[is.na(wetda2$Longitude.x)]=wetda2$
    Longitude.y[is.na(wetda2$Longitude.x)]
```

```
wetda2$Datum.x[is.na(wetda2$Datum.x)]=wetda2$Datum.y[is.na(
    wetda2$Datum.x)]
wetda2$Stunde.x[is.na(wetda2$Stunde.x)]=wetda2$Stunde.y[is.
    na(wetda2$Stunde.x)]

# Afterwards the redundant data are deleted and the rows
    named correctly
wetda2$Station.y=NULL
wetda2$Latitude.y=NULL
wetda2$Longitude.y=NULL
wetda2$Datum.y=NULL
wetda2$Stunde.y=NULL
names(wetda2)=c("STATIONS_ID","Zeit","Station","Latitude","
    Longitude","Datum","Stunde","Temperatur","Luftfeuchte","
    Niederschlag","Windgesch","Windrichtung")

# Then the data containing the german meteorological
    surveys error code "-999" are replaced by the R intern
    NoData value NA
wetda2$Temperatur[which(wetda2$Temperatur==-999)]=NA
wetda2$Luftfeuchte[which(wetda2$Luftfeuchte==-999)]=NA
wetda2$Niederschlag[which(wetda2$Niederschlag==-999)]=NA
wetda2$Windgesch[which(wetda2$Windgesch==-999)]=NA
wetda2$Windrichtung[which(wetda2$Windrichtung==-999)]=NA
wetda2$Datum=as.Date(substr(wetda2$Zeit,2,9),"%y-%m-%d")
wetda2$Stunde=as.numeric(substr(wetda2$Zeit,11,12))

##########################################################################
# Following the new downloaded weather data are merged #
# with the weather data, downloaded before             #
##########################################################################

# The file "wetdat_bisinkl2015.RData" contains the variable
    "wetdat" which consists of the weather data from 1995
    to 2015
load(file = "1. FTP Download\\wetdat_bisinkl2015.RData")

# The old dataset is combined with the new weather dataset
wetdat=rbind(wetdat,wetda2)

# Potentialle wrong data in the wind direction are excluded
```

```
wetdat$Windrichtung[which(wetdat$Windrichtung>360)]=NA

# Finally the dataset consisting of the weather data from
    1995 to 2017 is saved as variable "wetdat" in the file "
    wetda_aktuell.RData"
save(wetdat, file = "Daten\\Wetterdaten\\wetda_aktuell.
    RData")
```

B.3 Interpolation of weather data

Data used First, a brief overview of the data used in this script is given:

The file *"wetterundraster.RData"* contains a raster stack, a combination of different rasters. Its structure can be seen in figure 3.5.

The file *"wetda_aktuell.RData"* contains the dataframe wetdat created in the foregone script:

Table B.2: wetdat

STATIONS_ID	Zeit	Station	Latitude	Longitude	Datum	Stunde	Temperatur	Luftfeuchte	Niederschlag	Windgesch	Windrichtung
704	(96-01-26 09:00:00)	Bremervörde	53.50	9.17	1996-01-26	9	-9.90	74.00	0.00	7.00	80.00
704	(96-01-26 10:00:00)	Bremervörde	53.50	9.17	1996-01-26	10	-8.50	66.00	0.00	7.20	80.00
704	(96-01-26 11:00:00)	Bremervörde	53.50	9.17	1996-01-26	11	-7.50	64.00	0.00	6.80	90.00
704	(96-01-26 12:00:00)	Bremervörde	53.50	9.17	1996-01-26	12	-6.50	55.00	0.00	7.30	90.00
704	(96-01-26 13:00:00)	Bremervörde	53.50	9.17	1996-01-26	13	-6.20	53.00	0.00	7.30	90.00
704	(96-01-26 14:00:00)	Bremervörde	53.50	9.17	1996-01-26	14	-6.40	56.00	0.00	7.00	90.00

The shapefiles *"counties2oh"*, *"counties_polygoh"*, *"Bundeslaender_BKG_UTM"* and *"Bundeslaender_BKG_UTM_l"* contain the geometries of the study area and the surrounding.

```
# This script aims at the automated regionalization of the
    recent weather data of the German Meteorological Survey
# Created 2016-08-04
# Created by: Wolfgang B. Hamer

# First the required packages need to be installed and the
    working directory needs to be defined
library(gstat)       # Spatial Geostatistical Modelling
library(rgdal)# Access to projection/transformation ops
library(raster)    # Working with gridded spatial data
library(chron)         # Manage time
library(deldir)# Calculates the Delaunay triangulation
library(RColorBrewer)# Provides color schemes for maps
library(automap)       # Performs automatic interpolation
library(sampSurf)      # Sampling surface simulation
library(rgeos)         # Interface to Geometry Engine
setwd("D:/Doktorarbeit/R_Scripte/PaPros")

# Then manually created additional functions are
    implemented
source("2. Variablen Regionalisieren/NuetzlicheFunktionen.R
    ")

# Load the file containing the raster stack "gradienten"
```

```
# "gradienten" contains the regionalized climatic data of
    the german meteorological surveys ftp server
# as well as the SRTM elevation model and derived
    parameters like slope, etc.
load("Daten/Wetterdaten/wetterundraster.RData")

# Load the file containing the variable "wetdat"
# "wetdat" contains the hourly local weather data which
    will be interpolated in this script
load("Daten/Wetterdaten/wetda_aktuell.RData")

# A variable is defined containing the coordinate system
    ETRS89 / UTM zone 32N
CRS.new=CRS("+init=epsg:25832")

# Polygons of Germany and Schleswig-Holsteins are imported
    and transformed
shmap1=readOGR("Daten/Geometrien","counties2oh")
shmap=spTransform(shmap1, CRS.new)
shmapol=readOGR("Daten/Geometrien","counties_polygoh")
shmapo=spTransform(shmapol, CRS.new)
demap=readOGR("Daten/Geometrien","Bundeslaender_BKG_UTM")
demap=spTransform(demap, CRS.new)
demapl=readOGR("Daten/Geometrien","Bundeslaender_BKG_UTM_l"
    )
demapl=spTransform(demapl, CRS.new)

# An empty SpatialGridDataFrame is created
# It will be filled with the interpolated information
grdtop=GridTopology(cellcentre.offset=c(425260,5913600),
    cellsize=c(2000,2000), cells.dim=c(115,95))
grsp=SpatialGrid(grdtop)
proj4string(grsp)=CRS("+init=epsg:25832")

# All possible years to deal with are evaluated by checking
     which year has been interpolated
# For each year/month combination calculated before a .csv
    file is stored in folder "Daten/Regionalisierte
    Wetterdaten/"
# All years between the actual year and the newest foregone
     year are identified
```

```
mogljahr=sort(as.character(max(as.numeric(substr(dir(path=
    paste("Daten/Regionalisierte␣Wetterdaten/",sep="")),
    pattern ="csv"),8,11))):2017))

# For all of these years the interpolation takes place (
    variable "u")
for(u in mogljahr){

# All times for which weather data are available are subset
    by the year, sorted and saved as vector "zeitpuj"
zeitpuj=sort(unique(wetdat$Zeit[which(substr(wetdat$Datum
    ,1,4)==u)]))

# First of all year/month combination calculated before the
    ones in the year we are looking at are selected
dop=gsub(paste("optimix",u,"_",sep=""),"",dir(path=paste("
    Daten/Regionalisierte␣Wetterdaten/",sep=""),pattern ="
    csv")[which(as.numeric(substr(dir(path=paste("Daten/
    Regionalisierte␣Wetterdaten/",sep=""),pattern ="csv")
    ,8,11))==max(as.numeric(substr(dir(path=paste("Daten/
    Regionalisierte␣Wetterdaten/",sep=""),pattern ="csv")
    ,8,11))))])
# Of the selected combinations the newest is selected and
    stored as "dop"
dop=paste("optimix",u,"_",max(as.numeric(gsub(".csv","",dop
    ))),".csv",sep="")

# The following part takes place for each month available
    in the local weahter data (variable "t") in year "u"
for(t in 1:length(unique(months(zeitpuj)))){

# Only if for the month no .csv file exists, indicating a
    completed interpolation for this month ...
if(is.element(paste("optimix",u,"_",as.character(t),".csv",
    sep=""),dir(path=paste("Daten/Regionalisierte␣
    Wetterdaten/",sep=""),pattern ="csv")[which(dir(path=
    paste("Daten/Regionalisierte␣Wetterdaten/",sep=""),
    pattern ="csv")!=dop)])){
print(paste("optimix",u,"_",as.character(t),".csv","␣
    already␣exists!",sep=""))
}else{
```

```
# ... the following steps take place:

# Each time of the month for which weather data exist is
    saved in the vector "zeitpu"
zeitpu=zeitpuj[which(months(zeitpuj)==unique(months(zeitpuj
    ))[t])]

# Then the weather data for this month are selected and
    saved as dataframe "jewr"
jewr=wetdat[is.element(wetdat$Zeit,zeitpu),]

# A matrix of all weather stations in this month is created
     (and later stored as .csv)
# In no weather stations are availabe the month will be
    skipped
mart=matrix(NA,1,4)
mart[1,1]=paste(as.character(unique(months(zeitpuj))[t]),u)
mart[1,2]=length(unique(jewr$STATIONS_ID[!is.na(jewr$
    Temperatur)]))
mart[1,3]=length(unique(jewr$STATIONS_ID[!is.na(jewr$
    Niederschlag)]))
mart[1,4]=length(unique(jewr$STATIONS_ID[!is.na(jewr$
    Windgesch)]))
colnames(mart)=c("Monat&Jahr","Temperatur&Luftfeuchte","
    Niederschlag","Windgeschw&Richtung")
if(any(as.numeric(mart[,c(2:4)])==0)){
print(paste(mart[1,1],"nicht bearbeitet da zu wenig Daten!"
    ))
mart
flush.console()
}else{

# Variable "berg" is defined to stop the processing time
berg=Sys.time()

# The following events take place for each hour "i" of the
    month "t" of the year "u"
for(i in 1:length(zeitpu)){

# All weather data of hour "i" are extracted from "wetdat"
    and NA values of the coordinates are erased
```

```
jewe=wetdat[which(wetdat$Zeit==zeitpu[i]),]
jewe=jewe[!is.na(jewe$Latitude),]
jewe=jewe[!is.na(jewe$Longitude),]
jewe$Latitude=as.numeric(as.character(jewe$Latitude))
jewe$Longitude=as.numeric(as.character(jewe$Longitude))

# The dataframe gets spatialised and transformed to the
    coordinate system of "gradienten"
coordinates(jewe) = c("Longitude","Latitude")
proj4string(jewe)=CRS("+init=epsg:4326") # WGS 84
jewe=spTransform(jewe, projection(gradienten))

# The elevation information for the weather measuring
    stations are extracted of the elevation model
# Locations without fitting raster are erased
jewe$dgm100=extract(raster(gradienten,layer="dgm100"),jewe)
jewe=jewe[!is.na(jewe$dgm100),]
jewe$Aspect=extract(raster(gradienten,layer="Aspect"),jewe)
jewe=jewe[!is.na(jewe$Aspect),]
jewe$Slope=extract(raster(gradienten,layer="Slope"),jewe)
jewe=jewe[!is.na(jewe$Slope),]

# Also the climatic information for the weather measuring
    stations are extracted of the raster stack "gradienten"
# Locations without fitting raster are erased
jewe$Bodenfeuchte=extract(raster(gradienten,layer="
    Bodenfeuchte"),jewe)
jewe=jewe[!is.na(jewe$Bodenfeuchte),]
jewe$Bodentemperatur=extract(raster(gradienten,layer="
    Bodentemperatur"),jewe)
jewe=jewe[!is.na(jewe$Bodentemperatur),]
jewe$DroughtIndex=extract(raster(gradienten,layer="
    DroughtIndex"),jewe)
jewe=jewe[!is.na(jewe$DroughtIndex),]
jewe$Eistage=extract(raster(gradienten,layer="Eistage"),
    jewe)
jewe=jewe[!is.na(jewe$Eistage),]
jewe$PotEvap=extract(raster(gradienten,layer="PotEvap"),
    jewe)
jewe=jewe[!is.na(jewe$PotEvap),]
jewe$RealEvap=extract(raster(gradienten,layer="RealEvap"),
```

```
    jewe)
jewe=jewe[!is.na(jewe$RealEvap),]
jewe$Frosttage=extract(raster(gradienten,layer="Frosttage")
    ,jewe)
jewe=jewe[!is.na(jewe$Frosttage),]
jewe$Heissetage=extract(raster(gradienten,layer="Heissetage
    "),jewe)
jewe=jewe[!is.na(jewe$Heissetage),]
jewe$MaxLufttemp=extract(raster(gradienten,layer="
    MaxLufttemp"),jewe)
jewe=jewe[!is.na(jewe$MaxLufttemp),]
jewe$MeanLufttemp=extract(raster(gradienten,layer="
    MeanLufttemp"),jewe)
jewe=jewe[!is.na(jewe$MeanLufttemp),]
jewe$MinLufttemp=extract(raster(gradienten,layer="
    MinLufttemp"),jewe)
jewe=jewe[!is.na(jewe$MinLufttemp),]
jewe$MonNiederschlag=extract(raster(gradienten,layer="
    MonNiederschlag"),jewe)
jewe=jewe[!is.na(jewe$MonNiederschlag),]
jewe$Schneetage=extract(raster(gradienten,layer="Schneetage
    "),jewe)
jewe=jewe[!is.na(jewe$Schneetage),]
jewe$Sommertage=extract(raster(gradienten,layer="Sommertage
    "),jewe)
jewe=jewe[!is.na(jewe$Sommertage),]
jewe$Sonnenscheindauer=extract(raster(gradienten,layer="
    Sonnenscheindauer"),jewe)
jewe=jewe[!is.na(jewe$Sonnenscheindauer),]
jewe$Wasserbilanz=extract(raster(gradienten,layer="
    Wasserbilanz"),jewe)
jewe=jewe[!is.na(jewe$Wasserbilanz),]
jewe$Windgeschwindigkeit=extract(raster(gradienten,layer="
    Windgeschwindigkeit"),jewe)
jewe=jewe[!is.na(jewe$Windgeschwindigkeit),]

# Also the corodinates are defined
jewe$xcord=coordinates(jewe)[,1]
jewe$ycord=coordinates(jewe)[,2]

#####################################################################
```

```
# Following the temperature data will be interpolated #
##########################################################
# First the check, if at least three valid locations are
    available
if(any(!is.na(jewe$Temperatur))){
if(length(jewe$Temperatur[!is.na(jewe$Temperatur)])>3){

# Then SpatialPointDataFrame "tempint" is created,
    containing only valid temperature data
tempint=jewe[!is.na(jewe[paste("Temperatur")]@data[,1]),]

# Then the function tries to interpolate the termperature
    data using KED (autoKrige) with the covariables:
# x coordinate
# y coordinate
# Climatic mean air temperature
vers=try(autoKrige(Temperatur~xcord+ycord+MeanLufttemp,
    tempint, as(gradienten, 'SpatialGridDataFrame')), silent
    =TRUE)

# If an error occurs the IDW method is used
if ('try-error' %in% class(vers)){
iw=idw(Temperatur~1, tempint,as(gradienten, '
    SpatialGridDataFrame'))
temp=raster(iw)
names(temp)="tempKED"

# Else the KED result is used, brought to the raster format
     and named "tempKED"
}else{
uk = vers
temp=raster(uk$krige_output)
names(temp)="tempKED"
}

# If not enough valid locations are available an empty
    raster is created
}else{
temp=gradienten$Bodenfeuchte
temp@data@values=NA
names(temp)="tempKED"
```

```
}
}else{
temp=gradienten$Bodenfeuchte
temp@data@values=NA
names(temp)="tempKED"
}

#########################################################
# Following the windspeed will be interpolated #
#########################################################
# First the check, if at least three valid locations are
    available
if(any(!is.na(jewe$Windgesch))){
if(length(jewe$Windgesch[!is.na(jewe$Windgesch)])>2){

# Then SpatialPointDataFrame "tempint" is created,
    containing only valid windspeed data
tempint=jewe[!is.na(jewe[paste("Windgesch")]@data[,1]),]

# Then the function tries to interpolate the windspeed data
     using KED (autoKrige) with the covariables:
# x coordinate
# y coordinate
# Climatic windspeed
vers=try(autoKrige(Windgesch~xcord+ycord+
    Windgeschwindigkeit, tempint, as(gradienten, '
    SpatialGridDataFrame')), silent=TRUE)

# If an error occurs the IDW method is used
if ('try-error' %in% class(vers)){
iw=idw(Windgesch~1, tempint,as(gradienten, '
    SpatialGridDataFrame'))
win=raster(iw)
names(win)="winKED"

# Else the KED result is used, brought to the raster format
     and named "winKED"
}else{
uk = vers
win=raster(uk$krige_output)
names(win)="winKED"
```

```
}

# If not enough valid locations are available an empty
    raster is created
}else{
win=gradienten$Bodenfeuchte
win@data@values=NA
names(win)="winKED"
}
}else{
win=gradienten$Bodenfeuchte
win@data@values=NA
names(win)="winKED"
}

# If the interpolated windspeed is below 0 it is set to 0
win[win<0]=0

#####################################################################
# Following the winddirection will be interpolated #
#####################################################################
# First the check, if at least one valid location is
    available
if(any(!is.na(jewe$Windrichtung))){

# Then SpatialPointDataFrame "tempint" is created,
    containing only valid winddirection data
tempint=jewe[!is.na(jewe[paste("Windrichtung")]@data[,1]),]

# Method vop of the additional functions is used to
    regionalize the winddirections as thiessen polygons
winri=vop(jewe=tempint,"Windrichtung",shmapo)

# Afterwards the raster gets resampled to fit to the "
    gradienten" raster stack
winri=resample(winri,gradienten)
names(winri)="winriVor"

# If not enough valid locations are available an empty
    raster is created
}else{
```

```
winri=gradienten$Bodenfeuchte
winri@data@values=NA
names(winri)="winriVor"
}

#####################################################################
# Following the precipitation data will be interpolated #
#####################################################################
# First the check, if at least four valid location is
    available
if(any(!is.na(jewe$Niederschlag))){
if(length(jewe$Niederschlag[!is.na(jewe$Niederschlag)])>3){

# Then SpatialPointDataFrame "tempint" is created,
    containing only valid precipitation data
tempint=jewe[!is.na(jewe[paste("Niederschlag")]@data[,1]),]

# The precipitation data are interpolated using the IDW
    procedure
iw=idw(Niederschlag~1, tempint,as(gradienten, '
    SpatialGridDataFrame'))
nied=raster(iw)
names(nied)="niedOK"

# If all observed precipitation data are 0 a zero raster is
     created
} else if (mean(jewe$Niederschlag[!is.na(jewe$Niederschlag)
    ])==0){
nied=gradienten$Bodenfeuchte
nied@data@values=rep(0,length(nied@data@values))
names(nied)="niedOK"

# If not enough valid locations are available an empty
    raster is created
}else{
nied=gradienten$Bodenfeuchte
nied@data@values=NA
names(nied)="niedOK"
}
}else{
nied=gradienten$Bodenfeuchte
```

```
nied@data@values=NA
names(nied)="niedOK"
}

# If the interpolated precipitation is below 0 it is set to
    0
nied[nied<0]=0

##############################################################
# Following the humidity data will be interpolated #
##############################################################
# First the check, if at least three valid locations are
    available
if(any(!is.na(jewe$Luftfeuchte))){
if(length(jewe$Luftfeuchte[!is.na(jewe$Luftfeuchte)])>3){

# Then SpatialPointDataFrame "tempint" is created,
    containing only valid humidity data
tempint=jewe[!is.na(jewe[paste("Luftfeuchte")]@data[,1]),]

# If all valid humidity data are equal the IDW procedure is
    used for the interpolation
if(sd(jewe$Luftfeuchte[!is.na(jewe$Luftfeuchte)])==0){
iw=idw(Luftfeuchte~1, tempint,as(gradienten, '
    SpatialGridDataFrame'))
luf=raster(iw)
names(luf)="lufOK"
}else{

# Otherwise the function tries to interpolate the humidity
    data using KED (autoKrige) with the covariables:
# x coordinate
# y coordinate
# Climatic Drought Index
vers=try(autoKrige(Luftfeuchte~xcord+ycord+DroughtIndex,
    tempint, as(gradienten, 'SpatialGridDataFrame')), silent
    =TRUE)

# If an error occurs the IDW method is used
if ('try-error' %in% class(vers)){
iw=idw(Luftfeuchte~1, tempint,as(gradienten, '
```

```
    SpatialGridDataFrame'))
luf=raster(iw)
names(luf)="lufOK"

# Else the KED result is used, brought to the raster format
    and named "lufOK"
}else{
ok = vers
luf=raster(ok$krige_output)
names(luf)="lufOK"
}
}

# If not enough valid locations are available an empty
    raster is created
}else{
luf=gradienten$Bodenfeuchte
luf@data@values=NA
names(luf)="lufOK"
}
}else{
luf=gradienten$Bodenfeuchte
luf@data@values=NA
names(luf)="lufOK"
}

# If the interpolated precipitation is below 0 or above 100
    it is set to 0 or 100
luf[luf>100]=100
luf[luf<0]=0

# The interpolated rasters are combined in a raster stack
    named after the hour and date
assign(paste("wetter",substr(zeitpu[i],2,3),substr(zeitpu[i
    ],5,6),substr(zeitpu[i],8,9),substr(zeitpu[i],11,12),sep
    ="")),stack(temp,win,winri,nied,luf))

# In regular gaps the progress status is given out
if(i%%10==0){
plot(get(paste("wetter",substr(zeitpu[i],2,3),substr(zeitpu
    [i],5,6),substr(zeitpu[i],8,9),substr(zeitpu[i],11,12),
```

```
    sep="")))
print(paste("Hour",i,"of",length(zeitpu),"concluded at",Sys
    .time()))
print(paste("That means",round((100/length(zeitpu))*i,2),"%
    of",as.character(unique(months(zeitpuj))[t]),u))
print(paste("End at",(((Sys.time()-berg)/i)*(length(zeitpu)
    -i))+Sys.time()))
flush.console()
}
}
}

# If enough locations of the german meteorological survey
    exist the processed data are saved as .RData file
if(!any(as.numeric(mart[,c(2:4)])==0)){
save.image(paste("Daten/Regionalisierte Wetterdaten/optimix
    ",u,"_",as.character(t),".RData",sep=""))
}

# Also the information about the location count is saved as
    the .csv file, which was used in the beginning of the
    script
write.csv2(mart,paste("Daten/Regionalisierte Wetterdaten/
    optimix",u,"_",as.character(t),".csv",sep=""))
}

# Finally all interpolated weather rasters of the month are
    deleted and the data of the next month follow in the
    next iteration
rm(list = ls(pattern="wetter"))
gc(T)
}
}
#q("no")
```

B.4 Aggregation of interpolated weather data

Data used First, a brief overview of the data used in this script is given:

As can be seen in the foregone script, the interpolated data are stored in a specific way. The following script checks this order and aggregates on a monthly scale. As exemplary input, the file *"optimix2017_4.RData"* consists of one raster stack for each hour of April 2017. For example, the raster stack *wetter17041602* contains the interpolated weather data for April the 16th 2017 at two o' clock in the morning. The structure is displayed in figure B.1.

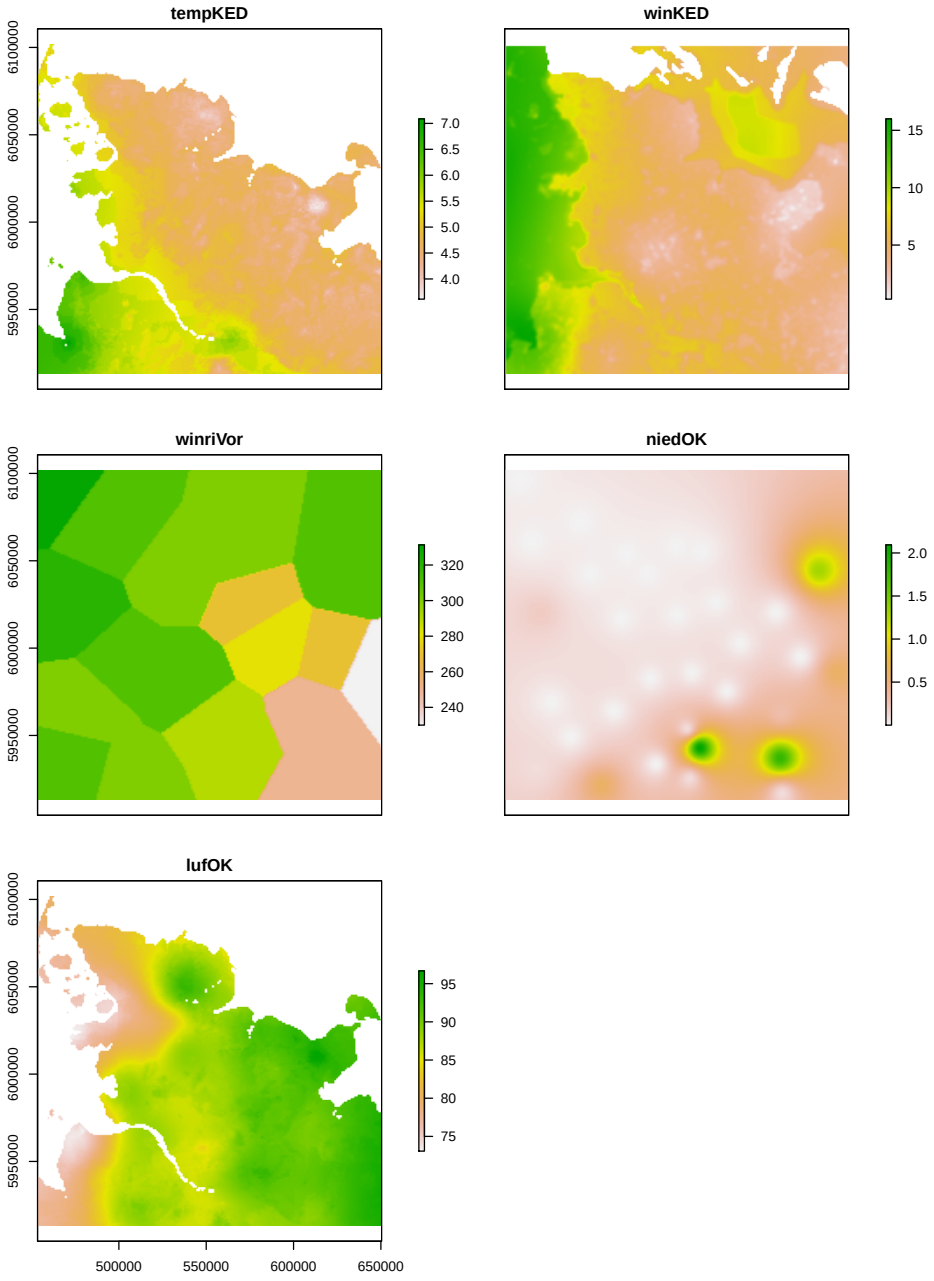


Figure B.1: wetter17041602

```
# This script aims at the automated aggregate of the recent
    regionalized weather data of the German Meteorological
    Survey
# The aggregation depends on the assumed durations of
    infection and incubation
# The weather data of the infection duration are aggregated
    ...
# ... and allocated to a date as many days later as the
    incubation duratoin
# Created 2017-01-13
# Created by: Wolfgang B. Hamer

# First the required packages need to be installed and the
    working directory needs to be defined
library(raster) # Working with gridded spatial data
library(chron)  # Manage time
setwd("D:/Doktorarbeit/R_Scripte/PaPros")

# Then manually created additional functions are
    implemented
source("2. Variablen Regionalisieren/NuetzlicheFunktionen.R
    ")


# Here the assumed durations of infection and incubation
    are defined
infper=3  # infection  (days)
incper=9  # incubation (days)

# First the script checks, for which years aggregations
    already exist
schoex=dir(path=paste("Daten/Zusammengefasste Wetterdaten/"
    ,sep=""),pattern ="kwett")
schoex=gsub("kwett","",gsub(".RData","",schoex))
if(any(schoex=="NA")){schoex=schoex[which(schoex!="NA")]}

# The last year is excluded, since it possible could not be
    finished
schoex=schoex[-length(schoex)]

# Then the script checks, for which years regionalized
```

```
    weather data exist
jada=dir(path=paste("Daten/Regionalisierte_Wetterdaten/",
    sep=""),pattern ="optimix")
jada=jada[grep(".RData",jada)]
jaa=unique(substr(jada,8,11))

# The years which are possible not finally aggregated are
    saved as variable "jaa"
jaa=jaa[(length(schoex)+1):length(jaa)]

# Since the aggregated weather data of one year beginn
    aggregating with the seed of the ...
# ... winter wheat in October of the foregone year, the
    aggregation begins with the second available year ("er")
for(er in 2:length(jaa)){

# Of the available regionalized monthly weather data, a
    list of all available months in the relevant period ...
# ... October of the foregone year to September of year "er
    " is subset and stored as vector "akj"
akj=jada[c(grep(paste(jaa[er-1],"_10",sep=""),jada),grep(
    paste(jaa[er-1],"_11",sep=""),jada),grep(paste(jaa[er
    -1],"_12",sep=""),jada),grep(paste(jaa[er],"_1.RData",
    sep=""),jada),grep(paste(jaa[er],"_2.RData",sep=""),jada
    ),grep(paste(jaa[er],"_3.RData",sep=""),jada),grep(paste
    (jaa[er],"_4.RData",sep=""),jada),grep(paste(jaa[er],"_
    5.RData",sep=""),jada),grep(paste(jaa[er],"_6.RData",sep
    =""),jada),grep(paste(jaa[er],"_7.RData",sep=""),jada),
    grep(paste(jaa[er],"_8.RData",sep=""),jada),grep(paste(
    jaa[er],"_9.RData",sep=""),jada))]

# The following happens for specific months "w" of period "
    akj"
for(w in 1:length(akj)){

# The regionalized weatherdata of the specific month are
    imported
load(paste("Daten/Regionalisierte_Wetterdaten/",akj[w],sep=
    ""))

# Then the available days of the regionalized data are
```

```
   found out and stored as "tage"
tage=sort(as.Date(unique(substr(ls(pattern="wetter"),7,12))
   ,"%y%m%d"))

print(paste(gsub(".RData","",gsub("optimix","",akj[w]))))
flush.console()

# The following happens for each of the available days ("i
   ") with the forerun of ...
# ... the count of days to be summarized for the infection
   period ("infper")
pb=txtProgressBar(min = 2, max = length(tage), style = 3)
for(i in infper:length(tage)){

# The rasters of the days to be aggregated for this day are
    selected
# First the rasters of the foregone days if the infection
   period is longer than one day
if(infper>1){
for(z in 1:(infper-1)){
assign(paste("stacker_",z,sep=""),get(ls(pattern=paste("
   wetter",substr(tage[i]-z,3,4),substr(tage[i]-z,6,7),
   substr(tage[i]-z,9,10),sep=""))))
}
}
# And second the weather data of the actual day
assign(paste("stacker_",infper,sep=""),get(ls(pattern=paste
   ("wetter",substr(tage[i],3,4),substr(tage[i],6,7),substr
   (tage[i],9,10),sep=""))))

# Then the selected days are stacked together
neu=do.call("stack",lapply(c(ls(pattern="stacker_")), get))

# And the redundant rasters are deleted
rm(list = ls(pattern="stacker_"))

# Calculating the mean Temperature of the first of the
   aggregated days to calculate the DTU
wedda=do.call("stack",lapply(ls(pattern=paste("wetter",
   substr(tage[i]-(infper-1),3,4),substr(tage[i]-(infper-1)
   ,6,7),substr(tage[i]-(infper-1),9,10),sep="")), get))
```

```
wedda=dropLayer(wedda, grep("temp",names(wedda),invert=T))
wedda=mean(wedda,na.rm=T)

# Applying the dtu function at the averaged temperature
    raster and assigning this to a raster with a date...
# ... the days of the incubation later
assign(paste("dtue",substr(tage[i]+incper,3,4),substr(tage[
    i]+incper,6,7),substr(tage[i]+incper,9,10),sep=""),dtu(
    wedda))   # +incper As period of incubation

# Subsetting the weather data from the raster stack
tek=neu[[which(grepl("temp", names(neu)))]]
wik=neu[[which(grepl("winKED", names(neu)))]]
wirk=neu[[which(grepl("winri", names(neu)))]]
nik=neu[[which(grepl("nied", names(neu)))]]
luk=neu[[which(grepl("luf", names(neu)))]]

# Creating a stack of the min/mean/max weather data, the
    actual DTU and the CTU values and renaming its
    valriables and its name
kwett=stack(min(tek,na.rm=T),max(tek,na.rm=T),mean(tek,na.
    rm=T),
min(wik,na.rm=T),max(wik,na.rm=T),mean(wik,na.rm=T),
modus(wirk),
min(nik,na.rm=T),max(nik,na.rm=T),mean(nik,na.rm=T),
min(luk,na.rm=T),max(luk,na.rm=T),mean(luk,na.rm=T),dtu(
    wedda),sum(do.call("stack",lapply(ls(pattern="dtue"),
    get)),na.rm=T))
names(kwett)=c("mintemp","maxtemp","meantemp","minwin","
    maxwin","meanwin","uerwinr","minnid","maxnid","meannid",
    "minluf","maxluf","meanluf","DTU","CTU")
assign(paste("kwett",substr(tage[i]+incper,3,4),substr(tage
    [i]+incper,6,7),substr(tage[i]+incper,9,10),sep=""),
    kwett)       # +9 As period of incubation

setTxtProgressBar(pb, i)
}
close(pb)

# Deleting all but the two last weather elements (necessary
    to keep for the summary of infper days)
```

```
if(length(unique(substr(ls(pattern="wetter"),7,8))[is.
    element(unique(substr(ls(pattern="wetter"),7,8)),c("00",
    "99"))])==2){
 rm(list = ls(pattern="wetter99"))
 rm(list = ls(pattern="wetter00")[-c((length(ls(pattern="
     wetter00"))-(infper-2)):length(ls(pattern="wetter00")))
     ])
}else{
 rm(list = ls(pattern="wetter")[-c((length(ls(pattern="
     wetter"))-(infper-2)):length(ls(pattern="wetter")))])
}
gc(T)
}

# Deleting all needless variables and save the data of this
    year
rm(list = ls(pattern="wetter"))
rm(list = ls(pattern="dtue"))
save.image(paste("Daten/Zusammengefasste␣Wetterdaten/kwett"
    ,jaa[er],".RData",sep=""))
rm(list = ls(pattern="kwett"))
gc(T)
}
```

B.5 Evaluation of prediction methods

Data used First, a brief overview of the data used in this script is given:

The file *"onedataset.RData"* contains the dataframe dataset created of the aggregated weather data created in the foregone script, the climate data derived by CDC and the infestation information:

Table B.3: onedataset

Standort	Datum	mintemp	maxtemp	meantemp	minwin	maxwin	meanwin	minnid	maxnid
Barlt	1996-05-06	6.50	7.99	7.37	0.77	3.33	2.13	0.00	0.01
Barlt	1999-05-03	7.46	8.56	8.09	2.77	3.84	3.24	0.00	0.00
Barlt	2014-06-16	10.11	15.10	12.38	1.70	6.33	3.66	0.00	0.56

meannid	minluf	maxluf	meanluf	DTU	CTU	Bodentemperatur	DroughtIndex	RealEvap	Frosttage
0.00	86.45	96.50	92.36	3.80	324.99	101.67	3.58	363.66	62.00
0.00	88.14	95.74	92.70	4.49	276.06	101.67	3.58	363.66	62.00
0.19	89.79	93.15	91.01	7.58	721.57	101.67	3.58	363.66	62.00

Heissetage	MaxLufttemp	MeanLufttemp	MinLufttemp	MonNiederschlag	Windgeschwindigkeit	dgm100	Klasse	IBHB	IBHBF
2.08	124.08	90.00	57.00	869.12	1801.34	0.08	9	10.00	0.00
2.08	124.08	90.00	57.00	869.12	1801.34	0.08	9	0.00	0.00
2.08	124.08	90.00	57.00	869.12	1801.34	0.08	5	0.00	0.00

```
## This script aims at the usuage of different
   classification methods to predict disease incidences
# Created 2017-04-16
# Created by: Wolfgang B. Hamer

# Setting the working directory
setwd("E:\\Doktorarbeit\\R Skripte\\2. Methodenvergleich\\
   Random Forest testarea\\ClusterForest")
#setwd("C:\\Users\\Wolfgang\\Desktop\\Dok\\
   Methodenvergleich\\ClusterForest")
library(chron)
# Including the package randomForest for the randomForest
   algorithms
library(randomForest)
# Including the package class for the knn algorithms
library("class")
# Including the package C50 for the decision tree
   algorithms
library(C50)
#Including packages for the result interpretation
library(ggplot2)
```

```
library(caret)
library(ROCR)
# Including packages for the parallel processing parts
library(R.utils)
library(foreach)
library(doSNOW)

# How many cores to use of the PC
corestouse=4

# Powdery Mildew or Brown Rust?
whatto="Mildew"


# Import of the dataframe "dataset" which contains the
    disease incidence, the climatic and the weather
    information
if(whatto=="Mildew"){load("./Daten/onedataset.RData")}else{
    load("./Daten/onedataset_brown_rust.RData")}

# Setting the seed of further random functions to create
    reproducible results
set.seed(9)

# Creating a normalization function for the input
    parameters
normalize=function(x){return((x-min(x))/(max(x)-min(x)))}

# Defining the dataset for the kNN method
dataset_kn=dataset

# It is necessary to define the vulnerability class as
    numeric for this procedure, although it is formally
    correct defined as factor
dataset_kn$Klasse=as.numeric(as.character(dataset_kn$Klasse
    ))

# Using the function to normalize the independent variables
     for the kNN method
n_dataset=as.data.frame(lapply(dataset_kn[3:(dim(dataset_kn
    )[2]-2)], normalize))
```

```
# Then manually created additional functions are
    implemented
source("NuetzlicheFunktionen.R")

# Following different combinations of calibration and
    valiations years are tested
# Potential years are selected (1996-2003 and 2005-2016)
potja=sort(as.numeric(as.character(unique(years(dataset$
    Datum)))))
# number of possible samples:
nCr = factorial(length(potja)) / (factorial(length(potja)-
    round((length(potja)/3)*2,0)) *factorial(round((length(
    potja)/3)*2,0)))
# One percent of the samples
oneperc=signif(nCr/400,1)   #=200 == 0.25 % of all possible
     combinations

# Creating a list of samples
arbsellist=list(sort(sample(potja, round((length(potja)/3)*
    2,0))))
  tz=2
  while (length(arbsellist)<oneperc){
   newarb=sort(sample(potja, round((length(potja)/3)*2,0)))
   if(!any(duplicated(list(arbsellist,newarb)))){
    arbsellist[[tz]]=newarb
    tz=tz+1
    }
  }



# A time measurement is used to predict the iterations time
starttime=Sys.time()

for(i in 1:length(arbsellist)){
# Dataframes for the results are created
bestab_kn=data.frame(Observed=NA, Predicted=NA, PredictedProb
    =NA, method="kn",study="standard")
bestab_dt=data.frame(Observed=NA, Predicted=NA, PredictedProb
```

```
    =NA, method="dt", study="standard")
bestab_dte=data.frame(Observed=NA, Predicted=NA,
    PredictedProb=NA, method="dte", study="standard")
bestab_rf=data.frame(Observed=NA, Predicted=NA, PredictedProb
    =NA, method="rf", study="standard")

# Dataframes for the LOOCV results
bestab_kn_l=data.frame(Location=NA, Year=NA, Observed=NA,
    Predicted=NA, PredictedProb=NA, method="kn", study="loocv")
bestab_dt_l=data.frame(Location=NA, Year=NA, Observed=NA,
    Predicted=NA, PredictedProb=NA, method="dt", study="loocv")
bestab_dte_l=data.frame(Location=NA, Year=NA, Observed=NA,
    Predicted=NA, PredictedProb=NA, method="dte", study="loocv"
    )
bestab_rf_l=data.frame(Location=NA, Year=NA, Observed=NA,
    Predicted=NA, PredictedProb=NA, method="rf", study="loocv")


# Dataframes for the test of location and year count
bestab_kn_ly=data.frame(Observed=NA, Predicted=NA,
    PredictedProb=NA, CountY=NA, CountLoc=NA, method="kn")
bestab_dt_ly=data.frame(Observed=NA, Predicted=NA,
    PredictedProb=NA, CountY=NA, CountLoc=NA, method="dt")
bestab_dte_ly=data.frame(Observed=NA, Predicted=NA,
    PredictedProb=NA, CountY=NA, CountLoc=NA, method="dte")
bestab_rf_ly=data.frame(Observed=NA, Predicted=NA,
    PredictedProb=NA, CountY=NA, CountLoc=NA, method="rf")

# A dataframe for the MeanDecreaseAccuracy is created
modig=data.frame(Names=sort(names(dataset)[c(3:(dim(dataset
    )[2]-2))]))


# Calibration (2/3) and validation (1/3) years are selected
    from random arbsellist
arbsel=arbsellist[[i]]
antarb=potja[!is.element(potja, arbsel)]

#####################################
#kNN-Prediction:
```

```
# Creating a training and a test dataset
knn_train=n_dataset[is.element(years(dataset_kn$Datum),
    arbsel),]
knn_test=n_dataset[is.element(years(dataset_kn$Datum),
    antarb),]

# Selecting the classified disease incidence values as
    labels
knn_train_labels=dataset_kn[is.element(years(dataset_kn$
    Datum),arbsel),(dim(dataset_kn)[2])]
knn_test_labels=dataset_kn[is.element(years(dataset_kn$
    Datum),antarb),(dim(dataset_kn)[2])]

# Searching for a useful k-value
kval=kv_fit(dataset_kn[is.element(years(dataset_kn$Datum),
    arbsel),],30)

# Predicting the test datasets values
knnprediction=knn(train = knn_train, test = knn_test, cl =
    knn_train_labels, k =kval,prob = T)
knnprediction2=attributes(knnprediction)
knnprediction3=data.frame(knnPred=knnprediction,knnPredProb
   =knnprediction2$prob)
knnprediction3$PredOne=ifelse(as.numeric(as.character(
    knnprediction3[,1]))==0,1-knnprediction3[,2],
    knnprediction3[,2])

bestab_kn2=data.frame(Observed=knn_test_labels,Predicted=
    knnprediction3$knnPred,PredictedProb=knnprediction3$
    PredOne,method="kn",study="standard")
bestab_kn=rbind(bestab_kn,bestab_kn2)

# Predicting kNN according to count of locations and years
print("Predicting kNN according to count of locations and
    years")
allposloc=unique(dataset_kn[is.element(years(dataset_kn$
    Datum),arbsel),"Standort"])
  for(g in 1:length(arbsel)){

  # Random sampling of years
  usedyea=sample(arbsel,g)
```

```
allposloc2=unique(dataset_kn[is.element(years(dataset_kn$
    Datum),usedyea),"Standort"])
for(p in 1:(length(allposloc2)-1)){
 # Random sampling of locations
 usedloc=sample(allposloc2,p)

 # Creating a training and a test dataset based on the
     random samples
 knn_train_ly=n_dataset[is.element(years(dataset_kn$Datum
     ),usedyea)&is.element(dataset_kn$Standort,usedloc),]
 knn_test_ly=n_dataset[is.element(years(dataset_kn$Datum)
     ,antarb),]

 # Selecting the classified disease incidence values as
     labels
 knn_train_labels_ly=dataset_kn[is.element(years(dataset_
     kn$Datum),usedyea)&is.element(dataset_kn$Standort,
     usedloc),(dim(dataset_kn)[2])]
 knn_test_labels_ly=dataset_kn[is.element(years(dataset_
     kn$Datum),antarb),(dim(dataset_kn)[2])]

 # Skip if no two classes in the prediction variable
 if(length(unique(knn_train_labels_ly))==2){

 # Searching for a useful k-value
 kval=kv_fit(dataset_kn[is.element(years(dataset_kn$Datum
     ),usedyea)&is.element(dataset_kn$Standort,usedloc)
     ,],30)

 # Predicting the test datasets values
 knnprediction=knn(train = knn_train_ly, test = knn_test_
     ly, cl = knn_train_labels_ly, k =kval,prob = T)
 knnprediction2=attributes(knnprediction)
 knnprediction3=data.frame(knnPred=knnprediction,
     knnPredProb=knnprediction2$prob)
 knnprediction3$PredOne=ifelse(as.numeric(as.character(
     knnprediction3[,1]))==0,1-knnprediction3[,2],
     knnprediction3[,2])
 bestab_kn2_ly=data.frame(Observed=knn_test_labels_ly,
     Predicted=knnprediction3$knnPred,PredictedProb=
     knnprediction3$PredOne,CountY=g,CountLoc=p,method="kn
```

```
      ")
    bestab_kn_ly=rbind(bestab_kn_ly,bestab_kn2_ly)
    }
   }
   }

# Introducing the LOOCV
test_loc=unique(dataset_kn[is.element(years(dataset_kn$
   Datum),antarb),1])
for(u in test_loc){
 # One location is erased from the training data and all
    other locations are erased from the test data
 knn_train2=n_dataset[is.element(years(dataset_kn$Datum),
    arbsel)&!is.element(dataset_kn$Standort,u),]
 knn_test2=n_dataset[is.element(years(dataset_kn$Datum),
    antarb)&is.element(dataset_kn$Standort,u),]
 knn_train_labels2=dataset_kn[is.element(years(dataset_kn$
    Datum),arbsel)&!is.element(dataset_kn$Standort,u),(dim(
    dataset_kn)[2])]
 knn_test_labels2=dataset_kn[is.element(years(dataset_kn$
    Datum),antarb)&is.element(dataset_kn$Standort,u),(dim(
    dataset_kn)[2])]
 testyears=as.numeric(as.character(years(dataset_kn[is.
    element(years(dataset_kn$Datum),antarb)&is.element(
    dataset_kn$Standort,u),2])))

 # Searching for a useful k-value
 kval=kv_fit(dataset_kn[is.element(years(dataset_kn$Datum),
    arbsel)&!is.element(dataset_kn$Standort,u),],30)

 # Predicting the test datasets values
 knnprediction=knn(train = knn_train2, test = knn_test2, cl
     = knn_train_labels2, k =kval,prob = T)
 knnprediction2=attributes(knnprediction)
 knnprediction3=data.frame(knnPred=knnprediction,
    knnPredProb=knnprediction2$prob)
 knnprediction3$PredOne=ifelse(as.numeric(as.character(
    knnprediction3[,1]))==0,1-knnprediction3[,2],
    knnprediction3[,2])
 bestab_kn2_l=data.frame(Location=u,Year=testyears,Observed
    =knn_test_labels2,Predicted=knnprediction3$knnPred,
```

```
    PredictedProb=knnprediction3$PredOne, method="kn", study=
    "loocv")
 bestab_kn_l=rbind(bestab_kn_l, bestab_kn2_l)
}


##################################
#Decision tree prediction:

# Creating a training and a test dataset
dt_train=dataset[is.element(years(dataset$Datum), arbsel), ]
dt_test=dataset[is.element(years(dataset$Datum), antarb), ]

# The decision tree model is fitted to the training dataset
model=C5.0(dt_train[,c(3:(dim(dataset)[2]-2))], as.factor(
   dt_train[,(dim(dt_train)[2])]),control = C5.0Control(
   minCases =round((dim(dt_train)[1]/100)*.7,0)))

# Predicting the test datasets values
bestab_dt2=data.frame(Observed=as.factor(dt_test$IBHBF),
   Predicted=predict(model, dt_test), PredictedProb=predict(
   model, dt_test, type="prob")[,2], method="dt", study="
   standard")
bestab_dt=rbind(bestab_dt, bestab_dt2)

##################################
#Boosted decision trees prediction:

# Evaluation of the optimum ensemble size
trials=evtrials(dt_train)

# The decision tree model is fitted to the training dataset
model_dte=C5.0(dt_train[,c(3:(dim(dataset)[2]-2))], as.
   factor(dt_train[,(dim(dt_train)[2])]),control = C5.0
   Control(minCases =round((dim(dt_train)[1]/100)*.7,0)),
   trials =trials)

# Predicting the test datasets values
bestab_dte2=data.frame(Observed=as.factor(dt_test$IBHBF),
   Predicted=predict(model_dte, dt_test), PredictedProb=
   predict(model_dte, dt_test, type="prob")[,2], method="dte"
```

```
    ,study="standard")
bestab_dte=rbind(bestab_dte,bestab_dte2)

# Predicting Decision trees and decision tree ensembles
    according to count of locations and years
print("Predicting Decision trees and decision tree
    ensembles according to count of locations and years")
allposloc=unique(dt_train$Standort)
 pb=txtProgressBar(min = 0, max = length(arbsel), style =
    3)
 for(g in 1:length(arbsel)){

  # Random sampling of years
  usedyea=sample(arbsel,g)
  allposloc2=unique(dt_train[is.element(years(dt_train$
     Datum),usedyea),"Standort"])

  cl=makeSOCKcluster(corestouse)
  registerDoSNOW(cl)
  on.exit(stopCluster(cl))
  bestab_dtb_ly2=foreach (p = 1:(length(allposloc2)-1),.
     packages=c('chron','randomForest','class','C50','caret
     ','ROCR'),.combine=rbind, .errorhandling = 'remove') %
     dopar% {

  #for(p in 1:(length(allposloc2)-1)){
   # Random sampling of locations
   usedloc=sample(allposloc2,p)
   # Creating a training and a test dataset based on the
      random samples
   dt_train_ly=dt_train[is.element(years(dt_train$Datum),
      usedyea)&is.element(dt_train$Standort,usedloc),]

   # Skip if no two classes in the prediction variable
   if(length(unique(dt_train_ly$IBHBF))==2){

   # Evaluation of the optimum ensemble size
   trials=evtrials(dt_train_ly)

   # The decision tree model is fitted to the training
      dataset
```

```
    model=C5.0(dt_train_ly[,c(3:(dim(dt_train_ly)[2]-2))],
        as.factor(dt_train_ly[,(dim(dt_train_ly)[2])]),
        control = C5.0Control(minCases =round((dim(dt_train_
        ly)[1]/100)*.7,0)))
    model_dte=C5.0(dt_train_ly[,c(3:(dim(dt_train_ly)[2]-2))
        ], as.factor(dt_train_ly[,(dim(dt_train_ly)[2])]),
        control = C5.0Control(minCases =round((dim(dt_train_
        ly)[1]/100)*.7,0)),
    trials =trials)

    # Predicting the test datasets values
    rbind(data.frame(Observed=as.factor(dt_test$IBHBF),
        Predicted=predict(model, dt_test),PredictedProb=
        predict(model, dt_test,type="prob")[,2],CountY=g,
        CountLoc=p,method="dt"),data.frame(Observed=as.factor
        (dt_test$IBHBF),Predicted=predict(model_dte, dt_test)
        ,PredictedProb=predict(model_dte, dt_test,type="prob"
        )[,2],CountY=g,CountLoc=p,method="dte"))
  }
  }
  stopCluster(cl)
  bestab_dt_ly2=bestab_dtb_ly2[which(as.character(bestab_
     dtb_ly2$method)=="dt"),]
  bestab_dt_ly=rbind(bestab_dt_ly,bestab_dt_ly2)
  bestab_dte_ly2=bestab_dtb_ly2[which(as.character(bestab_
     dtb_ly2$method)=="dte"),]
  bestab_dte_ly=rbind(bestab_dte_ly,bestab_dte_ly2)
  setTxtProgressBar(pb, g)
  }
  close(pb)


# Introducing the LOOCV
print("Predicting Decision trees LOOCV")
test_loc=unique(dt_test$Standort)

pb=txtProgressBar(min=1, max=length(test_loc), style=3)
progress=function(n) setTxtProgressBar(pb, n)
opts=list(progress=progress)
cl=makeSOCKcluster(corestouse)
registerDoSNOW(cl)
```

```
bestab_dtg2_l=foreach (m = 1:length(test_loc),.packages=c('
   chron','randomForest','class','C50','caret','ROCR'),.
   combine=rbind,.options.snow=opts, .errorhandling = '
   remove') %dopar% {
 u=test_loc[m]
 dt_train2=dt_train[!is.element(dt_train$Standort,u), ]
 dt_test2=dt_test[is.element(dt_test$Standort,u), ]

 # The decision tree model is fitted to the training
    dataset
 model=C5.0(dt_train2[,c(3:(dim(dt_train2)[2]-2))], as.
    factor(dt_train2[,(dim(dt_train2)[2])]),control = C5.0
    Control(minCases =round((dim(dt_train2)[1]/100)*.7,0)))

 # Predicting the test datasets values
 bestab_dt2_l=data.frame(Location=u,Year=as.numeric(as.
    character(years(dt_test2$Datum))),Observed=as.factor(dt
    _test2$IBHBF),Predicted=predict(model, dt_test2),
    PredictedProb=predict(model, dt_test2,type="prob")[,2],
    method="dt",study="loocv")
 #bestab_dt_l=rbind(bestab_dt_l,bestab_dt2_l)

 ###################################
 #Boosted decision trees prediction:

 # Evaluation of the optimum ensemble size
 trials=evtrials(dt_train2)

 # The decision tree model is fitted to the training
    dataset
 model_dte=C5.0(dt_train2[,c(3:(dim(dt_train2)[2]-2))], as.
    factor(dt_train2[,(dim(dt_train2)[2])]),control = C5.0
    Control(minCases =round((dim(dt_train2)[1]/100)*.7,0)),
    trials =trials)

 # Predicting the test datasets values
 bestab_dte2_l=data.frame(Location=u,Year=as.numeric(as.
    character(years(dt_test2$Datum))),Observed=as.factor(dt
    _test2$IBHBF),Predicted=predict(model_dte, dt_test2),
    PredictedProb=predict(model_dte, dt_test2,type="prob")
    [,2],method="dte",study="loocv")
```

```
 #bestab_dte_l=rbind(bestab_dte_l,bestab_dte2_l)
 rbind(bestab_dt2_l,bestab_dte2_l)
}
close(pb)
stopCluster(cl)
bestab_dt2_l=bestab_dtg2_l[which(as.character(bestab_dtg2_l
   $method)=="dt"),]
bestab_dt_l=rbind(bestab_dt_l,bestab_dt2_l)
bestab_dte2_l=bestab_dtg2_l[which(as.character(bestab_dtg2_
   l$method)=="dte"),]
bestab_dte_l=rbind(bestab_dte_l,bestab_dte2_l)


##################################
#Random forest prediction:

# Creating a training and a test dataset
rf_train=dataset[is.element(years(dataset$Datum),arbsel),]
rf_test=dataset[is.element(years(dataset$Datum),antarb),]

# First the optimum classwt parameters and the optimal mtry
    paramters are evaluated
mtrwt=evclassmtry(rf_train)
bestwt=mtrwt[2]
best=mtrwt[1]

# Then the Random Forest model itself can be created
  cl=makeSOCKcluster(corestouse)
  registerDoSNOW(cl)
model=foreach(ntree=rep((1000/max(which(((1000/c(1:
   corestouse))%%1)==0))),(max(which(((1000/c(1:corestouse
   ))%%1)==0)))),.combine=combine,.packages='
   randomForest') %dopar%
   randomForest(as.formula(paste("as.factor(",names(rf_
       train)[(dim(rf_train)[2])],")~",paste(names(rf_train)
       [c(3:(dim(rf_train)[2]-2))],collapse="+"))),data=rf_
       train, importance=TRUE, ntree=ntree,mtry=best,classwt
        = c(1,bestwt))
  stopCluster(cl)

#Getting the mean decrease accuracy
```

```
modi=as.data.frame(importance(model))
modig=cbind(modig,modi$MeanDecreaseAccuracy[order(rownames(
    modi))])

# Predicting the test datasets values
bestab_rf2=data.frame(Observed=as.factor(rf_test$IBHBF),
    Predicted=predict(model, rf_test),PredictedProb=predict(
    model, rf_test,type="prob")[,2],method="rf",study="
    standard")
bestab_rf=rbind(bestab_rf,bestab_rf2)

# Predicting Random Forests according to count of locations
    and years
print("Predicting Random Forests according to count of
    locations and years")
allposloc=unique(rf_train$Standort)
 pb=txtProgressBar(min = 0, max = length(arbsel), style =
    3)
 for(g in 1:length(arbsel)){
  # Random sampling of years
  usedyea=sample(arbsel,g)
  allposloc2=unique(rf_train[is.element(years(rf_train$
     Datum),usedyea),"Standort"])

  cl=makeSOCKcluster(corestouse)
  registerDoSNOW(cl)
  on.exit(stopCluster(cl))
  bestab_rf_ly2=foreach (p = 1:(length(allposloc2)-1),.
     packages=c('chron','randomForest','class','C50','caret
     ','ROCR','R.utils'),.combine=rbind, .errorhandling = '
     remove') %dopar% {

  tryCatch({
   # Random sampling of locations
   usedloc=sample(allposloc2,p)
   # Creating a training and a test dataset based on the
      random samples
   rf_train_ly=rf_train[is.element(years(rf_train$Datum),
      usedyea)&is.element(rf_train$Standort,usedloc),]

   # Skip if no two classes in the prediction variable
```

```
    if(length(unique(rf_train_ly$IBHBF))==2){

    # First the optimum classwt parameters and the optimal
        mtry paramters are evaluated
    mtrwt=evclassmtry(rf_train_ly)
    bestwt=mtrwt[2]
    best=mtrwt[1]

    # Then the application of the parameters on the
        RandomForest are tried out
    model=evalWithTimeout({try(randomForest(as.formula(paste
        ("as.factor(",names(rf_train_ly)[(dim(rf_train_ly)
        [2])],")~",paste(names(rf_train_ly)[c(3:(dim(rf_train
        _ly)[2]-2))],collapse="+"))),
    data=rf_train_ly, importance=TRUE, ntree=1000,mtry=best,
        classwt = c(1,bestwt)), silent=TRUE)},timeout=6000,
        onTimeout="warning")

    # Predicting the test datasets values
    return(data.frame(Observed=as.factor(rf_test$IBHBF),
        Predicted=predict(model, rf_test),PredictedProb=
        predict(model, rf_test,type="prob")[,2],CountY=g,
        CountLoc=p,method="rf"))
  }
  }, error=function(cond) {
            # If an error occurs an empty output is created
            return(data.frame(Observed=NA,Predicted=NA,
                PredictedProb=NA,CountY=p,CountLoc=g,method=
                "rf"))
        }
    )
  }
  stopCluster(cl)
  bestab_rf_ly=rbind(bestab_rf_ly,bestab_rf_ly2)
  setTxtProgressBar(pb, g)
  }
  close(pb)

# Introducing the LOOCV
print("Predicting Random Forests LOOCV")
test_loc=unique(rf_test$Standort)
```

```
pb=txtProgressBar(min=1, max=length(test_loc), style=3)
progress=function(n) setTxtProgressBar(pb, n)
opts=list(progress=progress)
cl=makeSOCKcluster(corestouse)
registerDoSNOW(cl)
bestab_rf2_l=foreach (m = 1:length(test_loc),.packages=c('
   chron','randomForest','class','C50','caret','ROCR','R.
   utils'),.combine=rbind,.options.snow=opts, .
   errorhandling = 'remove') %dopar% {
 u=test_loc[m]
 rf_train2=rf_train[!is.element(rf_train$Standort,u), ]
 rf_test2=rf_test[is.element(rf_test$Standort,u), ]

 # First the optimum classwt parameters and the optimal
    mtry paramters are evaluated
 mtrwt=evclassmtry(rf_train2)
 bestwt=mtrwt[2]
 best=mtrwt[1]

 # Then the application of the parameters on the
    RandomForest are tried out
 model=evalWithTimeout({try(randomForest(as.formula(paste("
    as.factor(",names(rf_train2)[(dim(rf_train2)[2])],")~",
    paste(names(rf_train2)[c(3:(dim(rf_train2)[2]-2))],
    collapse="+"))),
 data=rf_train2, importance=TRUE, ntree=1000,mtry=best,
    classwt = c(1,bestwt)), silent=TRUE)},timeout=6000,
    onTimeout="warning")

 # Predicting the test datasets values
 data.frame(Location=u,Year=as.numeric(as.character(years(
    rf_test2$Datum))),Observed=as.factor(rf_test2$IBHBF),
    Predicted=predict(model, rf_test2),PredictedProb=
    predict(model, rf_test2,type="prob")[,2],method="rf",
    study="loocv")
}
close(pb)
stopCluster(cl)
bestab_rf_l=rbind(bestab_rf_l,bestab_rf2_l)
```

```
print(i)

# The time measurement is used to predict the iterations
    time
nowtime=Sys.time()
remaintime=(((nowtime-starttime)/((i-1)+1))*(oneperc-i))
print(paste("Step taken:",nowtime))
print(paste("Finalised:",remaintime+Sys.time()))
print(paste("Finalised in:",round(as.numeric(remaintime,
    units = "days"),2),"days or",round(as.numeric(remaintime
    ,units = "hours"),2),"hours"))
flush.console()

save.image(paste0("./Datenausgabe/Pathogen_",whatto,"_
    Iteration_",i,".RData"))

rm(bestab_kn,bestab_dt,bestab_dte,bestab_rf,bestab_kn_l,
    bestab_dt_l,bestab_dte_l,bestab_rf_l,bestab_kn_ly,bestab
    _dt_ly,bestab_dte_ly,bestab_rf_ly)
gc(T)
}
```

B.6 Real time modelling of infestation risks in 2017

Data used First, a brief overview of the data used in this script is given:

As described in chapter B.4, the aggregated weather data were stored for each year. Therefore the file *"kwett2017.RData"* contains one raster stack for each day from October 2016 to September 2017. For example, the raster stack *kwett170530* contains the interpolated weather data for May the 30th 2017. The structure is displayed in figure B.2.

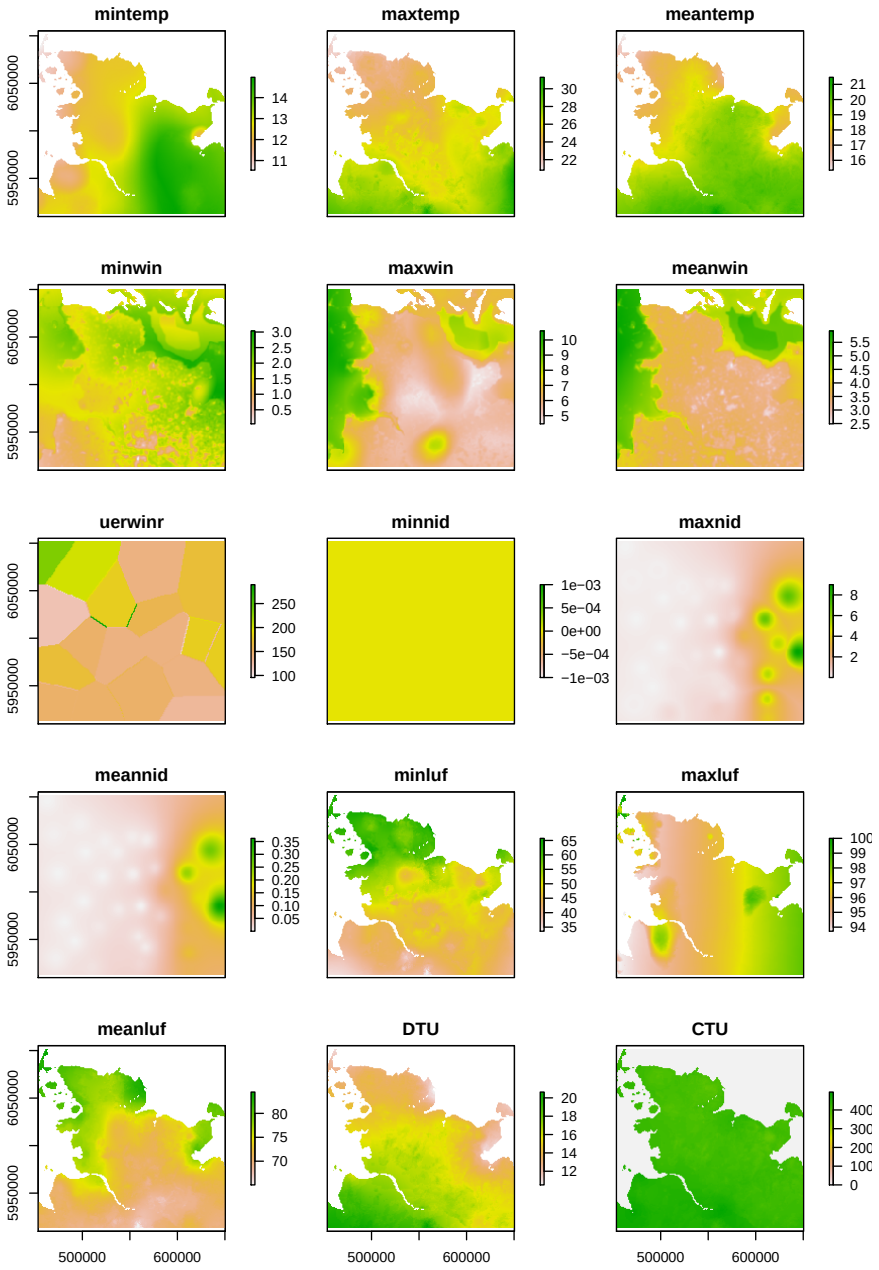


Figure B.2: kwett170530

```
## This script aims at the usuage of different
   classification methods to predict disease incidences
# Created 2017-08-16
# Created by: Wolfgang B. Hamer

library(chron)
# Including the package randomForest for the randomForest
   algorithms
library(randomForest)
# Including the package class for the knn algorithms
library("class")
# Including the package C50 for the decision tree
   algorithms
library(C50)
#Including packages for the result interpretation
library(ggplot2)
library(caret)
library(ROCR)
# Including packages for the parallel processing parts
library(R.utils)
library(foreach)
library(doSNOW)
# Including packages for the spatial statistics
library(gstat)
library(rgdal)
library(raster)
library(deldir)
library(automap)
library(sampSurf)
library(rgeos)
# Including a package for the colors of the plots
library(RColorBrewer)
library(xtable)

# Powdery Mildew or Brown Rust?
whattos="Mildew"
corestouse=4

# Preparating the files:

if(whattos=="Mildew"){load("../Daten/Powdery_mildew/
```

```
    onedataset.RData")}else{load("../Daten/Brown_rust/
    onedataset_brown_rust.RData") }

dataset=dataset[order(dataset$Datum),]

if(whattos=="Mildew"){
 load("../Daten/Powdery_mildew/Ritmo_Mehltau_aktuell.RData"
     )
 powa=Erysiphe_graminis_Ritmo_kon[which(substr(Erysiphe_
     graminis_Ritmo_kon$Datum,1,4)=="2017"),c(1,2,19)]
 powa$Klasse=5
 names(powa)[3]="IBHB"
 load("../Daten/Powdery_mildew/neben_ritmo_aktuell.RData")
}else{
 load("../Daten/Brown_rust/Ritmo_Braunrost_aktuell.RData")
 powa=Puccinia_recondita_Ritmo_kon[which(substr(Puccinia_
     recondita_Ritmo_kon$Datum,1,4)=="2017"),c(1,2,19)]
 powa$Klasse=8
 names(powa)[3]="IBHB"
 load("../Daten/Brown_rust/neben_ritmo_Braunrost_aktuell.
     RData")
}

powan=egra[which(substr(egra$Datum,1,4)=="2017"),c
    (20,19,18,21)]
names(powan)[3]="IBHB"
powab=rbind(powa,powan)
powab$IBHBF=0

if(whattos=="Mildew"){powab$IBHBF[which(powab$IBHB>=70)]=1}
    else{powab$IBHBF[which(powab$IBHB>=30)]=1}

##########################################################
#Climate Data
if(whattos=="Mildew"){
 load("../Daten/Powdery_mildew/kwett2017.RData")
 load("../Daten/Powdery_mildew/onedataset.RData")
}else{
 load("../Daten/Brown_rust/kwett2017.RData")
 load("../Daten/Brown_rust/onedataset_brown_rust.RData")
}
```

```
dataset=dataset[order(dataset$Datum),]

pos=read.csv2("../Daten/Standorte.csv")
names(pos)[1]="Standort"
coordinates(pos) = c("xcord", "ycord")
proj4string(pos) <- CRS(("+init=epsg:25832"))
grapo=pos@data
grapo=cbind(grapo,extract(gradienten,pos))

datespos=sort(unique(powab$Datum))
neudg=data.frame(Standort=NA,mintemp=NA,maxtemp=NA,meantemp
    =NA,minwin=NA,maxwin=NA,meanwin=NA,uerwinr=NA,minnid=NA,
    maxnid=NA,meannid=NA,minluf=NA,maxluf=NA,meanluf=NA,DTU=
    NA,CTU=NA,Datum=datespos[1])

for(i in 1:length(datespos)){
 fid=datespos[i]
 se=paste0("kwett",substr(fid,3,4),substr(fid,6,7),substr(
     fid,9,10))

 neud=cbind(pos@data,extract(get(se),pos))
 neud$Datum=fid
 neudg=rbind(neudg,neud)
}
neudg=neudg[-1,]

powab2=merge(powab,neudg,by=c("Standort","Datum"),all.x=
    TRUE)

powab2=merge(powab2,grapo,by=c("Standort"),all.x=TRUE)

powab3=powab2[,c(names(dataset))]

dataset2=rbind(dataset,powab3)

if(whattos=="Mildew"){
 saveRDS(dataset2,"../Daten/Powdery_mildew/all_data.rds")
}else{
 saveRDS(dataset2,"../Daten/Brown_rust/all_data_rust.rds")
}
```

```
#####################
#############################################################

datesposap=sort(ls(pattern="kwett1"))
neudg=data.frame(Standort=NA, mintemp=NA, maxtemp=NA, meantemp
    =NA, minwin=NA, maxwin=NA, meanwin=NA, uerwinr=NA, minnid=NA,
    maxnid=NA, meannid=NA, minluf=NA, maxluf=NA, meanluf=NA,DTU=
    NA,CTU=NA, Datum=datespos[1])

for(i in 1:length(datesposap)){
 se=datesposap[i]

 neud=cbind(pos@data,extract(get(se),pos))
 neud$Datum=as.Date(substr(se,6,11),"%y%m%d")
 neudg=rbind(neudg,neud)
print(paste(i,"of",length(datesposap)))
flush.console()
}
neudg=neudg[-1,]

allfpre=merge(neudg,grapo,by=c("Standort"),all.x=TRUE)

allfpre2=allfpre[,c(names(dataset)[is.element(names(dataset
    ),names(allfpre))])]


if(whattos=="Mildew"){
 saveRDS(allfpre2,"../Daten/Powdery_mildew/all_data_b.rds")
}else{
 saveRDS(allfpre2,"../Daten/Brown_rust/all_data_b_rust.rds"
     )
}

#####################
#############################################################
# Following the predictions of the machine learning
    techniques are created:

# Setting the seed of further random functions to create
    reproducible results
```

```
set.seed(9)

# Import of the dataframe "dataset" which contains the
    disease incidence, the climatic and the weather
    information
if(whattos=="Mildew"){load("../Daten/Powdery_mildew/
    kwett2017.RData")}else{load("../Daten/Brown_rust/
    kwett2017.RData") }
if(whattos=="Mildew"){dataset=readRDS("../Daten/Powdery_
    mildew/all_data.rds")}else{dataset=readRDS("../Daten/
    Brown_rust/all_data_rust.rds") }

# Creating a normalization function for the input
    parameters
normalize=function(x){return((x-min(x))/(max(x)-min(x)))}

# Defining the dataset for the kNN method
dataset_kn=dataset

# It is necessary to define the vulnerability class as
    numeric for this procedure, although it is formally
    correct defined as factor
dataset_kn$Klasse=as.numeric(as.character(dataset_kn$Klasse
    ))

# Using the function to normalize the independent variables
     for the kNN method
n_dataset=as.data.frame(lapply(dataset_kn[3:(dim(dataset_kn
    )[2]-2)], normalize))

# Then manually created additional functions are
    implemented
source("NuetzlicheFunktionen.R")

arbsel=c(1996:2016)
antarb=2017

#####################################
# First the kNN-predictions are created

# Creating a training and a test dataset
```

```
knn_train=n_dataset[is.element(years(dataset_kn$Datum),
    arbsel),]
knn_test=n_dataset[is.element(years(dataset_kn$Datum),
    antarb),]

# Selecting the classified disease incidence values as
    labels
knn_train_labels=dataset_kn[is.element(years(dataset_kn$
    Datum),arbsel),(dim(dataset_kn)[2])]
knn_test_labels=dataset_kn[is.element(years(dataset_kn$
    Datum),antarb),(dim(dataset_kn)[2])]

# Searching for a useful k-value
kval=kv_fit(dataset_kn[is.element(years(dataset_kn$Datum),
    arbsel),],30)

# Predicting the test datasets values
knnprediction=knn(train = knn_train, test = knn_test, cl =
    knn_train_labels, k =kval,prob = T)
knnprediction2=attributes(knnprediction)
knnprediction3=data.frame(knnPred=knnprediction,knnPredProb
   =knnprediction2$prob)
knnprediction3$PredOne=ifelse(as.numeric(as.character(
    knnprediction3[,1]))==0,1-knnprediction3[,2],
    knnprediction3[,2])

bestab_kn2=data.frame(Observed=knn_test_labels,Predicted=
    knnprediction3$knnPred,PredictedProb=knnprediction3$
    PredOne,method="kn",study="standard")

table(bestab_kn2$Observed,bestab_kn2$Predicted)

#####################################
# Then the decision tree predictions are created


dt_train=dataset[is.element(years(dataset$Datum),arbsel),]
dt_test=dataset[is.element(years(dataset$Datum),antarb),]

# The decision tree model is fitted to the training dataset
model_dt=C5.0(dt_train[,c(3:(dim(dataset)[2]-2))], as.
```

```
    factor(dt_train[,(dim(dt_train)[2])]),control = C5.0
    Control(minCases =round((dim(dt_train)[1]/100)*.7,0)))

# A plot of the model is created:
 dev.new(width=15, height=8)
 plot(model_dt)
 savePlot(filename =paste0("../Daten/export/",whattos,"_
    Decision_Tree.pdf"), type = c("pdf"), device =
    dev.cur())

# Predicting the test datasets values
bestab_dt2=data.frame(Observed=as.factor(dt_test$IBHBF),
    Predicted=predict(model_dt, dt_test),PredictedProb=
    predict(model_dt, dt_test,type="prob")[,2],method="dt",
    study="standard")

table(bestab_dt2$Observed,bestab_dt2$Predicted)

#####################################
# Then the boosted decision tree predictions are created

# Evaluation of the optimum ensemble size
trials=evtrials(dt_train)

# The decision tree model is fitted to the training dataset
model_dte=C5.0(dt_train[,c(3:(dim(dataset)[2]-2))], as.
    factor(dt_train[,(dim(dt_train)[2])]),control = C5.0
    Control(minCases =round((dim(dt_train)[1]/100)*.7,0)),
    trials =trials)

# Predicting the test datasets values
bestab_dte2=data.frame(Observed=as.factor(dt_test$IBHBF),
    Predicted=predict(model_dte, dt_test),PredictedProb=
    predict(model_dte, dt_test,type="prob")[,2],method="dte"
    ,study="standard")

table(bestab_dte2$Observed,bestab_dte2$Predicted)

#####################################
# Then the random forest predictions are created
```

```
rf_train=dataset[is.element(years(dataset$Datum),arbsel),]
rf_test=dataset[is.element(years(dataset$Datum),antarb),]

# First the optimum classwt parameters and the optimal mtry
    paramters are evaluated
mtrwt=evclassmtry(rf_train)
bestwt=mtrwt[2]
best=mtrwt[1]

# Then the Random Forest model itself can be created
  cl=makeSOCKcluster(corestouse)
  registerDoSNOW(cl)
model_rf=foreach(ntree=rep((1000/max(which(((1000/c(1:
    corestouse))%%1)==0))),(max(which(((1000/c(1:corestouse
    ))%%1)==0)))),.combine=combine,.packages='
    randomForest') %dopar%
    randomForest(as.formula(paste("as.factor(",names(rf_
        train)[(dim(rf_train)[2])],")~",paste(names(rf_train)
        [c(3:(dim(rf_train)[2]-2))],collapse="+"))),data=rf_
        train, importance=TRUE, ntree=ntree,mtry=best,classwt
         = c(1,bestwt))
  stopCluster(cl)

modi=as.data.frame(importance(model_rf))
modi=modi[order(modi$MeanDecreaseAccuracy),]


bestab_rf2=data.frame(Observed=as.factor(rf_test$IBHBF),
    Predicted=predict(model_rf, rf_test),PredictedProb=
    predict(model_rf, rf_test,type="prob")[,2],method="rf",
    study="standard")

table(bestab_rf2$Observed,bestab_rf2$Predicted)

if(whattos=="Mildew"){save.image("../Daten/export/pow2017.
    RData") }else{save.image("../Daten/export/rust2017.RData
    ") }


#
    ##########################################################################
```

```
#Following the predictions are summarized to achieved
    interpretable results:

###################################
# kNN-predictions:
bestab_kn=bestab_kn2
ergeb_kn=table(bestab_kn$Observed,bestab_kn$Predicted)

# Accuracy, sensitivity, specifity and precision of the
    model
# Accuracy: number of true predictions divided by the total
     number of predictions
accuracy_kn=(ergeb_kn[1,1]+ergeb_kn[2,2])/sum(ergeb_kn)
# Sensitivity: proportion of observed threshold exceedances
     classified correctly
sensitivity_kn=sensitivity(as.factor(bestab_kn$Predicted),
    as.factor(bestab_kn$Observed),positive="1")
# Specifity: proportion of observed threshold underruns
    classified correctly
specificity_kn=specificity(as.factor(bestab_kn$Predicted),
    as.factor(bestab_kn$Observed),negative="0")
# Precision: proportion of predicted threshold exceedances
    classified correctly
precision_kn=posPredValue(as.factor(bestab_kn$Predicted),as
    .factor(bestab_kn$Observed),positive="1")

# A prediction object is created which can be used to
    create th ROC curve and the ROC AUC
kn_pre=prediction(predictions=as.numeric(bestab_kn$
    PredictedProb),labels=bestab_kn$Observed)
kn_perf=performance(kn_pre,measure="tpr",x.measure="fpr")
kn_perfa=performance(kn_pre,measure="auc")
kn_auc=unlist(kn_perfa@y.values)


###################################
# Decision trees-predictions:
bestab_dt=bestab_dt2
# Cross table of all predictions and observations
ergeb_dt=table(bestab_dt$Observed,bestab_dt$Predicted)
```

```
# Accuracy, sensitivity, specifity and precision of the
    model
accuracy_dt=(ergeb_dt[1,1]+ergeb_dt[2,2])/sum(ergeb_dt)
sensitivity_dt=sensitivity(as.factor(bestab_dt$Predicted),
    as.factor(bestab_dt$Observed),positive="1")
specificity_dt=specificity(as.factor(bestab_dt$Predicted),
    as.factor(bestab_dt$Observed),negative="0")
precision_dt=posPredValue(as.factor(bestab_dt$Predicted),as
    .factor(bestab_dt$Observed),positive="1")

# A prediction object is created which can be used to
    create th ROC curve and the ROC AUC
dt_pre=prediction(predictions=as.numeric(bestab_dt$
    PredictedProb),labels=bestab_dt$Observed)
dt_perf=performance(dt_pre,measure="tpr",x.measure="fpr")
dt_perfa=performance(dt_pre,measure="auc")
dt_auc=unlist(dt_perfa@y.values)


#####################################
# Boosted decision trees-predictions:
bestab_dte=bestab_dte2
ergeb_dte=table(bestab_dte$Observed,bestab_dte$Predicted)

# Accuracy, sensitivity, specifity and precision of the
    model
accuracy_dte=(ergeb_dte[1,1]+ergeb_dte[2,2])/sum(ergeb_dte)
sensitivity_dte=sensitivity(as.factor(bestab_dte$Predicted)
    ,as.factor(bestab_dte$Observed),positive="1")
specificity_dte=specificity(as.factor(bestab_dte$Predicted)
    ,as.factor(bestab_dte$Observed),negative="0")
precision_dte=posPredValue(as.factor(bestab_dte$Predicted),
    as.factor(bestab_dte$Observed),positive="1")

# A prediction object is created which can be used to
    create th ROC curve and the ROC AUC
dte_pre=prediction(predictions=as.numeric(bestab_dte$
    PredictedProb),labels=bestab_dte$Observed)
dte_perf=performance(dte_pre,measure="tpr",x.measure="fpr")
dte_perfa=performance(dte_pre,measure="auc")
```

```
dte_auc=unlist(dte_perfa@y.values)


##################################
# Random Forest-predictions:
bestab_rf=bestab_rf2
# Cross table of all predictions and observations
ergeb_rf=table(bestab_rf$Observed,bestab_rf$Predicted)

# Accuracy, sensitivity, specifity and precision of the
    model
accuracy_rf=(ergeb_rf[1,1]+ergeb_rf[2,2])/sum(ergeb_rf)
sensitivity_rf=sensitivity(as.factor(bestab_rf$Predicted),
    as.factor(bestab_rf$Observed),positive="1")
specificity_rf=specificity(as.factor(bestab_rf$Predicted),
    as.factor(bestab_rf$Observed),negative="0")
precision_rf=posPredValue(as.factor(bestab_rf$Predicted),as
    .factor(bestab_rf$Observed),positive="1")

# A prediction object is created which can be used to
    create th ROC curve and the ROC AUC
rf_pre=prediction(predictions=as.numeric(bestab_rf$
    PredictedProb),labels=bestab_rf$Observed)
rf_perf=performance(rf_pre,measure="tpr",x.measure="fpr")
rf_perfa=performance(rf_pre,measure="auc")
rf_auc=unlist(rf_perfa@y.values)

# Mean decrease accuracy:
modig2=modi[order(modi$MeanDecreaseAccuracy),]
modig2$Names=rownames(modig2)
modig2$Names[which(modig2$Names=="Heissetage")]="Hot␣days␣(
    C)"
modig2$Names[which(modig2$Names=="MaxLufttemp")]="Max␣
    temperature␣(C)"
modig2$Names[which(modig2$Names=="minnid")]="Min␣
    precipitation"
modig2$Names[which(modig2$Names=="dgm100")]="Elevation"
modig2$Names[which(modig2$Names=="Bodentemperatur")]="Soil␣
    temperature␣(C)"
modig2$Names[which(modig2$Names=="MeanLufttemp")]="Mean␣
    temperature␣(C)"
```

```
modig2$Names[which(modig2$Names=="Frosttage")]="Frost days
    (C)"
modig2$Names[which(modig2$Names=="DroughtIndex")]="Drought
    Index (C)"
modig2$Names[which(modig2$Names=="MinLufttemp")]="Min
    temperature (C)"
modig2$Names[which(modig2$Names=="Windgeschwindigkeit")]="
    Wind speed (C)"
modig2$Names[which(modig2$Names=="RealEvap")]="Real
    Evaporation (C)"
modig2$Names[which(modig2$Names=="maxwin")]="Max wind speed
    "
modig2$Names[which(modig2$Names=="maxtemp")]="Max
    temperature"
modig2$Names[which(modig2$Names=="MonNiederschlag")]="
    Monthly precipitation (C)"
modig2$Names[which(modig2$Names=="maxnid")]="Max
    precipitation"
modig2$Names[which(modig2$Names=="meannid")]="Mean
    precipitation"
modig2$Names[which(modig2$Names=="minwin")]="Min wind speed
    "
modig2$Names[which(modig2$Names=="meantemp")]="Mean
    temperature"
modig2$Names[which(modig2$Names=="minluf")]="Min humidity"
modig2$Names[which(modig2$Names=="DTU")]="Daily thermal
    unit"
modig2$Names[which(modig2$Names=="meanwin")]="Mean wind
    speed"
modig2$Names[which(modig2$Names=="maxluf")]="Max humidity"
modig2$Names[which(modig2$Names=="meanluf")]="Mean humidity
    "
modig2$Names[which(modig2$Names=="mintemp")]="Min
    temperature"
modig2$Names[which(modig2$Names=="Klasse")]="Vulnerability
    class"
modig2$Names[which(modig2$Names=="CTU")]="Cumulative
    thermal unit"

#par(las=2)
dev.new(width=6, height=7)
```

```
par(las=2,mar=c(3,10,4,2))
barplot(log(modig2$MeanDecreaseAccuracy), main="Log., 
    average Mean Decrease Accuracy", horiz=TRUE,names.arg=
    modig2$Names,cex.names=0.9)
savePlot(filename =paste0("../Daten/export/
    MeanDecreaseAccuracy_",whattos,".pdf"), type = c("pdf
    "), device = dev.cur())

###################################
# Compareable results:

actres=data.frame(Method=c("kNN","Decision tree","Boosting 
    Decision tree","Random Forest"),Accuracy=c(accuracy_kn,
    accuracy_dt,accuracy_dte,accuracy_rf),
  Sensitivity=c(sensitivity_kn,sensitivity_dt,sensitivity_
      dte,sensitivity_rf),Specificity=c(specificity_kn,
      specificity_dt,specificity_dte,specificity_rf),
  Precision=c(precision_kn,precision_dt,precision_dte,
      precision_rf),AUC=c(kn_auc,dt_auc,dte_auc,rf_auc))

# Export to LATEX format
acx=xtable(actres)
print(acx, type="latex", file=paste0("../Daten/export/
    Actres_2017_",whattos,".tex"),include.rownames=FALSE)


###################################
# Preparation of spatial and temporal prediction:

datespos=sort(unique(dt_test$Datum))
# Creation of raster stacks
  all_r_dt=gradienten[[1]]
  all_r_dte=gradienten[[1]]
  all_r_rf=gradienten[[1]]
  all_r_knn=gradienten[[1]]


# For each day observations has been done, a spatial
    prediction is applied with each method
for(i in 1:length(datespos)){
  fid=datespos[i]
```

```
 se=paste0("kwett",substr(fid,3,4),substr(fid,6,7),substr(
    fid,9,10))
 Klasse=gradienten[[1]]
 if(whattos=="Mildew"){Klasse[]=rep(5,length(Klasse[]))}
    else{Klasse[]=rep(8,length(Klasse[]))}
 names(Klasse)="Klasse"
 neur=stack(get(se),gradienten,Klasse)
 neur_dt=predict(neur,model_dt,type='prob', index=1:2)[["
    layer.2"]]
 names(neur_dt)=as.character(fid)
all_r_dt=stack(all_r_dt,neur_dt)
 neur_dte=predict(neur,model_dte,type='prob', index=1:2)[["
    layer.2"]]
 names(neur_dte)=as.character(fid)
all_r_dte=stack(all_r_dte,neur_dte)
 neur_rf=predict(neur,model_rf,type='prob', index=1:2)[["
    layer.2"]]
 names(neur_rf)=as.character(fid)
all_r_rf=stack(all_r_rf,neur_rf)


 # knn prediction:
  neurd=as.data.frame(neur)
  neurd1=neurd[which(!is.na(apply(neurd,1,sum))),]
  neurd2=neurd1[,c(names(n_dataset))]
  neurd3=rbind(neurd2,dataset_kn[3:(dim(dataset_kn)[2]-2)])
  neurd4=as.data.frame(lapply(neurd3, normalize))
  neurd5=neurd4[c(1:dim(neurd2)[1]),]
  knnprediction=knn(train = knn_train, test = neurd5, cl =
     knn_train_labels, k =kval,prob = T)
  knnprediction2=attributes(knnprediction)
  knnprediction3=data.frame(knnPred=knnprediction,
     knnPredProb=knnprediction2$prob)
  knnprediction3$PredOne=ifelse(as.numeric(as.character(
     knnprediction3[,1]))==0,1-knnprediction3[,2],
     knnprediction3[,2])
  neur_knn=Klasse
  neur_knn[]=rep(NA,length(neur_knn[]))
 neur_knn[which(!is.na(apply(neurd,1,sum)))]=knnprediction3
    $PredOne
 names(neur_knn)=as.character(fid)
```

```
all_r_knn=stack(all_r_knn,neur_knn)

}
# The first raster of each stack, used to start the stacks
    is deleted
all_r_dt=all_r_dt[[-1]]
all_r_dte=all_r_dte[[-1]]
all_r_rf=all_r_rf[[-1]]
all_r_knn=all_r_knn[[-1]]


if(whattos=="Mildew"){save.image("../Daten/export/pow2017_
    sp.RData") }else{save.image("../Daten/export/rust2017_sp
    .RData") }


#####################################
# Temporal prediction and visualization:

# Colors for the temporal prediction
plotcol=brewer.pal(4,"Set1")
set.seed(9)

if(whattos=="Mildew"){load("../Daten/export/pow2017_sp.
    RData")}else{load("../Daten/export/rust2017_sp.RData") }
if(whattos=="Mildew"){dataset_all=readRDS("../Daten/Powdery
    _mildew/all_data_b.rds")}else{dataset_all=readRDS("../
    Daten/Brown_rust/all_data_b_rust.rds") }

# Subset of the data in the relevant time frame
dataset_all=dataset_all[which(dataset_all$Datum>="
    2017-04-17"&dataset_all$Datum<="2017-07-17"),]
dataset_all=dataset_all[order(dataset_all$Datum),]
dataset_allt=dataset_all

# Since only the Ritmo variety is predicted here the
    continuous weather data are connected with the pathogens
     susceptibility
if(whattos=="Mildew"){
 dataset_allt$Klasse=factor(5,levels=0:9)
 test1=dt_test[which(dt_test$Klasse==5),]
```

```
}else{
 dataset_allt$Klasse=factor(8,levels=0:9)
 test1=dt_test[which(dt_test$Klasse==8),]
 #Remove Dekan
 test1=test1[!is.element(test1$Datum,as.Date(c("01.05.2017"
    ,"30.05.2017","13.06.2017","27.06.2017","11.07.2017"),
    format="%d.%m.%Y")),]
}

# Here the models created before are used for the
    prediction of the continuous table
dataset_allt$dtpre=predict(model_dt, dataset_allt,type="
    prob")[,2]
dataset_allt$dtepre=predict(model_dte, dataset_allt,type="
    prob")[,2]
dataset_allt$rfpre=predict(model_rf, dataset_allt,type="
    prob")[,2]

# Only for the kNN-prediction no model can be created;
    therefore here the new prediction:
  neurd2=dataset_allt[,c(names(n_dataset))]
  neurd2$Klasse=as.numeric(as.character(neurd2$Klasse))
  neurd3=rbind(neurd2,dataset_kn[3:(dim(dataset_kn)[2]-2)])
  neurd4=as.data.frame(lapply(neurd3, normalize))
  neurd5=neurd4[c(1:dim(neurd2)[1]),]
  knnprediction=knn(train = knn_train, test = neurd5, cl =
     knn_train_labels, k =kval,prob = T)
  knnprediction2=attributes(knnprediction)
  knnprediction3=data.frame(knnPred=knnprediction,
     knnPredProb=knnprediction2$prob)
  knnprediction3$PredOne=ifelse(as.numeric(as.character(
     knnprediction3[,1]))==0,1-knnprediction3[,2],
     knnprediction3[,2])
dataset_allt$knnpre=knnprediction3$PredOne

# A vector of possible locations
posl=unique(dt_test$Standort)

for(r in 1:length(posl)){
# First a subset of the location
dataset_alltl=dataset_allt[which(dataset_allt$Standort==
```

```
    posl[r]),]
test=test1[which(test1$Standort==posl[r]),]

# If the predictions are wrong codes for the symbols in the
    plot are defined pch=(3,4,5,21)
dataset_alltl2=merge(dataset_alltl[,c(2,29:32)],test[,c
    (2,29,30)],by="Datum",all.x=T)
dataset_alltl2$dtprep=NA
 dataset_alltl2$dtprep[dataset_alltl2$dtpre>=.50&dataset_
    alltl2$IBHBF==0]=3
 dataset_alltl2$dtprep[dataset_alltl2$dtpre<.50&dataset_
    alltl2$IBHBF==1]=3
dataset_alltl2$dteprep=NA
 dataset_alltl2$dteprep[dataset_alltl2$dtepre>=.50&dataset_
    alltl2$IBHBF==0]=4
 dataset_alltl2$dteprep[dataset_alltl2$dtepre<.50&dataset_
    alltl2$IBHBF==1]=4
dataset_alltl2$rfprep=NA
 dataset_alltl2$rfprep[dataset_alltl2$rfpre>=.50&dataset_
    alltl2$IBHBF==0]=5
 dataset_alltl2$rfprep[dataset_alltl2$rfpre<.50&dataset_
    alltl2$IBHBF==1]=5
dataset_alltl2$knnprep=NA
 dataset_alltl2$knnprep[dataset_alltl2$knnpre>=.50&dataset_
    alltl2$IBHBF==0]=21
 dataset_alltl2$knnprep[dataset_alltl2$knnpre<.50&dataset_
    alltl2$IBHBF==1]=21



if(whattos=="Mildew"){
dev.new(width=30, height=16)
par(mar=c(9,6,3,5))
plot(dataset_alltl2$Datum,dataset_alltl2$dtpre,type="l",
    ylim=c(0,1),lwd=2, yaxt="n",ylab="Probability␣of␣BHB␣>␣
    70␣%",xlab="",xlim=c(dataset_alltl2$Datum[1],dataset_
    alltl2$Datum[dim(dataset_alltl2)[1]]),cex.lab=1.9,main=
    paste(dataset_alltl$Standort[1],substr(dataset_alltl2$
    Datum[1],1,4)),cex.main=2,cex.axis=1.5,cex=1.5,col=
    plotcol[2],lty=2)
lines(dataset_alltl2$Datum,dataset_alltl2$dtepre,type="l",
```

```
    lwd=2,col=plotcol[3],lty=6)
lines(dataset_alltl2$Datum,dataset_alltl2$rfpre,type="l",
    lwd=2,col=plotcol[4],lty=5)
lines(dataset_alltl2$Datum,dataset_alltl2$knnpre,type="l",
    lwd=2,col=plotcol[1],lty=4)

points(dataset_alltl2$Datum,dataset_alltl2$dtpre,type="p",
    col=plotcol[2],pch=dataset_alltl2$dtprep,cex=2,lwd=2)
points(dataset_alltl2$Datum,dataset_alltl2$dtepre,type="p",
    col=plotcol[3],pch=dataset_alltl2$dteprep,cex=2,lwd=2)
points(dataset_alltl2$Datum,dataset_alltl2$rfpre,type="p",
    col=plotcol[4],pch=dataset_alltl2$rfprep,cex=2,lwd=2)
points(dataset_alltl2$Datum,dataset_alltl2$knnpre,type="p",
    col=plotcol[1],pch=dataset_alltl2$knnprep,cex=2,lwd=2)


axis(side=2, at = c(0,.25,.50,.75,1),cex.axis=1.2)


par(new=TRUE)
plot(dataset_alltl2$Datum[!is.na(dataset_alltl2$IBHB)],
    dataset_alltl2$IBHB[!is.na(dataset_alltl2$IBHB)],type="p
    ",ylab="",ylim=c(0,(100/50)*70),yaxt="n", xaxt="n",lwd
    =2,col="blue",xlab="",xlim=c(dataset_alltl2$Datum[1],
    dataset_alltl2$Datum[dim(dataset_alltl2)[1]]),cex=1.2,
    pch=19)

axis(side=4, at = c(0,30,50,70,100),col.axis="blue",col="
    blue",cex.axis=1.2)
mtext("BHB [%]", side=4, line=3,cex=1.9,col="blue")
abline(h=70,lty=2,col="black")

legend("bottom", xpd = TRUE, horiz = TRUE, inset = c
    (0,-.18), c("Observed","kNN","Decision Tree","Boosted DT
    ","Random Forest"),col=c("blue",plotcol[1],plotcol[2],
    plotcol[3],plotcol[4]),cex=1.8,lty=c(NA,4,2,6,5),lwd=c(
    NA,2,2,2,2),pch=c(19,NA,NA,NA,NA),bg = "white", bty="n")
legend("bottom", xpd = TRUE, horiz = TRUE, inset = c
    (0,-.25), c("Wrong predictions:",paste0("kNN (",length(
    which(!is.na(dataset_alltl2$knnprep))),"x)"),
paste0("Decision Tree (",length(which(!is.na(dataset_alltl2
```

```
   $dtprep))),"x)"),
paste0("Boosted␣DT␣(",length(which(!is.na(dataset_alltl2$
   dteprep))),"x)"),
paste0("Random␣Forest␣(",length(which(!is.na(dataset_alltl2
   $rfprep))),"x)")),col=c(NA,plotcol[1],plotcol[2],plotcol
   [3],plotcol[4]),cex=1.8,pch=c(NA,21,3,4,5),bg = "white",
    bty="n")

savePlot(filename =paste0("../Daten/export/temp/",whattos,"
   _",posl[r],".pdf"), type = c("pdf"), device = dev.
   cur())


}else{
dev.new(width=30, height=16)
par(mar=c(9,6,3,5))
plot(dataset_alltl2$Datum,dataset_alltl2$dtpre,type="l",
   ylim=c(0,(100/30)*.5),lwd=2, yaxt="n",ylab="Probability␣
   of␣BHB␣>␣30␣%",xlab="",xlim=c(dataset_alltl2$Datum[1],
   dataset_alltl2$Datum[dim(dataset_alltl2)[1]]),cex.lab
   =1.9,main=paste(dataset_alltl$Standort[1],substr(dataset
   _alltl2$Datum[1],1,4)),cex.main=2,cex.axis=1.5,cex=1.5,
   col=plotcol[2],lty=2)
lines(dataset_alltl2$Datum,dataset_alltl2$dtepre,type="l",
   lwd=2,col=plotcol[3],lty=6)
lines(dataset_alltl2$Datum,dataset_alltl2$rfpre,type="l",
   lwd=2,col=plotcol[4],lty=5)
lines(dataset_alltl2$Datum,dataset_alltl2$knnpre,type="l",
   lwd=2,col=plotcol[1],lty=4)

points(dataset_alltl2$Datum,dataset_alltl2$dtpre,type="p",
   col=plotcol[2],pch=dataset_alltl2$dtprep,cex=2,lwd=2)
points(dataset_alltl2$Datum,dataset_alltl2$dtepre,type="p",
   col=plotcol[3],pch=dataset_alltl2$dteprep,cex=2,lwd=2)
points(dataset_alltl2$Datum,dataset_alltl2$rfpre,type="p",
   col=plotcol[4],pch=dataset_alltl2$rfprep,cex=2,lwd=2)
points(dataset_alltl2$Datum,dataset_alltl2$knnpre,type="p",
   col=plotcol[1],pch=dataset_alltl2$knnprep,cex=2,lwd=2)

axis(side=2, at = c(0,.25,.50,.75,1),cex.axis=1.2)
```

```
par(new=TRUE)
plot(dataset_alltl2$Datum[!is.na(dataset_alltl2$IBHB)],
    dataset_alltl2$IBHB[!is.na(dataset_alltl2$IBHB)],type="p
    ",ylab="",ylim=c(0,100),yaxt="n", xaxt="n",lwd=2,col="
    blue",xlab="",xlim=c(dataset_alltl2$Datum[1],dataset_
    alltl2$Datum[dim(dataset_alltl2)[1]]),cex=1.2,pch=19)

axis(side=4, at = c(0,30,50,70,100),col.axis="blue",col="
    blue",cex.axis=1.2)
mtext("BHB [%]", side=4, line=3,cex=1.9,col="blue")
abline(h=30,lty=2,col="black")

legend("bottom", xpd = TRUE, horiz = TRUE, inset = c
    (0,-.18), c("Observed","kNN","Decision Tree","Boosted DT
    ","Random Forest"),col=c("blue",plotcol[1],plotcol[2],
    plotcol[3],plotcol[4]),cex=1.8,lty=c(NA,4,2,6,5),lwd=c(
    NA,2,2,2,2),pch=c(19,NA,NA,NA,NA),bg = "white", bty="n")
legend("bottom", xpd = TRUE, horiz = TRUE, inset = c
    (0,-.25), c("Wrong predictions:",paste0("kNN (",length(
    which(!is.na(dataset_alltl2$knnprep))),"x)"),
paste0("Decision Tree (",length(which(!is.na(dataset_alltl2
    $dtprep))),"x)"),
paste0("Boosted DT (",length(which(!is.na(dataset_alltl2$
    dteprep))),"x)"),
paste0("Random Forest (",length(which(!is.na(dataset_alltl2
    $rfprep))),"x)")),col=c(NA,plotcol[1],plotcol[2],plotcol
    [3],plotcol[4]),cex=1.8,pch=c(NA,21,3,4,5),bg = "white",
     bty="n")

savePlot(filename =paste0("../Daten/export/temp/",whattos,"
    _",posl[r],".pdf"), type = c("pdf"), device = dev.
    cur())
}
}
graphics.off()

#####################################
# Spatial prediction and visualization:
if(whattos=="Mildew"){load("../Daten/export/pow2017_sp.
    RData")}else{load("../Daten/export/rust2017_sp.RData") }
```

```
if(whattos=="Mildew"){test1=dt_test[which(dt_test$Klasse
   ==5),]}else{test1=dt_test[which(dt_test$Klasse==8),] }

pos=read.csv2("../Daten/Standorte.csv")
names(pos)[1]="Standort"

test2=merge(test1,pos,by=c("Standort"),all.x=T)
test2=test2[,c(1,2,30:32)]


coordinates(pos) = c("xcord", "ycord")
proj4string(pos) <- CRS(("+init=epsg:25832"))


allrit=c("24.04.","02.05.","08.05.","15.05.","22.05.","
   29.05.","06.06.","12.06.","19.06.","26.06.","03.07.","
   10.07.")

prew=c("all_r_knn","all_r_dt","all_r_dte","all_r_rf")
prew2=c("knn","dt","dte","rf")

for(o in 1:length(prew)){
spacer=get(prew[o])

dev.new(width=7, height=9.4)
par(mfrow=c(4,3),mar=c(0,0,2,0),oma=c(6,2,2,2))
for(i in 1:16){
 if(is.element(paste0(substr(names(spacer)[i],10,11),".",
    substr(names(spacer)[i],7,9)),allrit)){
 datte=paste0(substr(names(spacer[[i]]),2,5),"-",substr(
    names(spacer[[i]]),7,8),"-",substr(names(spacer[[i]])
    ,10,11))
  plot(spacer[[i]],main=datte,
  legend=FALSE,axes=FALSE,col=rev(brewer.pal(4,'PRGn')),
     breaks=c(0,.25,.5,.75,1),cex.main=1.7)
 testa=test2[which(test2$Datum==datte),]
  coordinates(testa) = c("xcord", "ycord")
  proj4string(testa) <- CRS(("+init=epsg:25832"))
  testa$obs=extract(spacer[[i]],testa)
  testa$pc=19
  testa$pc[testa$obs<.50&testa$IBHBF==0]=19
```

```
    testa$pc[testa$obs>=.50&testa$IBHBF==0]=4
    testa$pc[testa$obs<.50&testa$IBHBF==1]=4
    points(testa,pch=testa$pc,cex=2,lwd=2)
  }
}
legend(x=-15000, y=5920000, legend=c("0 - 0.25","0.25 - 0.5
    ","0.5 - 0.75","0.75 - 1"), fill=rev(brewer.pal(4,'PRGn'
    )), ncol=1,
  xpd=NA, bty="n",cex=2.2,horiz = TRUE)
legend(x=70000, y=5885000, legend=c("Correct prediction","
    Wrong prediction"), pch=c(19,4), ncol=1,
  xpd=NA, bty="n",cex=2.2,horiz = TRUE)
savePlot(filename =paste0("../Daten/export/maps/",whattos,"
    _",prew2[o],".pdf"), type = c("pdf"), device = dev
    .cur())
}

graphics.off()
```

Wolfgang B. Hamer

Curriculum Vitae

Blocksberg 8
24103 Kiel
Deutschland
☎ *+49 (431) 880 2955*
✉ *hamer@geographie.uni-kiel.de*
www.lgi.geographie.uni-kiel.de/de/team/wolfgang_hamer

Education

1998–2007 **Abitur**, *Emil-von-Behring-Gymnasium*, Großhansdorf, *2,5*.

2008–2012 **Bachelor of Science**, *Christian-Albrechts-Universität zu Kiel*, *1,9*.
Geographie / *Geography*

2011–2014 **Master of Science**, *Christian-Albrechts-Universität zu Kiel*, *1,3*.
Umweltgeographie und -management / *Environmental geography and management*

Scientific career

2010–2014 **Chair for Landscape Ecology and Geoinformation**, *CAU*, Kiel.
Student Assistant

Since 2014 **Chair for Landscape Ecology and Geoinformation**, *CAU*, Kiel.
PhD student

Since 2016 **Chair for Landscape Ecology and Geoinformation**, *CAU*, Kiel.
Research Assistant

Publications

2016 Hamer, W.B., J.-A. Verreet and R. Duttmann: *Räumliche und zeitliche Vorhersage der Eintrittswahrscheinlichkeit eines ertragsgefährdenden Mehltauereignisses an Winterweizen mit der Random-Forest-Methode.* In: AGIT – Journal für Angewandte Geoinformatik, 2016 (2).

2016 Hamer, W.B., J.-A. Verreet and R. Duttmann: *Spatial prediction of the infestation risks of winter wheat by the pathogen Blumeria graminis f. sp. tritici (powdery mildew) in Schleswig-Holstein using semi-empirical and machine learning techniques.* In: GIS.Science, 29 (4), S. 140-148.

Award and scholarship

2012 Annual prize of the Verein für die Förderung der Math.-Nat. Fakultät der Christian-Albrechts-Universität zu Kiel e.V.

2014–2016 Scholarship of the Stiftung Schleswig-Holsteinische Landschaft

Presentations

2015 Hamer, W.B., and R. Duttmann: *Geo-räumliche Prognose der Befallsrisiken von Winterweizen durch das Weizenpathogen Blumeria graminis (Echter Mehltau)* At: Deutscher Kongress für Geographie in Berlin, October 1-4, 2015.

2016 Hamer W.B., J.-A. Verreet and R. Duttmann: *Flächendifferenzierte Vorhersage von Mehltau-Infektionsereignissen an Winterweizen mittels Random Forest Modellierung in Schleswig-Holstein* At: Workshop zur Simulation in den Umwelt- und Geowissenschaften in Hamburg, April 20-22, 2016.

2016 Hamer W.B., J.-A. Verreet and R. Duttmann: *Räumliche und zeitliche Vorhersage der Eintrittswahrscheinlichkeit eines ertragsgefährdenden Mehltauereignisses an Winterweizen mit der Random-Forest-Methode* At: AGIT 2016 - Symposium und Fachmesse Angewandte Geoinformatik in Salzburg, July 6-8, 2016.

2016 Hamer, W.B., J.A. Verreet, H. Klink, T. Birr and R. Duttmann: *Anwendung einer Random Forest Modellierung zur räumlichen und zeitlichen Vorhersage der Wahrscheinlichkeit ertragsrelevanter Befallsereignisse mit Blumeria graminis f. sp.tritici in Schleswig-Holstein* At: 60. Deutsche Pflanzenschutztagung in Halle (Saale), September 20-23, 2016.

2017 Hamer, W.B., J.A. Verreet, H. Klink, T. Birr and R. Duttmann: *Random Forest Modellierung zur witterungsbasierten Vorhersage der Wahrscheinlichkeit ertragsrelevanter Befallsereignisse an Winterweizen in Schleswig-Holstein* At: 30. Tagung des DPG-Arbeitskreises "Krankheiten in Getreide und Mais" in Braunschweig, January 30-31, 2017.

2017 Hamer, W.B., D. Knitter, O. Nakoinz and R. Duttmann: *Plans on an agent based model approach on prehistoric scale* At: Socio-Environmental Dynamics over the Last 12,000 Years: The Creation of Landscapes in Kiel, March 20-24, 2017.

2018 Hamer, W.B., J.A. Verreet, H. Klink, T. Birr and R. Duttmann: *Vergleich maschineller Lernverfahren zur räumlichen und zeitlichen Vorhersage ertragsrelevanter Befallsereignisse windbürtiger Weizenpathogene in Schleswig-Holstein* At: 31. Tagung des DPG-Arbeitskreises "Krankheiten in Getreide und Mais" in Braunschweig, January 29-30, 2018.

Teaching

11/2014 07/2015 03/2016 *GIS for Marine Sciences* For: Integrated School of Ocean Sciences, CAU Kiel.

SS 2016 SS 2017 *Modellierung von Landschaftsprozessen* For: Master of Science Umweltgeographie und -management, CAU Kiel.

SS 2016 SS 2017 *Geostatistik* For: Master of Science Umweltgeographie und -management, CAU Kiel.

WT 2016/17 WT 2017/18 *Geographische Informationssysteme II* For: Bachelor of Science Geographie, CAU Kiel.

Erklärung des Autors

Hiermit erkläre ich, dass die vorliegende Abhandlung - abgesehen von der Beratung durch die Betreuer - nach Inhalt und Form meine eigene Arbeit ist. Ebenso war diese Arbeit weder ganz noch in Teilen schon an anderer Stelle Gegenstand eines Prüfungsverfahrens und ist nicht veröffentlicht oder zur Veröffentlichung eingereicht worden. Ich habe diese Arbeit zudem unter Einhaltung der Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft verfasst.

Kiel, 5.6.2018

Wolfgang Berengar Hamer