UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR TECHNISCHE INFORMATIK

An Approach to Classify Plant Electrophysiological Responses to Environmental Stimuli

Ein Ansatz zur Klassifizierung elektrophysiologischer Reaktionen von Pflanzen auf Umweltreize

Bachelorarbeit

verfasst am
Institut für Technische Informatik

im Rahmen des Studiengangs
Robotik und autonome Systeme
der Universität zu Lübeck

vorgelegt von
Tim-Lucas Rabbel

ausgegeben und betreut von
Prof. Dr.-Ing. Heiko Hamann

mit Unterstützung von
Dr.-Ing. Mostafa Wahby

Lübeck, den 3. Dezember 2021

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Tim-Lucas Rabbel

Zusammenfassung

Pflanzen erzeugen elektrische Signale als Reaktion auf sich ändernde Umweltbedingungen. Wir verwenden Pflanzen in Kombination mit elektronischen Komponenten als biohybride Sensoren, indem wir die elektrischen Signale in ihren Stängeln messen und diese auf die sie auslösenden Umweltreize zurückführen. Durch Hinzufügen eines Edge-Devices zur Datenerfassung von mehreren biohybriden Sensoren konstruieren wir ein lebendiges Sensornetzwerk. Dieses wird in einer Laborumgebung verwendet, um die Realisierbarkeit der Nutzung von Pflanzen zur Umweltbeobachtung zu untersuchen. Die Pflanzen werden abwechselnd roten und blauen Lichtreizen ausgesetzt. Aus den gemessenen Signalen berechnen wir elf statistische Merkmale, mit denen wir mithile verschiedener Varianten der Diskriminanzanalyse den angewandten Stimulus klassifizieren. Wir erhalten eine maximale Korrektklassifikationsrate von 75, 56% mit univariater und 89, 63% mit multivariater Klassifizierung. Das lebendige Sensornetzwerk wird in Zukunft für die weitere Erforschung der Elektrophysiologie von Pflanzen verwendet.

Abstract

Plants produce electrical signals as a response to changing environmental conditions. We utilize plants in combination with electronics as biohybrid sensors by measuring the electrical responses in their stems and assigning them to the triggering environmental stimuli. Through the addition of a centralized edge device for data collection from multiple biohybrid sensor nodes, we construct a living sensor network. The developed network is used inside a lab environment to investigate the feasibility of utilizing plants for environmental monitoring. We expose plants alternately to blue and red light stimuli and measure their responses. After extracting eleven statistical features from the electrical signal, we perform different discriminant analysis variants to classify the applied stimuli. We obtain a best classification accuracy of 75.56% with univariate classification and 89.63% with multivariate classification. The living sensor network is used in the future for further research on plant electrophysiology.

Acknowledgements

First and foremost, I want to thank my supervisors Dr. Mostafa Wahby and Prof. Heiko Hamann for their assistance and support in my thesis. Through many meetings during the last few months, they provided me with valuable advice and taught me how to properly formulate my research results.

Since I wrote this thesis in the context of an international project, other researchers worked with me together. I want to thank Eddy and Till here in Lübeck for their help in preparing my experiments and reviewing my code. Thanks should also go to our collaboration partners in Zagreb, especially to Marko for testing and contributing to the living sensor network.

Writing a thesis means reviewing and reworking, and this would not have been possible without the assistance of colleagues and friends. I very much appreciated the help and expertise of Niclas Bockelmann from the Institute for Robotics and Cognitive Systems and Prof. Karsten Keller from the Institute of Mathematics for guidance in their respective research fields.

I want to thank my friends Pia, Tavia, Nanna, and Kim for reading my thesis and providing me with comprehensive comments and suggestions for improvement.

Most importantly, none of this could have happened without my parents. They provided me with unfailing support and continuous encouragement, not only for my thesis but throughout all my years of study. I am extremely grateful for having them in my life.

Contents

1

Introduction

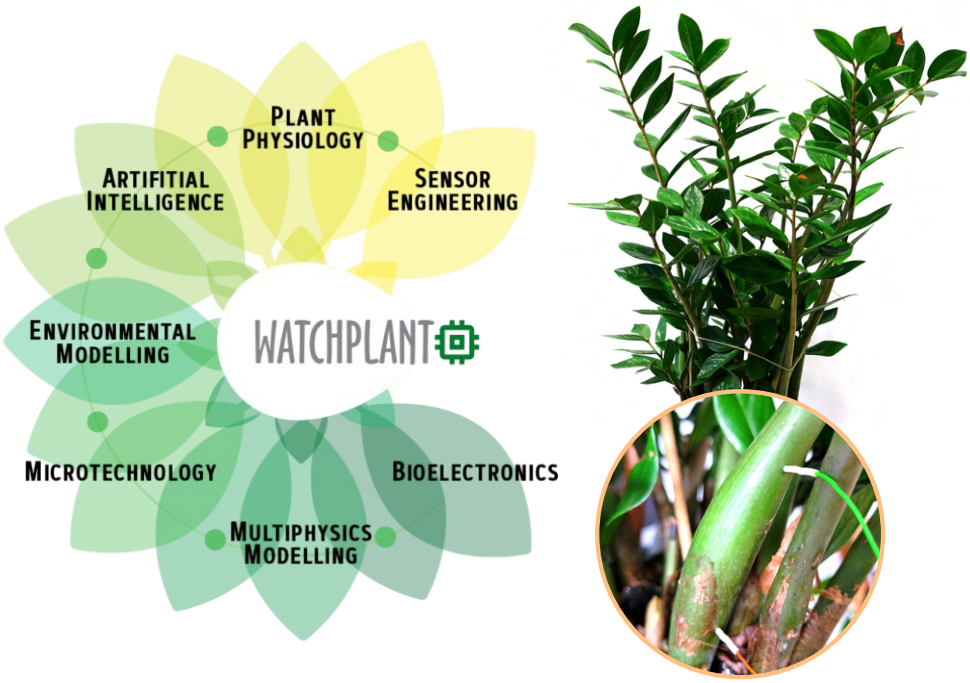

Figure 1.1: The WatchPlant project develops biohybrid sensors on the basis of plants (so-called phytosensors) for environmental monitoring. Depicted is a Zamioculcas plant with two electrodes attached to its stem for measuring the electrical signal inside of the plant. WatchPlant logo is taken from the project website [1].

Plants are sensitive organisms capable of sensing different stimuli like temperature, water balance, and light direction, and can, to some degree, adjust themselves to these stimuli. A common observable example can be found in phototropism: The plant grows

towards the light, not only upwards in the direction of the sun, but also around obstacles [2]. Reactions of plants, be they easily visible like phototropism or internal, for us not directly apparent processes, correlate with intracellular and intercellular electrical signals [3]. By measuring and assigning these signals to their respective triggering stimuli, we can obtain information about the surrounding environmental conditions. Phytosensing, that is, using plants as sensors, opens the way for exciting opportunities in environmental or plant health monitoring. The EU-funded project *WatchPlant* [1] will develop state-of-the-art wireless sensor nodes that measure the physiological plant responses to changing environmental conditions (see Figure 1.1 for an overview about main research fields of WatchPlant). A machine learning approach will be used to analyze the data and use the plants as an in-situ monitoring system of urban environments [4].

The newly developed sensors utilize clean energy sources like solar panels, enzymatic biofuel cells, or even the plant's own sap. Furthermore, the devices will be constructed with longevity and durability in mind for use in long-term studies and analysis. They will communicate wirelessly and form an ad-hoc network so that the communication between the sensor nodes is resilient, even if some of them get damaged or relocated. The so-created living sensor network can monitor the environment in a self-organizing manner for a long period, thus allowing the research of pollution or climate change, even in challenging and uncertain environments [4, 5, 6]. This technology has further potential uses in agriculture for crop health monitoring so that the harvest can be maximized, or in forestry. WatchPlant may also contribute to fundamental plant research by investigating internal plant processes that produce electrical or otherwise measurable responses. The exploitation of phloem sap for energy harvesting is a novel concept that WatchPlant will research.

1.1 Contributions of this Thesis

In this thesis, we want to develop a *living sensor network* to run experiments. An experiment in this context means measuring the electrical signal of the networks' living part, a plant, which we expose to different stimuli. In addition to collecting the electrical signals, we also collect the environmental information with conventional sensors. The network must, therefore, reliably collect and store data. To not influence the experiments, we will limit the presence of irrelevant stimuli by not entering the lab during their execution, so the control of the measurements has to be done wirelessly from outside the lab. *Biohybrid sensor nodes*, a combination of plants and electronics, are placed inside the lab and send their data to an *edge device*, which serves as centralized data storage. To still surveil the running experiment, we implement a live plotting script that can show immediate plant responses to, e.g., touch or light. We use the described setup to conduct experiments with blue and red light as two different stimuli. They are applied alternately every three hours for ten minutes. We collect datasets with a total of 296 stimulus applications, 148 each for red and blue light.

Chatterjee et al. introduced a classification approach for the stimuli ozone, sulfuric acid, and sodium chloride [7]. We test the presented approach on the collected red and blue

light datasets. Eleven statistical features are calculated from the raw datasets, which are used for classification with five variants of discriminant analysis. We tested univariate and multivariate classification. The best performing dataset yields a univariate accuracy of 75.56% using quadratic discriminant analysis (QDA) with the feature power spectral density. The same dataset reaches a multivariate accuracy of 89.63% using again QDA and eight of eleven features. However, we observed unexplainable responses and trends in the measurements.

1.2 Structure of this Thesis

Apart from the introduction, this thesis is organized into five further chapters. Chapter 2 presents the state of the art of biohybrid systems and provides an overview of traditional and recent environmental monitoring approaches. We also cover the mathematical fundamentals for classification as used in this thesis, which include feature extraction, five different discriminant analysis methods, and a short outlook on machine learning techniques. Chapter 3 describes the living sensor network, which was developed and used for this thesis. This includes the introduction of the used plants, the hardware and software of the biohybrid sensor nodes, the communication with the edge device, and the live plotting of data from currently running experiments. We highlight challenges we encountered while trying to integrate plants into the sensor node and discuss possible solutions. Chapter 4 first covers the collection of datasets from red and blue light stimuli, using the living sensor network developed in Chapter 3. We use the collected data to test the classification approach by Chatterjee et al. [7]. For that purpose, we use feature extraction followed by discriminant analysis, using the methods as described in Chapter 2. We also discuss unexpected plant behavior we noticed during experiments. Chapter 5 evaluates the results, gives an outlook for future work, and concludes this thesis.

2

Related Work and Fundamentals

This chapter covers the current research on biohybrid systems as well as environmental monitoring. It provides an overview of challenges and applications of both fields and describes the role of projects like WatchPlant for future research. Moreover, the mathematical foundations for the statistical analysis and classification of plant electrophysiological responses as used in this thesis are given.

2.1 Biohybrid Systems

Even today, plants and animals outperform robots in many domains. The combination of nature and technology is a promising tool for developing an understanding of biological processes. This knowledge may open the way for new hardware and software designs and more capable robots.

2.1.1 Robot-Animal Biohybrids

Robot-animal biohybrids can be seen at an individual level, where a single animal is bonded to a robotic system to form a cyborg, or at society level, where individuals of a swarm or herd are replaced by robots, thus forming a mixed society. An extensive overview of the state of the art was carried together by Romano et al. [8] in 2019.

For cyborg systems, research mostly includes the remote control of individuals. Fields of applications for these systems are environment mapping and localisation of disaster victims [9, 10]. Some approaches utilize a direct stimulation of muscles to force movement. Tested insects for muscle stimulation include tobacco hawkmoths [11] or beetles [12]. Other approaches stimulate the brain, ganglia, or antennas to simulate external stimuli. E.g., cockroaches are controllable to make right and left turns with the electrical stimulation of their antennas [13]. Furthermore, the cockroach control was linked to a human using a brain-computer interface [14]. Forward and turning movements of goldfishes were achieved by stimulation of the medial longitudinal fasciculus [15]. Even reptiles like

lizards [16] or mammals like rats [17] could be navigated using direct brain stimulation with implanted electrodes.

Mixed societies introduce one or more robots into an animal swarm. Research topics include studying the animals' behaviour or influencing their collective decision making [18]. Halloy et al. [19] successfully integrated robots into a cockroach swarm. The cockroaches had two possible shelters available to protect against light. Without robots, the cockroaches aggregated under the darker shelter in 73% of the cases. After integrating the robots, which were programmed to prefer the light shelter, the mixed swarm aggregated under the lighter shelter in 61% of the cases. Female túngara frogs were introduced to a male robot that simulated mating behaviour. The replica frog was able to combine audio and visual clues or show them separately to the female, thus allowing to study what aspects of the mating ritual have the most significance [20]. A robotic fish was able to influence the risk-taking of golden shiners in the presence of a supposed threat [21]. Benelli et al. [22] developed a mechatronic device to simulate a potential host for ticks. Since ticks can be dangerous to humans and animals, experiments to understand their behaviour can be useful in combating diseases.

Studies like this allow the investigation of specific animal behaviours, both on individual and swarm basis. They bring new biological understandings that otherwise could only be reached through long-term observations in nature or even not at all.

2.1.2 Robot-Plant Biohybrids

Biohybrid systems consisting of plants and robotic systems offer a variety of applications: In a private setting as home health monitors to extend the capabilities of smart home systems [23], for plant health monitoring in greenhouses for agricultural use [24], which is of importance due to the projected growth of the world population [25], or for urban environment research [1]. The last example is one of the main research goals of the WatchPlant project.

Plant growth and its direction is controllable if the stimuli leading to the wanted reactions are controllable. The EU-funded project *flora robotica* [26] developed closely linked symbiotic relationships between robots and plants. It examined the possibilities of biohybrid plant-robot societies to grow living spaces. It researched current technologies to construct living buildings [27], especially the use of blue portions of light and hormone-based growth inhibitors, to manipulate the rate and direction of plant growth so that plants can be combined with robots in a biohybrid system to grow architectural artefacts [28, 29]. Evaluated methods include machine learning approaches as well as evolutionary robotics [30, 31].

Plants react to many stimuli: Not only light but temperature, air pollution, air and soil humidity, physical stimulation, damage like cutting, etc. Reactions to these stimuli can be measured as electro potentials inside the plant. A project similar to WatchPlant, called *PLants Employed As SEnsing Devices (PLEASED)* [32], had the goal to collect and analyze electro potential responses from plants to use the plants as sensors. Using classification algorithms, the measured signals can be interpreted to the triggering stimuli [7, 33].

To effectively implement biohybrids as sensor networks in remote areas, a power supply for the electronic component is needed. Power supplies can be scarce in remotely monitored areas. Therefore conventional supplies may not be available, thus demanding the development of new methods to compensate for the power consumption. Different approaches of energy harvesting directly from the plant are already researched in the context of renewable energy resources [34]. These approaches contain energy harvesting from motion and vibration via piezoelectric transducers [35], utilizing photosynthesis [36] or using concepts like the microbial fuel cell [37]. WatchPlant will research the use of phloem sap, a fluid consisting of sugars, mineral elements and hormones soluted in water, as an additional possibility [1]. Phloem sap is transported in sieve tube elements inside the plant. These cells are responsible for the distribution of water and nutrients.

Another approach for developing phytosensors can be found in the field of biotechnology and genetic engineering. Two examples are the screening of genes that are upregulated in the presence of trinitrotoluene (TNT) as biological land mine detector [38] or the use of synthetic fluorescent proteins in a tobacco plant as visible indicators for plant pathogens [39]. This thesis does not not evaluate such approaches and instead concentrates on bio-hybrid systems.

2.2 Environmental Monitoring

Environmental monitoring describes the screening of water, air, and soil quality. This can be carried out through the collection and analysis of samples or includes real-time data collection. The importance of environmental monitoring lies in modeling the nature around us: It helps us to understand macroscopic processes, like climate change, and is a tool for forecasting, e.g., weather forecasting or predicting hurricanes or other catastrophic events. The information about traditional environmental monitoring methods is mostly taken from the book *Environmental Monitoring* [40], especially from the chapters 4: *Basic Concepts and Applications of Environmental Monitoring*, 22: *Monitoring, Assessment, and Environmental Policy*, 25: *Biological Indicators in Environmental Monitoring Programs: Can We Increase Their Effectiveness?*, and 27: *Major Monitoring Networks: A Foundation to Preserve, Protect, and Restore*.

2.2.1 Traditional Methods

The easiest way to control the quality of the environment is through observation and the collection of samples. These samples are then examined, e.g., in a research faculty. This approach has two main problems: First, taking only one sample in a big area is insufficient because the sample could be an outlier, thus being unrepresentative for the area. Only one sample can also never reflect changes, so it may be necessary to take multiple samples. Furthermore, samples only show the condition at the time of extraction. This is especially troublesome for water and air samples because both mediums have a potentially fast-changing nature in contrast to soil samples.

This method of environmental monitoring is often impractical because of its associated cost and effort. Analog data loggers can directly write data on paper and can be deployed for a longer time. The data has to be retrieved manually but it is nevertheless an improvement over sample collection. The next and currently most used step is digital high-precision sensors units, often consisting of several different sensors. They can reliably save data for a long time, but still, need to be picked up manually to download the data or have to be installed in a specific place with the required infrastructure to send the data continuously.

A cheaper alternative is the observation of flora and fauna, the so-called biological indicators. Instead of directly measuring, e.g., pollutants in the soil, the plant growth in this area is studied and compared with a known and healthy growing process. When using birds as biological indicators, monitoring their flight routes over multiple years can give us information about the environment: Fish-eating birds choosing a different route or nesting site could indicate the decline of fish in that area. The decline of fish itself can be another biological indicator: Maybe the area is overfished, or a temperature change forced the fishes to another place. However, using biological indicators is criticized as being unscientific. A change in flora and fauna is mostly a complex process; drawing the correct conclusion out observation is objectively difficult due to an information deficiency.

2.2.2 Wireless Sensor Networks

An attempt to solve the aforementioned problems is using a wireless sensor network. An individual sensor unit does not have to be as precise as digital high-precision sensors, and they do not need to have a similar variety of sensors. With these eased constraints, sensor nodes can be built cheaper and in a higher number to be deployed over a larger area. The combination of multiple low-precision sensors compensates for the lack of a single but expensive high-precision sensor. To reach a large variety of sensors, the network can consist of nodes with different sensor configurations that complement each other.

Although sensor networks initially only collected data, the development of smart sensors and smart sensor networks led to more complex application fields. Smart sensors are not only capable of collecting data, but can directly process and compute it, thus eliminating the need for an external computer for every sensor [41, 42]. According to Martinez et al. [43], the progress in this field opens the possibility for the realistic monitoring of our environment.

WatchPlant pursuits an approach with biohybrid sensors instead of purely artificial devices. It aims for a wireless ad-hoc sensor network, where the communication between sensor units is organized automatically without requiring knowledge about the network topology. The sensor nodes will harvest energy from the plants, so except for usable plants, nearly no other infrastructure is needed. Data can be retrieved from every sensor and is computed decentralized in the whole network.

WatchPlant is part of the *Environmental Intelligence Programme* [44]. Another project from

this programme, also for environmental monitoring, is *I-Seed* [45]. In contrast to WatchPlant, I-Seed's goal is not to deploy a sensor network for long-term measurements, but the fast and efficient deployment of biodegradable sensors for short-term use. The sensors are modeled after plant seeds and are dropped from drones. Together, WatchPlant and I-Seed aim to change environmental monitoring to be more efficient and environmentally friendly than traditional approaches.

2.3 Classification

Classification is the process of grouping observations into different classes. An algorithm that assigns a new observation to a class is therefore called a classifier. In this section, only the fundamentals of classification relevant for this thesis are explored: They include feature extraction, which reduces raw data to a smaller set of descriptive features, simple classifiers that are compared for their suitability, and a short overview of machine learning approaches, which will later be used predominantly for WatchPlant.

2.3.1 Feature Extraction

Feature extraction is a method used for dimensionality reduction. From a raw dataset, informative and non-redundant features are calculated. This process reduces the size of the raw dataset while preserving its crucial information. Having a smaller feature set reduces the training time of classifiers, thus saving processing power. For WatchPlants small, plant-powered sensor nodes, the reduction of consumed energy is important.

Eleven statistical features of the data are calculated as done by Chatterjee et al. [7]. These features include: Mean (μ), variance (σ^2), skewness (γ), kurtosis (β), interquartile range (IQR), Hjorth mobility, Hjorth complexity, Hurst exponent, detrended fluctuation analysis (DFA), wavelet packet entropy (WPE), and average spectral power (ASP).
For all features, let $x = \{x_1, x_2, \dots x_n\}$ be the series of measurements.

Standard Features

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i \,, \tag{2.1}$$

$$\sigma_x^2 = \frac{\sum_{i=1}^{n}(x_i - \mu_x)^2}{n} \,, \tag{2.2}$$

$$\gamma_x = \frac{m_3}{m_2^{3/2}} \,, \tag{2.3}$$

$$\beta_x = \frac{m_4}{m_2^2} \,, \tag{2.4}$$

$$IQR_x = Q_3 - Q_1 \,, \tag{2.5}$$

where $m_j = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)^j$. For the interquartile range IQR_x, Q_1 and Q_3 are the first and third quartile of the series x.

Hjorth Parameters

Hjorth parameters [46, 47] are commonly used to analyze signals in the time domain and for feature extraction in electroencephalography (EEG) signals. The parameters are based on the variances of the signal and its first and second derivatives. For discrete series, the variances can be calculated as

$$\sigma_1^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \,, \tag{2.6}$$

$$\sigma_2^2 = \frac{1}{n-1}\sum_{i=2}^{n}(x_i - x_{i-1})^2 \,, \tag{2.7}$$

$$\sigma_3^2 = \frac{1}{n-2}\sum_{i=2}^{n-1}(x_{i+1} - 2x_i + x_{i-1})^2 \,. \tag{2.8}$$

The Hjorth mobility m_H and Hjorth complexity c_H are defined as

$$m_H = \frac{\sigma_2}{\sigma_1} \,, \tag{2.9}$$

$$c_H = \sqrt{\frac{\sigma_3^2}{\sigma_2^2} - \frac{\sigma_2^2}{\sigma_1^2}} \,. \tag{2.10}$$

Hurst Exponent

The Hurst exponent is a feature measuring the long-term memory of a time series, as defined by Hurst [48]. It describes the autocorrelation of the series, that means the self-affinity or the correlation of the series with a delayed copy of itself. Self-affinity is closely related to self-similarity. A self-similar object, e.g., a fractal, is similar to a part of itself. In contrast, a self-affine object is similar to a skewed part of itself. The best-known method for calculating the Hurst exponent was given by Mandelbrot and Wallis [49]. First, the series of length n gets divided into smaller series s of length $m = n, \frac{n}{2}, \frac{n}{4}, \ldots$. The rescaled range for a series s is then given as $\frac{R(m)}{S(m)}$, where $R(m)$ and $S(m)$ are calculated as

$$Z_i = \sum_{i=1}^{m} s_i - \mu(s) \,, \tag{2.11}$$

$$S(m) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(s_i - \mu_x)^2} \,, \tag{2.12}$$

$$R(m) = \max(Z) - \min(Z) \,. \tag{2.13}$$

The Hurst exponent H is calculated by fitting $\mu(\frac{R(m)}{S(m)}) \propto cm^H$, where c is a constant and μ is the mean of the rescaled ranges for all partial series of length m. This is done by fitting

a straight line to the double logarithmic plot of $\frac{R(m)}{S(m)}$ against m. H is the slope of the fitted line. It lies between 0 and 1, and gives the following information about the series:

$$H \begin{cases} < 0.5 & \text{anti-correlated} \\ = 0.5 & \text{uncorrelated} \\ > 0.5 & \text{correlated} \end{cases} .$$

Detrended Fluctuation Analysis

The detrended fluctuation analysis is based on the Hurst exponent, thus also giving information about the self-affinity of a signal, as described by Peng et al. [50]. The main difference to the Hurst exponent is the detrending: Through subtraction of the local trend from all individual values, we can apply DFA to non-stationary signals and expect better results compared to using the Hurst exponent. First, the signal is shifted by its mean and cumulatively summed as

$$y_i = \sum_{i=1}^{n} (x_i - \mu_x) \ . \tag{2.14}$$

The cumulative sum y_i is now split into multiple chunks with window sizes $m = n, \frac{n}{2}, \frac{n}{4}, \ldots$. Every chunk is fitted to a straight line to calculate the local trend (for first order DFA, *nth*-order DFA uses a polynomial fit instead of a linear one), using linear least squares. The fluctuation is calculated as

$$F(m) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - Y(m)_i)^2} \ , \tag{2.15}$$

where $Y(m)$ is the linear estimate of y for window size m.

This is repeated for the different window sizes m. It applies $F(m) \propto m^{\alpha}$, where the exponent α can be calculated the same way as when calculating the Hurst exponent. Depending on α, the following cases can be distinguished [51]:

$$\alpha \begin{cases} < 1/2 & \text{anti-correlated} \\ = 1/2 & \text{white noise} \\ > 1/2 & \text{correlated} \\ = 1 & \text{pink noise, } 1/f \text{ noise} \\ = 3/2 & \text{Brownian noise, } 1/f^2 \text{ noise} \end{cases} .$$

Wavelet Packet Entropy

Shannon entropy is a measure of order, where a high entropy indicates a highly random process, while a low entropy indicates a predictable process. It is calculated as

$$H(R) = -\sum_{i=1}^{j} P(r_i) \log P(r_i) \ , \tag{2.16}$$

where R is a discrete random variable with possibles outcomes $r_1, \dots, r_j$ and their according probabilities of appearance $P(r_1), \dots, P(r_j)$.

The wavelet packet decomposition (WPT) is a generalization of wavelet decomposition (WD). In contrast to WD, WPD decomposes not only the low frequency part of a signal but also the high frequency part, thus decomposing the complete signal. WPD as a method for feature extraction has been shown to be effective, as it uses both frequency and time domain information [52]. The wavelet packet entropy WPE_k is then calculated as

$$WPE_k = -\sum_{c \in C_k} c^2 \log(c^2) \ , \tag{2.17}$$

where C_k is the set of wavelet coefficients at the kth wavelet decomposition level. Similar coefficients indicate a complex series, resulting in a high wavelet packet entropy.

Average Spectral Power

The average spectral power is the integral over the power spectral density (PSD), also called power spectrum. The PSD is a measure of energy variation in a signal, distributed over the different frequencies. We calculate the PSD using the estimate described by Welch [53]. The signal is first divided into k overlapping segments, then a Hann window is applied to each segment. After windowing, the periodogram of every segment is calculated using discrete Fourier transform. The k periodograms are averaged, the resulting function $PSD(f)$ is the power spectral density as a function of frequency f. For a continuous signal, the ASP is calculated as the integral over the PSD, but for a discrete signal, it can be simplified as

$$ASP = \sum_{f \in F} PSD(f) \ , \tag{2.18}$$

where F denotes all frequencies contained in the signal.

2.3.2 Simple Classifier

Two different approaches for classification are using the raw data with a complex classifier (e.g., neural network) or using features of the raw data with a simple classifier. In this thesis, the second approach is used. The basics of discriminant analysis are described in this section, based on the books *Introduction to Pattern Recognition: A MATLAB Approach* [54], *Principles of Multivariate Analysis: A User's Perspective* [55], and *Multivariate Observations* [56]. Let x be an univariate sample with class $y \in Y$ and $\vec{x} = x_1, \dots, x_n$ be a multivariate sample with class $y \in Y$.

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA)

LDA and QDA are related classification methods. Both are Gaussian maximum likelihood classifiers. LDA uses a linear function to separate the classes, while QDA is a generalization of LDA and utilizes a quadratic function. Assume two classes $y = 0$ and $y = 1$. Both LDA and QDA assume that the probability density functions of both classes are normally distributed with means μ_0, μ_1 and variances σ_0^2, σ_1^2. The probability density function (PDF) of a normal distribution $N(\mu, \sigma^2)$ is specified through parameters μ and σ^2 and defined as

$$pdf(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} . \tag{2.19}$$

The likelihood $\mathcal{L}$ indicates the probability that a given observation x is part of a normal distribution with parameters (μ, σ^2). Therefore,

$$\mathcal{L}(\mu, \sigma^2 \mid x) = pdf(x \mid \mu, \sigma^2) . \tag{2.20}$$

For both classes $y = 0$ and $y = 1$, normal distributions $N_0(\mu_0, \sigma_0^2)$ and $N_1(\mu_1, \sigma_1^2)$ are estimated. x is then assigned to the class with the greatest likelihood:

$$y(x) = \arg\max(\mathcal{L}(\mu_0, \sigma_0^2 \mid x), \mathcal{L}(\mu_1, \sigma_1^2 \mid x)) \tag{2.21}$$

The estimated probability that x belongs to a certain class $y \in Y$ is then calculated with Bayes' Theorem as

$$P(y = Y_j \mid x) = \frac{\pi_j \mathcal{L}(y = Y_j \mid x)}{\sum_i \pi_i \mathcal{L}(y = Y_i \mid x)} , \tag{2.22}$$

where π_j is the prior probability of class j.

This method can be generalized for multivariate distributions with a multivariate sample $\vec{x}$. The PDF uses the covariance matrix Σ instead of the variance:

$$pdf(\vec{x} \mid \vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^\top \Sigma^{-1} (\vec{x}-\vec{\mu})} , \tag{2.23}$$

where $|\Sigma| \equiv \det \Sigma$. The covariance matrix is calculated as

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^{n} (\vec{x_i} - \vec{\mu})(\vec{x_i} - \vec{\mu})^\top . \tag{2.24}$$

The difference between LDA and QDA lies in the covariance matrices. LDA restricts the classifier with the assumption that all classes have the same pooled covariance matrix, in the case of two classes: $\Sigma_0 = \Sigma_1$. The pooled covariance matrix Σ_p can be calcualated for two classes as

$$\Sigma_p = \frac{(n_0 - 1)\Sigma_0 + (n_1 - 1)\Sigma_1}{n_0 + n_1 - 2} . \tag{2.25}$$

For univariate samples, LDA assumes $\sigma_0^2 = \sigma_1^2$. The pooled variance σ_p^2 is the weighted arithmetic mean of all class variances.

Instead of the likelihood, the log-likelihood is often used due to computational convenience. It is the natural logarithm of the likelihood function. Maximizing the log-likelihood is the same as maximizing the likelihood because the natural logarithm is a strictly increasing function. To summarize the previous equations under the usage of log-likelihood: For classes $Y = y_1, \dots, y_j$ and a sample $\vec{x} = x_1, \dots, x_n$, the sample is classified with the functions *LDA* and *QDA* as

$$y(\vec{x} \mid \vec{\mu}, \Sigma_p) = \arg\max_j LDA_j(\vec{x} \mid \vec{\mu}, \Sigma_p) \tag{2.26}$$

$$y(\vec{x} \mid \vec{\mu}, \Sigma) = \arg\max_j QDA_j(\vec{x} \mid \vec{\mu}, \Sigma) \ , \tag{2.27}$$

where *LDA* is defined as the linear discriminant function

$$LDA_j(\vec{x} \mid \vec{\mu}, \Sigma_p) = \vec{x}^\top \Sigma_p^{-1} \vec{\mu_j} - \frac{1}{2} \vec{\mu_j}^\top \Sigma_p^{-1} \vec{\mu_j} + \log(\pi_j) \tag{2.28}$$

and *QDA* is defined as the quadratic discriminant function

$$QDA_j(\vec{x} \mid \vec{\mu}, \Sigma) = -\frac{1}{2} \log|\Sigma_j| - \frac{1}{2} (\vec{x} - \vec{\mu_j})^\top \Sigma_j^{-1} (\vec{x} - \vec{\mu_j}) + \log(\pi_j) \ . \tag{2.29}$$

π_j again denotes the prior probability of class j. The estimated normal distributions for LDA are only distinguishable due to their different means. Therefore, QDA reaches a better classification accuracy than LDA if the classes are not well linearly seperated.

Naive Bayes LDA and QDA

Naive Bayes classifiers assume a strong independence between features for multivariate samples. Due to its simpler nature, naive Bayes is often overshadowed by other classifiers, but it can yield good results for small datasets. Naive Bayes LDA and QDA also use Equation 2.23, but assume that the covariance matrix is diagonal. Since Equation 2.23 is a generalization of Equation 2.19, the univariate covariance matrix is a 1×1-matrix, which is always diagonal. That means both naive Bayes LDA and QDA only differ from standard LDA and QDA for multivariate samples.

Mahalanobis Distance

The Mahalanobis distance describes the distance between a point and the mean of a distribution, measured in standard deviations of said distribution. It is calculated as

$$d(\vec{x} \mid \vec{\mu}, \Sigma) = \sqrt{(\vec{x} - \vec{\mu})^\top \Sigma^{-1} (\vec{x} - \vec{\mu})} \ . \tag{2.30}$$

For univariate LDA and QDA, the euclidean distance is used, as seen in the exponent in Equation 2.19, while the multivariate generalization (Equation 2.23) uses the Mahalanobis distance. For multivariate classification, the Mahalanobis distance is a better

measure than the euclidean distance because it is influenced by data correlation (the covariance matrix). In contrast, euclidean distance uses only the mean of a distribution and ignores covariances.

Corresponding MATLAB Functions

We use MATLAB for classification, more specifically, the method *classify* [1]. It allows five different discriminant types, specified as *linear*, *quadratic*, *diagLinear*, *diagQuadratic*, and *Mahalanobis*. Linear and quadratic use the standard algorithm, Equations 2.28 and 2.29, respectively. DiagLinear and diagQuadratic are the naive Bayes variants of LDA and QDA, therefore using the same discriminant functions as standard LDA and QDA, but with a diagonal covariance matrix estimate. Mahalanobis is similar to QDA, but the discriminant function is only dependent on the Mahalanobis distance and ignores prior probabilities:

$$mQDA_j(\vec{x} \mid \vec{\mu}, \Sigma) = -(\vec{x} - \vec{\mu_j})^\top \Sigma_j^{-1} (\vec{x} - \vec{\mu_j}) \ . \tag{2.31}$$

2.3.3 Machine Learning Approaches

Machine learning emerged as a separate research field from artificial intelligence. It includes methods for informed decision making. These algorithms do not use hardcoded conditionals but dynamically adapt to new data. They extract the underlying coherences and dependencies in obfuscated data, thus constructing a mathematical model. This model can be used by machine learning algorithms to classify unknown data or predict the next data points based on learned trends. Machine learning is advantageous for complex and unstructured data. For a classification using e.g. LDA, the programmer has to determine well-discriminating features explicitly. While this may be simple for an incomplex dataset, it is nearly impossible for complex data [57].

Plants are still a black box: They react to many different stimuli, maybe even some yet unknown stimuli, in a way we do not completely understand. That makes the raw data hard to interpret, a premise speaking for the use of machine learning. WatchPlant wants to use machine learning for three main goals: Classification, short-term prediction, and long-term prediction. Classification means assigning a plant response to its triggering environmental stimulus. Long-term prediction is used to foretell climate development, e.g., an increase in temperature or a decline in pollution. Short-term prediction is not used for environmental monitoring but for efficient power consumption. The deployed sensors need to be self-sufficient and should use as little power as possible. Wireless communication needs a lot of energy, so the frequency of transmissions depends on the predicted probability that the sensor measures different data. If the predicted probability is low, the sensor stops transmitting for some time and continue later to save energy.

The specific machine learning algorithms WatchPlant will use have yet to be decided. Current propositions for online machine learning include computational light meth-

[1] Find the documentation of the MATLAB method "classify" at: https://de.mathworks.com/help/stats/classify.html

ods, like artificial neuronal networks (ANN), long short-term memory networks (LSTM), transformers, or recursive ANNs, coupled with evolutionary computation methods. For offline evolution, also more recent approaches like MAP-elites are considered [5, 6].

Despite the advantages of machine learning, we still observe the performance of classic statistical analysis and classification. This is useful because initial experiments for WatchPlant are done in a controlled environment with only two stimuli. The following experiments and classifications serve as feasibility analysis for differentiating blue and red light stimuli based on plant electrophysical responses.

3

Living Sensor Network

In this chapter, we will describe the developed living sensor network. The network consists of a biological part - a plant, and a technological part, which is a measuring and computing device. We will present the used plants and electronics, the difficulties we had with them and the implemented software to run the sensor network.

3.1

Figure 3.1: Two individuals of the plant genera we mostly used: A Dracaena on the left side and a Zamioculcas on the right.

The biological part of the biohybrid sensor node will be a natural plant that is attached to the electronic part. Useful plants for this project have to be plants that produce a strong electro potential response as a reaction to different stimuli. Furthermore, they need to survive in a lab environment and should be reasonably maintainable. We choose plants of the genera *Dracaena*, *Zamioculcas*, and *Maranta*, based on recommendations from Cybertronica Research, one of our collaboration partners in WatchPlant. We used Dracaena and Zamioculcas plants for most of our experiments, see Figure 3.1, but also did experiments on *Solanum lycopersicum* (tomato plant) to recreate the results by Chatterjee et al. [7]. All the plants we tested from this selection showed strong responses, but tomatoes proved to be much more sensitive to interfering stimuli than Dracaena and Zamioculcas. They needed more care, e.g., a stricter watering routine, and could not be used for long-term experiments over multiple days. Tomatoes are seasonal plants, so in addition to their sensibility they are not as useful as non-seasonal plants for a multiyear research project.

3.2 Electronics

Here we give an overview of the used hardware for every sensor node of the network. We introduce the phytosensor that measures electro potentials from plants, the computational unit for data processing, and discuss the sensors' attachment to plants.

3.2.1 CYBRES Measurement Unit

This thesis observes electro potential reactions, but in the further course of WatchPlant, also other physiological reactions can be of interest, e.g., the leaf transpiration. We use a phytosensor developed by CYBRES, the *phytosensing and phytoactuating system* [2]. It combines the modular version of the CYBRES Measurement Unit 3.4 (MU) with multiple sets of electrodes and external sensors, as seen in Figure 3.2. The sensor supplies two channels with two biopotential electrodes each to measure the electro potential between both electrodes. Additional sensors include two pairs of tissue impedance electrodes, external temperature, air humidity, leaf transpiration, light, soil moisture, and soil temperature sensors, an RF antenna, an internal accelerometer, magnetometer, temperature sensor, and EM power meter.

All electrodes are made out of silver (Ag99). Additional equipment that could be useful for later research includes stainless steel electrodes for tougher plant stems and a sap flow sensor to examine phloem sap. The MU itself is capable of performing different data processing methods as well as directly saving data onboard, thus being able to run experiments without any other devices. More information can be found in the manual [3].

[2] CYBRES phytosensing and phytoactuating system: http://cybertronica.de.com/?q=products/phytosensor

[3] Find the MU documentation at: http://cybertronica.de.com/download/MU-EIS_Manual_en.pdf

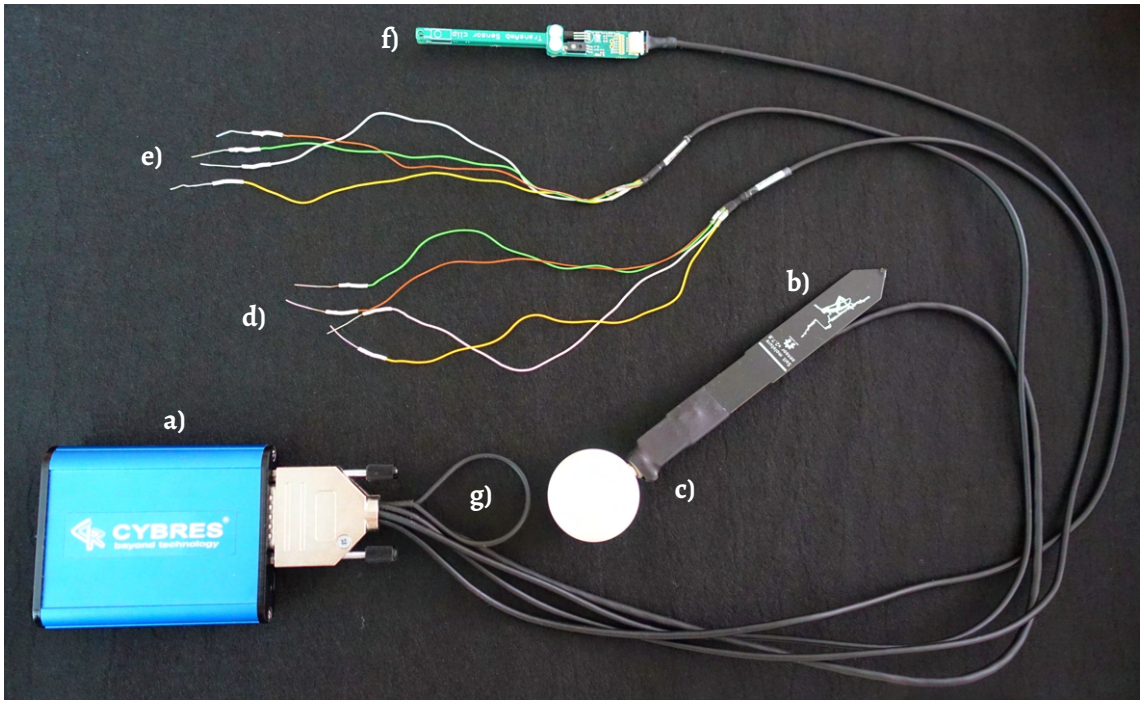

Figure 3.2: The a) CYBRES MU with the attached sensors: A b) soil sensor with c) RGB light actuator, d) impedance and e) biopotential electrodes, and a f) leaf transpiration sensor board with included temperature, air humidity, and light sensor. The cable loop is an g) RF antenna.

3.2.2 Raspberry Pi and Extensions

Even though we can run experiments only using the MU, we combine it with a Raspberry Pi 4 Model B [4] for additional utility. The Raspberry Pi is a small single-board computer commonly used in research. We connect it via its USB port with the MU. In this configuration, we only use the MU as a sensor for the Raspberry Pi and dismiss its onboard processing capabilities. We can extend our setup with other external sensors as long as they can be connected to the Raspberry Pi (using USB or general purpose input-output (GPIO) pins). Additional sensors we added are an RGB sensor (TCS34725) [5] and an ozone sensor (DGS-O3) [6]. We used them to measure the environmental data during experiments with light and ozone as stimuli. This data can be compared to the measured electro potentials and serve as reference data.

[4] Raspberry Pi 4: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/
[5] RGB sensor: https://www.adafruit.com/product/1334
[6] Ozone sensor: https://www.spec-sensors.com/product/digital-gas-sensor-ozone/

3.2.3 Attachment to Plants

The attachment of the sensors to the plant is straightforward: The electrodes of one channel both have to be inserted into the plant stem, or one inside the stem and the other inside the soil. Putting electrodes into the soil only works if the conductivity of the soil is high enough, so this is mostly an option for plants that need much water. The soil sensor is placed inside the soil. It should not be placed directly next to the plant to prevent triggering a stimulus reaction of the plant (through e.g. touching the plant or its roots). As long as the sensor is placed somewhere in the plants' pot, we can assume that temperature and moisture are distributed equally and that we measure correct values. The leaf transpiration sensor has to be placed on the leaf. Depending on the stability of the plant's leaf, it should be supported by a rod or fixed to the stem so that it does not damage the leaf. The other external sensors of the MU, the temperature, air humidity, and light sensor, are on the same board as the leaf transpiration sensor. Depending on its placement, we should use other external sensors to measure these environmental conditions because we can not place them completely free.

3.3 Challenges

During our initial experiments, we had difficulties collecting consistent datasets. Sometimes we could not measure any reaction to a stimulus application, other times, the base potential was so noisy that a response to a stimulus was not stronger than the noise itself. We discuss strategies to ensure producing better datasets based on our experience.

3.3.1 Electrode Placement

The electro potentials are measured with silver electrodes that are inserted into the plant. The correct attachment of the electrodes proved to be more difficult than we initially anticipated. We tested different electrode placement settings and compiled our results as guidelines for further experiments. Note that these results are only based on our observations with a limited choice of plants. An analysis of electrophysiological processes inside plants could help to find more scientific results.

Distance between Electrodes

The distance between the electrodes depends on the species of the used plant. For tomatoes, 3 to 4 cm seem to be ideal, while Dracaena and Zamioculcas plants give the best reponses at 5 to 10 cm. There may also be a correlation between plant size and electrode distance in the same plant species, but due to a lack of differently sized plants, this has yet to be confirmed.

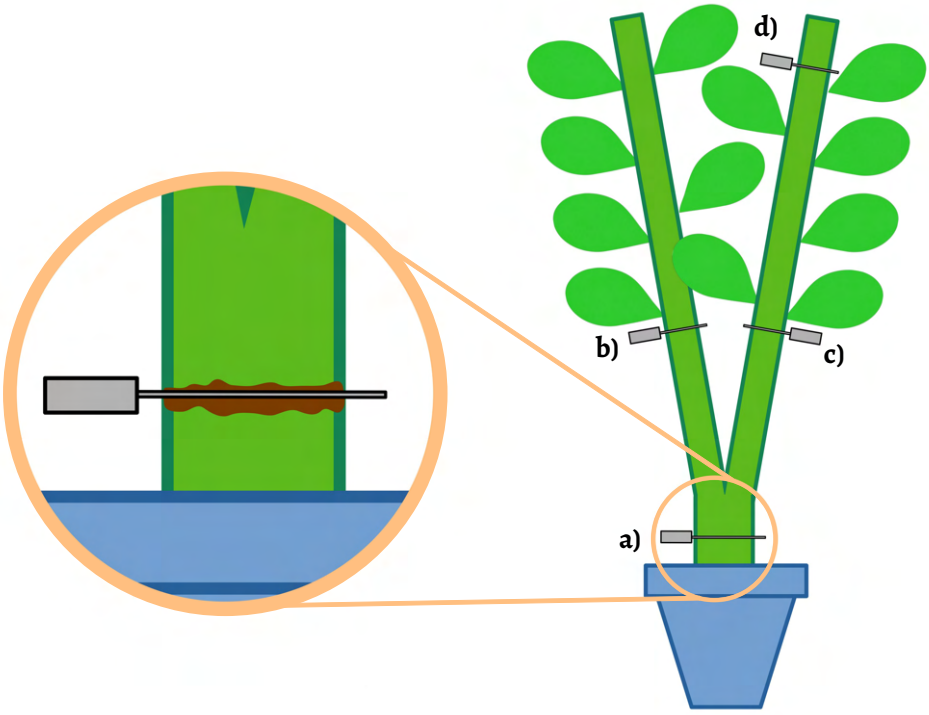

Figure 3.3: Examples of electrode placement. The electrodes are pushed completely through the stem. Measuring the electro potential from electrode a) to b) or a) to c) satisfies the requirements listed in this section. Between electrode d) and the other electrodes are leaves, so the signal is noisy. Our experiments have shown that measuring between electrodes b) and c) is possible without a decline in the response quality, even though the electrodes are no longer placed in the same stem. The magnification depicts the lignification of the plant around an inserted electrode. To prevent this, the electrode placement has to be changed regularly.

Electrode Insertion Depth

The electrodes have a length of 2 cm and should be inserted centrally through the stem. Some parts of the stem allow better measurements than others, but we did not have a method to reliably find these parts. By pushing the electrode completely through the widest part of the stem (see Figure 3.3), we maximize the chance to place the electrodes at positions where the electro potential is well measurable. For plants with hard stems, e.g., a Dracaena, a needle is used to drill a small hole in it so the electrodes do not bend.

Plant Topology

The placement of the electrodes in the context of plant topology has shown no significant difference. That means the electrodes could be placed in the same stem or in different stems, as long as all other placement criteria are adhered (see Figure 3.3).

Plant Leaves

Experiments have shown that meaningful plant reactions are only measured if the electrodes are inserted in a stem with sufficient leaves. The leaves seem to be the plants "sensors". There need not be leaves between the electrodes, just somewhere on the same stem. In fact, leaves between electrodes may actually worsen the signal because it gets noisier (see Figure 3.3). The reasons for this are probably many small, individual influences of each leaf. In a lab environment, they do not make a big difference, but this effect has to be studied in more depth for outdoor use.

Duration of Experiments

To some degree, plants are capable of healing small injuries themselves. Every electrode is a foreign object, every insertion a wound the plant wants to repair. The plant begins the process of lignification. This makes the measurements unusable, because small foreign objects inside the stem get encapsulated due to the lignification, thus are cut off from the living plant (see Figure 3.3). For this reason electrodes should not be inserted repeatedly in the same spot. The pace of this process seems to depend on the plant species and should be taken in account for long-term experiments. If the plant is rotting, we equally do not measure good responses in the rotten part of the stem.

3.3.2 Stimuli Interference

For the following experiments, only one stimulus is changed at every time. In this way, the behaviour of the electro potentials can be observed as a reaction to one single stimulus, which makes it easier to classify than reactions to multiple stimuli simultaneously. Since plants are very sensitive organisms, reducing all other environmental stimuli is not easy, especially because we do not know yet which stimuli plants may or may not react to.

Light, Temperature and other common Influences

All common stimuli should be fixed during an experiment. We could not find a difference if a stimulus, e.g., light, was applied or not during an experiment, but it has to be consistant: No changes in the light intensity are allowed during, e.g., a temperature experiment, regardless of whether the light is on or the lab is in complete darkness. Com-

mon stimuli that are relatively simple to avoid inside a lab are, e.g., light, temperature, air pressure, air humidity, mechanical stimuli like wind or the noise level.

Importance of Watering

Two points have to be considered regarding plant watering:

After the plant is watered, it yields a strong electrical response that can overshadow the real experiment. If the experiment only takes a few hours and the plant does not need to be watered in the meantime, the experiment should be started about half an hour after the watering. That gives the plant enough time to acclimate itself. For long-term experiments, watering may be necessary, but then strong responses have to be expected.

The hydration level also seems to be responsible for the general quality of responses. A well hydrated, healthy plant yields far better results than a plant that is over- or underwatered. Thus it is important to study the used species and its needs before executing experiments to ensure the health of the plants. We did not test the following hypothesis, but watering is probably not the only health factor that can influence measurements. This is important to consider for long-term experiments under lab conditions: Depending on the specific experiment, the plants health could decline over time due to deprivation of, e.g., light, water or oxygen. Measurements from these experiments are then potentially influenced by the experiment itself and do not portray the natural responses of plants outside of labs.

Electric and Magnetic Fields

In and around the lab are many electric devices that emit electric fields. Some of them may be strong enough to interfere with experiments, as the MU measures the electro potentials with a resolution of $\pm 64nV$. Furthermore, reactions to some stimuli may be very small, so even a weak electric field could produce significant noise. To reduce these effects, the plants can be placed inside a faraday cage. The cage we built has an inner skeleton out of aluminum, which is encased with aluminum mesh. The cage has a cylindrical shape with a height of 1 m and a diameter of approximately 62 cm. The mesh has a grid size of 1.53 mm to 1.19 mm and material strength of 0.22 mm. Our experiments have shown that the noise can be greatly reduced with a faraday cage for tomatos but the effect on Zamioculcas or Dracaenas was much smaller.

Magnetic fields can be measured with the MU. We did not observe significant changes in them and assume the magnetic field as stable, so constructing a shielding against it is not worth the cost or effort for a small, maybe even unnoticeable effect.

Repeating Stimuli

We noticed that plants can get used to stimuli if we apply them too often in short intervals. The measured amplitude of the response decreases with every stimulus application. For our experiments with Dracaena, Zamioculcas and tomato plants, we set a minimum

delay of 30 minutes between stimuli, but this has to be investigated for every new plant species individually. We could not observe long-term acclimatization to the same stimuli in experiments over a few weeks, only short-term acclimatization if we have intervals of e.g. five minutes.

3.4 Software

Here we describe the software that runs on the living sensor network. We will begin with an overview of the general structure and then go through implementation details of the communication between biohybrid sensor nodes and edge device, communication with the MU, and communication with further external sensors. Finally we look at the live plotting to monitor currently running experiments.

3.4.1 Overview and Structure

With the described hardware, we can build a single sensor and can locally obtain data from one plant. As we want to generate larger datasets with multiple different plants, a solution is needed to use any number of biohybrid sensor nodes easily. A centralized network approach is used to store the data from multiple nodes on one edge device. The edge device simply is another Raspberry Pi that is always running, ready to receive and save measurements from any experiments.

The structure of the network is shown in Figure 3.4. For both the sensor node and the edge device, a *main* method is implemented as an easy starting point of the program. Running the main methods creates a *sensor node* respectively *edge device* object, which handles the control logic of the network. Starting the sensor node over its main method allows the additional transfer of command line arguments, an excerpt is shown in Table 3.5. The sensor node receives measurements from the *MU* and optionally from *additional sensors* attached to the Raspberry Pi. After preprocessing the data, it is sent to the edge device using the *ZMQ publisher* and receiving it with the *ZMQ subscriber*. Wireless communication with ZMQ is described in more detail in Section 3.4.4. The edge device automatically creates folders for every connected biohybrid sensor node and stores CSV files in them. All files are saved in */home/$USER/measurements/hostname*, where hostname is the hostname of the sending node. Files created by the edge device are named *hostname_YYYY_MM_DD-hh_mm_ss.csv*. *YYYY* denotes the year with four digits, *MM*, *DD*, *hh*, *mm*, *ss* denote the month, day, hour, minute, and second with two digits each. The sensor nodes also save the data locally as additional backup.
The code is written in Python 3 and saved on the internal GitLab of the Institute of Computer Engineering, UzL. It is available in the *MU_interface*-repository[7].

[7] Find the code of the living sensor network at: `https://gitlab.iti.uni-luebeck.de/watchplant/mu_interface`

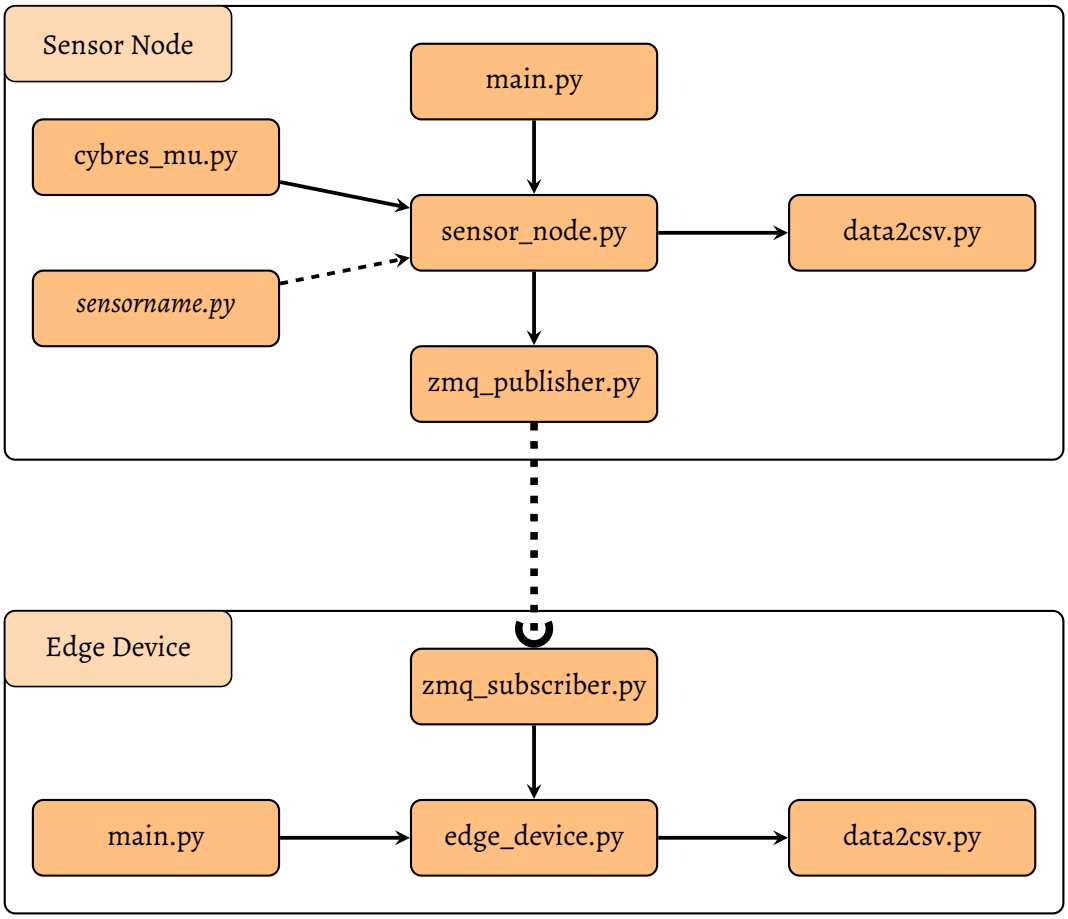

Figure 3.4: Main software structure of the living sensor network. While the figure shows only one edge device connected to one sensor node, it can run with an arbitrary number of sensor nodes that all send their data to one edge device. *sensorname.py* is a placeholder for arbitrary sensors that extend the base setup.

Parameter	Type	Default	Description
–port	String	/dev/ttyACM0	USB-port where the MU is connected
–int	Integer	10000	Measurement interval in miliseconds
–addr	String	localhost	Network address of the edge device

Table 3.5: Excerpt of important command line arguments for the sensor node with default values.

3.4.2 Accessing the MU

The MU is connected to the Raspberry Pi with a USB cable. CYBRES does not provide a Linux compatible client, but the MU operating system can communicate via an ASCII interface. It utilizes a serial port with a baud rate of 460.8 kBd and can be accessed with a terminal emulator. A command has the format

$$k_1 k_2 x_1 x_2 \dots x_n{}^* \ ,$$

where k_1 and k_2 are control parameters and $x_1, \dots, x_n$ are optional parameters depending on the command. The asterisk symbol (*) is used to indicate the end of a command. An excerpt of commands is displayed in Table 3.6, while the full list of commands can be found in the MU user manual [8].

k_1	k_2	x	Description
s	s		Show all parameters
s	r		Restart the system
m	s		Start measurements
m	p		Stop measurements
m	i	$x_1x_2x_3x_4x_5$	Set measurement interval to $x_1x_2x_3x_4x_5$ ms

Table 3.6: Excerpt of MU commands for communication with other devices using the ASCII interface.

3.4.3 Sensor Extensions

The aforementioned setup of the biohybrid sensor node is only a base configuration that can be extended. The only prerequisites for adding sensors is that they can be used with Python 3 on Raspberry Pi OS. To include a new sensor, a python class has to be written and saved in *Sensor/Additional_Sensors*. This class should implement an *__init__* method to set the sensor up, as well as a *getData* method that returns all relevant measurements. The new sensor can then be easily integrated in the setup. No changes are needed on the software of the edge device, a new sensor node with different sensor configuration can therefore be added to an existent setup without stopping the measurements.

3.4.4 Networking and Communication

To send data from a sensor node to the edge device, both devices must be connected to the same wifi network. They use ZeroMQ[9], a universal networking framework. We utilize the publisher-subscriber model, where the edge device subscribes to every data stream that is directed to it. After the sensor-side Raspberry Pi receives a data line from

[8] Find the MU documentation at: http://cybertronica.de.com/download/MU-EIS_Manual_en.pdf
[9] Find the ZeroMQ homepage at: https://zeromq.org/

the MU, it sends them as multipart message to the edge device. Multipart messages are a method provided by ZeroMQ to built custom message formats out of simpler methods. Examples of these simple methods are *send_string* or *send_json*. The corresponding methods *recv_string* and *recv_json* are used to receive the string or json on the other device. The multipart message for this setup consists of a header and a payload. The header includes the hostname of the sending node, an integer indicating the message type and thus also the payload type and a boolean indicating the usage of other external sensors apart from the MU. With the help of the header, the edge device can match incoming messages to the different sensor nodes and properly save the data.

If the sending sensor node is extended with additional sensors, the number of extra data fields as well as the names of them are also transmitted. These measurements are saved in the CSV file alongside all standard data fields. If a sensor node with a new sensor configuration is connected, no configuration on the edge device is needed.

The implemented ZeroMQ-based communication is not encrypted and should not be used to share measurements over the internet. Nevertheless, data can be shared across multiple devices using Syncthing[10]. Syncthing synchronizes the complete measurements folder of the edge device with any connected computer.

3.4.5 Live Plotting

To examine the experiments in real time, the last part of the living sensor network is a life plotting script. It reads the experiment data saved in CSV files and actualizes them automatically. The plotting itself is done with Matplotlib [58], a plotting library written for Python. The code is available in the *Data_analysis*-repository[11]. Apart from continuous data plotting, the script has the following interactive features:

- Sensor node selection: The script reads the measurement history of all connected sensor nodes and allows the user to switch between them.
- Sensor data selection: Different sensor data can be individually selected and displayed. In the base configuration, available data to choose from are:
  - temp-external
  - light-external
  - humidity-external
  - differential_potential_CH1
  - differential_potential_CH2
  - transpiration

  These originate from the MU, but when additional sensors are added (as described in Sections 3.2.2 and 3.4.3), the script can easily be extended to include them.
- Data scaling: The plot can be switched between actual values and normalized values (min-max normalization between zero and one).

[10] Find the Syncthing homepage at: https://syncthing.net/
[11] Find the code of the live plotting script at: https://gitlab.iti.uni-luebeck.de/watchplant/data_analysis/-/tree/master/liveplot

– Filtering: To smoothen the data, a Savitzky-Golay filter with varying window sizes can be applied.

An example of the interface is displayed in Figure 3.7. Since the plotting depends on CSV files, it will run without further changes on any device that is synced with the edge device. Thus, running experiments can be observed from any computer as long as it has internet access.

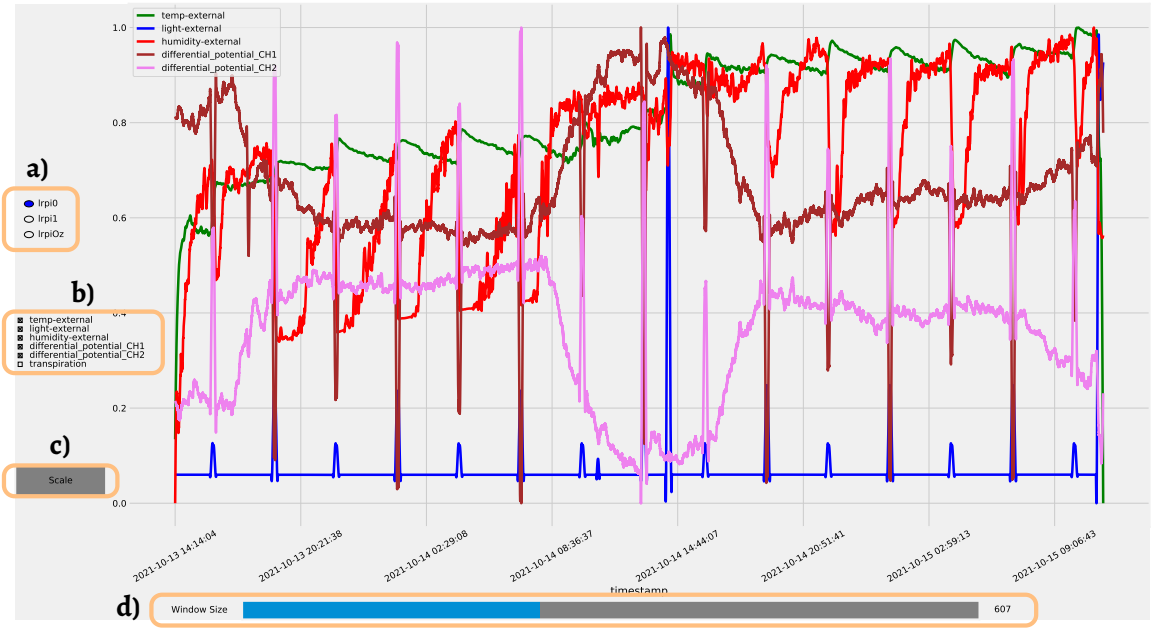

Figure 3.7: Live plotting script running on the edge device. We can choose a) to display data from different biohybrid sensor nodes, b) what sensor data to plot, c) whether to see total values or normalized data, and d) whether to filter the data or to display raw values.

4

Stimuli Classification

This chapter describes our approach to classify between red and blue light stimuli based on plant electrophysiological responses. We first go over the specifics of the experiment execution and explain the process of classification in detail. Afterwards, we examine the results and discuss unexpected plant behavior that occurred during the experiments.

4.1 Data Collection

With the following experiments, we collect electrical signals from plants in response to red and blue light stimuli. This data is later used to examine the influence of red and blue light on the electrical signal of a plant. Plants can perceive a light spectrum from UV-B (280 nm) to infrared (750 nm). Blue and UV-A light (340-500 nm) are known to be the trigger of phototropism: The plant reacts to the light by growing towards it [59]. If the light source is not positioned directly above the plant but only on one side of the stem, a plant hormone called auxin accumulates at the other side of the stem. As a consequence, this side of the stem is growing more than the illuminated side, thus pushing the tip of the plant towards the light. Red light (600-640 nm) does not trigger the phototropism, but has another important role for the plant: It is significantly more efficient in driving photosynthesis [60].

Due to the observable differences between red and blue light, we also expect a measurable difference in the electro potentials. Green light is excluded from the experiment, as green light is mostly reflected by plants so we don't expect interesting physiological responses [61].

Figure 4.1: The setup includes a), b), c) red LED panels and d), e) blue LED panels for stimuli applications. One blue panel was removed for better visibility. The f) MU is attached to the g) plant and the h) Raspberry Pi, which is extended by an i) RGB sensor. The attached electrodes are shown in normal light for better visibility.

This experiment uses the setup described in Chapter 3 and follows the guidelines discussed in Section 3.3. An RGB sensor is added to the setup (see Sections 3.2.2 and 3.4.3 for sensor extensions) to record light intensity values during experiments. The blue and red light stimuli are applied with three LED panels each surrounding the plant. Figure 4.1 shows the complete setup with a Zamioculcas. Two wifi-plugs automatically control the lights. Blue and red light experiments alternate every three hours. The stimulus for an experiment (blue or red light) is always applied for ten minutes, after which the light will again turn off. Experiments take place in a lab which doesn't allow the presence of external light. During experiments, nobody is allowed to open the lab and influence the plants. Electro potentials are measured with a frequency of 1 Hz. Initial experiments and comparisons between the different available plants showed that Zamioculcas have the most stable response, so all datasets for data analysis in Section 4.3 are collected on them.

This setup can collect eight individual experiment datasets per day from a single measurement channel of the sensor node; four red and four blue light datasets. An experiment dataset is defined by three parts: First, one hour background measuring, that is, measuring the plants electrical baseline without an active stimulus. This part is used for the background mean subtraction as described in Section 4.2. After that, ten minutes of measuring with stimulus application follow. Finally, again one hour without active stimulus is measured. This last hour is important to see if the stimulus has effects which last longer than the application time of ten minutes. This experiment procedure is not only used for this thesis, but was agreed on by WatchPlant collaboration partners UzL[12], FER[13] and CYBRES[14] for further experiments with other stimuli.

Every experiment ran between 36 and 96 hours, after which the plant was watered and exposed to sunlight. If a plant is kept longer than a week in the experiment, the quality of responses drops. As this is an indicator of declining health, the plants have to be changed regularly.

4.2 Data Processing and Classification

The here described steps of data processing are additionally shown in Figure 4.2 and are based on the method described by Chatterjee et al. [7]. They are done for every individual experiment dataset. Each raw experiment dataset contains the electrical response to exactly one stimulus application.

First, the starting time of the application splits the experiment dataset in a pre-stimulus and post-stimulus part. As a consequence, the pre-stimulus part contains one hour of the electrical baseline (the electrical signal of the plant without stimulus application). The post-stimulus part contains ten minutes of stimulus application and then another hour without stimulus application.

[12] Universität zu Lübeck, Institute of Computer Engineering
[13] University of Zagreb Faculty of Electrical Engineering and Computing, Laboratory for Robotics and Intelligent Control Systems
[14] CYBRES GmbH, Research Center of Advanced Robotics and Environmental Science

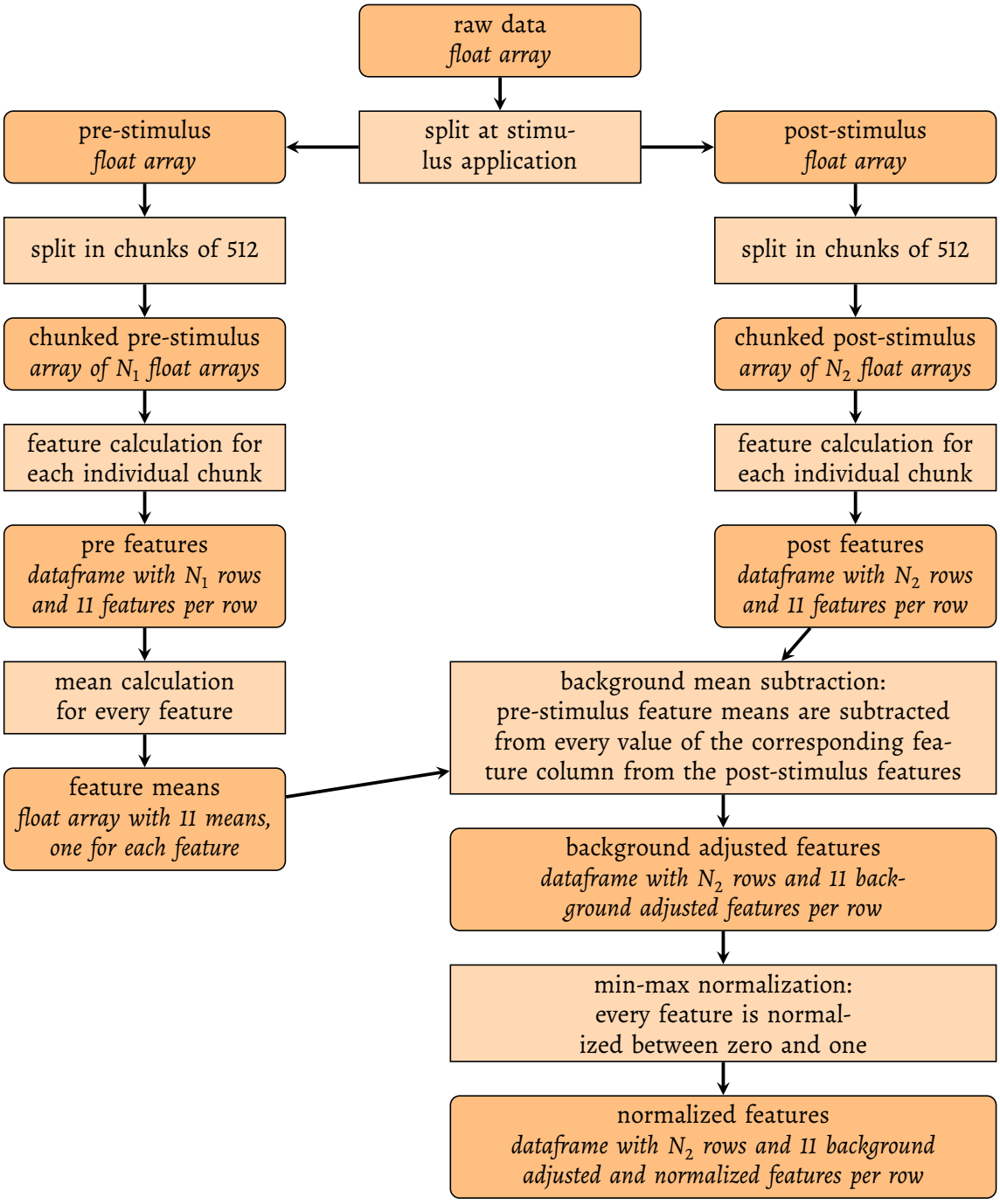

Figure 4.2: This flowchart describes the data processing and feature extraction. Each raw experiment dataset is processed in this way. All the calculated, background adjusted and normalized features from the same stimulus are then concatenated into one file. For two different stimuli, e.g. red and blue light, this leaves us with two files containing all features, which are now ready for classification.

Next, both the pre- and post-stimulus parts are divided in smaller chunks. The size of chunks is defined by the window size, for which we used 512. A smaller window size is preferable, because that means we need less data for classification, requiring less processing power and energy. A bigger window improves the classification accuracy, but only at the expense of needing more computational power.

After that, eleven features are calculated for each chunk, as explained in Section 2.3.1: The mean (μ), variance (σ^2), skewness (γ), kurtosis (β), interquartile range (IQR), Hjorth mobility, Hjorth complexity, Hurst exponent, detrended fluctuation analysis (DFA), wavelet packet entropy (WPE), and average spectral power (ASP).

Every plant has a different electrical baseline, depending on species, age, electrode placement and other factors which have yet to be figured out. As the classifiers shall distinguish between different stimuli and not between different plants, the electrical baseline is subtracted. To do that, we calculate the mean of each of the eleven features from the pre-stimulus part. To illustrate this with an example, assume we divided the pre-stimulus part in N_1 chunks. For every chunk, we calculated eleven features. This leaves us with N_1 values for the mean, N_1 values for the variance, etc. Now we calculate the mean of each of the eleven features: The mean of N_1 means, the mean of N_1 variances, etc. Eventually, we have exactly one value for each of the eleven features.

Then we perform the background mean subtraction: We subtract the feature means we calculated from the pre-stimulus part in the previous step from every calculated post-stimulus feature. By doing that, we only consider the incremental changes and do not accidentally train the classifier to distinguish between two different electrical baselines. To stick with our example, assume we divided the post-stimulus part in N_2 chunks and calculated the eleven features for every chunk. We now subtract the pre-stimulus feature means from each of the N_2 feature sets: We subtract the mean of pre-stimulus means from each of the N_2 post-stimulus means, the mean of pre-stimulus variances from each of the N_2 post-stimulus variances, etc. After this step, we have N_2 feature sets with background adjusted features.

Finally, the calculated features are normalized as preparation for classification. Normalization, also known as feature scaling, is done as min-max normalization: Every feature is adjusted so that the minimum and maximum values are zero and one. Let $\vec{x} = x_1, x_2 \ldots, x_n$ be a vector to be normalized between zero and one and $\hat{x}$ be the new, normalized value of x. Normalization is then calculated as

$$\hat{x_i} = \frac{x_i - \min(\vec{x})}{\max(\vec{x}) - \min(\vec{x})} \,. \tag{4.1}$$

Continuing with our example, the minimum value of all N_2 means is now zero and the maximum value for them is one, the minimum value for all N_2 variances is now zero and the maximum value is one, etc.

The implementation for the feature calculation is written in Python 3 [15]. For mean, variance, skewness, kurtosis and IQR, Python provides built-in methods for calculation. The

[15] Find the code for feature extraction and classification at: https://gitlab.iti.uni-luebeck.de/watchplant/classification

other features had to be self-implemented, partially based upon the methods provided in the MATS [16] MATLAB Toolbox. This toolkit was used by Chatterjee et al. [7].

The feature calculation may compute NaNs (Not a Number), which are undefined results of operations. For example, division by zero produces NaN as output. They lead to errors in the classification process. NaN values are replaced with the mean of the corresponding feature. The prepared features are classified with the five classifiers described in Section 2.3.2. They are validated using leave one out cross-validation (LOOCV). For a feature dataset with n datapoints, LOOCV chooses $n-1$ datapoints as training data and the remaining datapoint as test data. The test data is used to validate the performance of the classifier that was trained with the training data. This process is repeated n times so that every datapoint is used as test data once. The performance of the classification is averaged over all n iterations.

Chatterjee et al. [7] differentiated between one versus one (1v1) and one versus rest (1vR) classification. For different classes $Y = y_1, \dots, y_n$, 1v1 classifies every binary class combination: y_1 vs y_2, y_1 vs y_3, y_2 vs y_3, etc. 1vR merges all but one class together and handles them as one class: y_1 vs $\{y_2, \dots, y_n\}$, y_2 vs $\{y_1, y_3, \dots, y_n\}$. The classification uses all eleven features, all stimuli combinations (for 1v1) or stimuli (for 1vR) and all five classifiers. However, we only have two classes for the light experiments (red light and blue light), so both variants yield the same results.

4.3 Results

We collected 296 individual experiment datasets over multiple weeks, 148 for red light and 148 for blue light, with a total number of about 3.8 million data points. Some of them had to be discarded due to explainable faults, e.g., people walking into the lab during experiments. The results are displayed in Figures 4.3 and 4.4. Every dataset depicts one continuous measurement and includes multiple experiment datasets. Interestingly, the datasets show considerable differences, even though all of them were collected using the same method. Datasets 3, 4, 5, and 7 show an easily observable pattern consisting of responses of alternating amplitude. The responses of dataset 7 are not only alternating in amplitude but even in their orientation. The responses in datasets 1, 2, and 6 are not as easy to identify: While periodic responses can be seen in dataset 6, they are similar in amplitude and do not allow easy differentiation between both stimuli. Responses in datasets 1 and 2 are weaker than in the others and the datasets show generally more chaotic behaviour. Certain datasets show an underlying trend: The electro potential baseline is not stable but changes over a longer time. The cause for this is still unclear, but we explore different presumptions in Section 4.4.

[16] Find the MATLAB Measures of Analysis of Time Series toolkit at: https://de.mathworks.com/matlabcentral/fileexchange/27561-measures-of-analysis-of-time-series-toolkit-mats

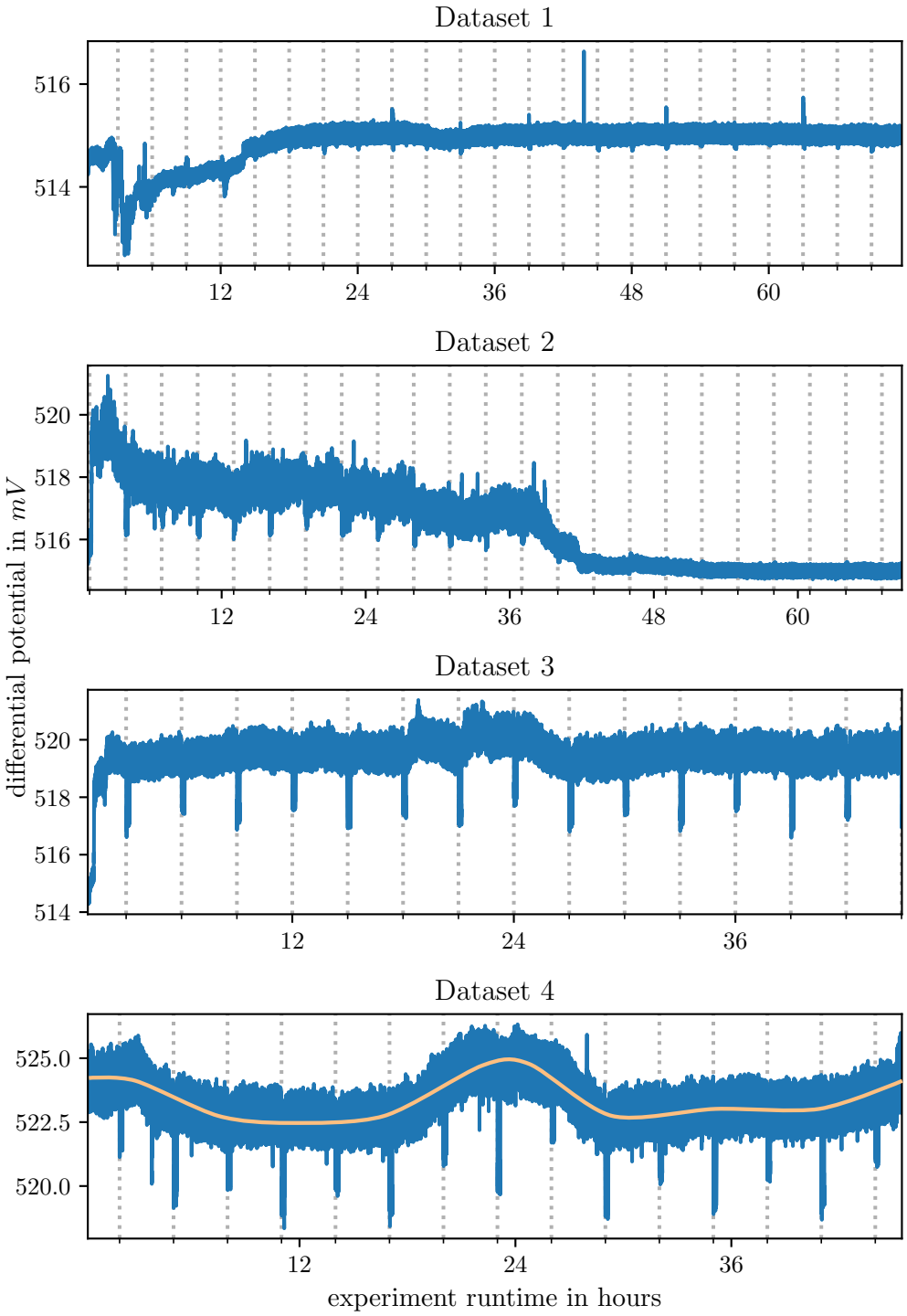

Figure 4.3: Datasets 1 to 4. The dotted lines mark stimuli applications for ten minutes in three-hour intervals. The orange line in dataset 4 is approximated to the signals electrical baseline. It is unstable and shows an underlying trend, but the dataset shows still clear reactions to the stimuli.

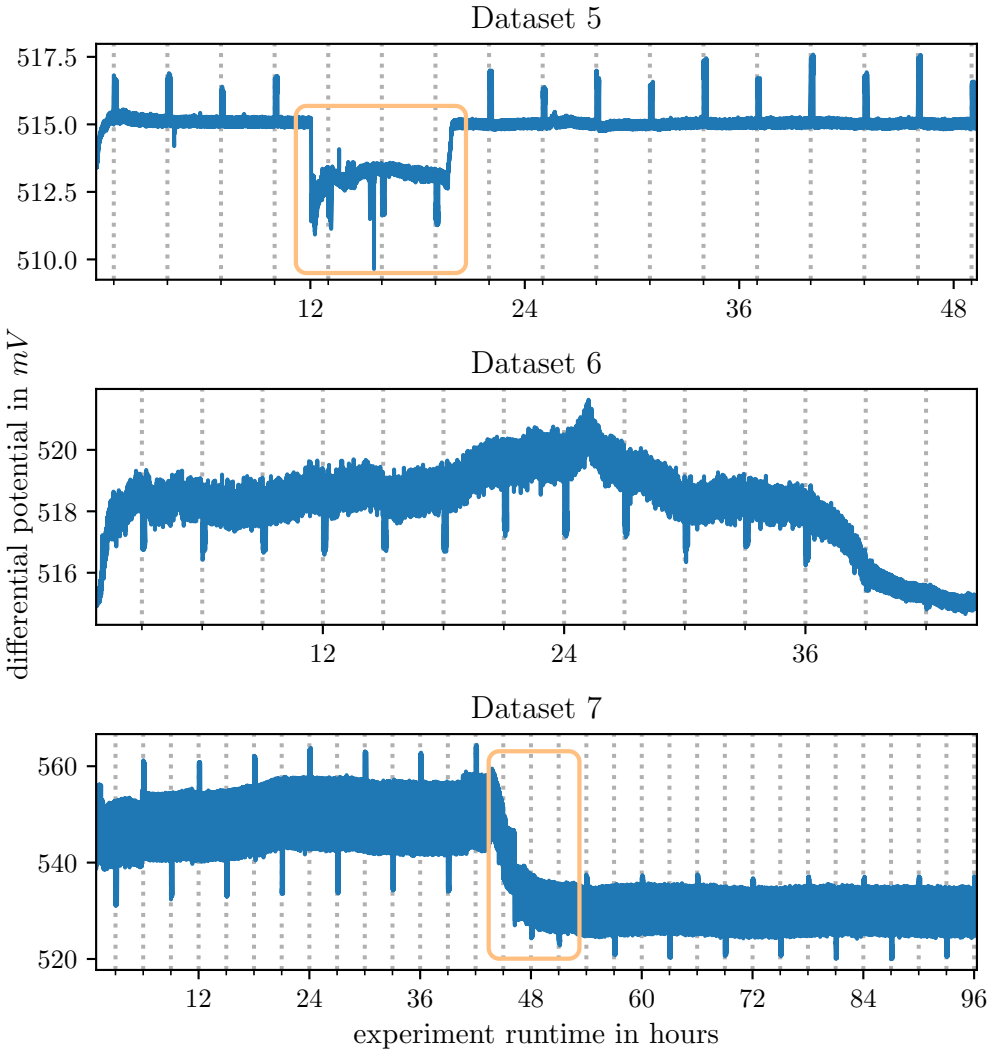

Figure 4.4: Datasets 5 to 7. The dotted lines mark stimuli applications for ten minutes in three-hour intervals. Marked in orange are drops in signals with otherwise stable electrical baselines. We don't know what caused them but discuss different hypotheses in Section 4.4.

4.3.1 Univariate Classification

Results for univariate classification using every feature and every classifier combination are shown in Table 4.5. The quality of the classification is evaluated mainly by the *accuracy* (total rate of correct classified values) but is supplemented with *sensitivity* and *specificity* (the rate of true positives and true negatives). The accuracy is a good measure because both classes are balanced and have the same significance regarding correct classifications. A big difference between sensitivity and specificity is a sign that one stimulus is more likely to be classified correctly than the other. The highest accuracy using univariate classification is provided by the Mahalanobis classifier using ASP but is with 0.6036 still too low for a reliable classification.

Because we can see quality differences in the datasets, the best possible accuracy for every individual dataset is worthwhile to investigate. Table 4.6 shows the best accuracy of about 75% for datasets 4 and 7, which corresponds with our manual evaluation of the plots. Five datasets achieve the best accuracy using the quadratic classifier, while the other two use the linear classifier. As every accuracy calculated with a quadratic classifier is better than the ones using a linear classifier, we can conclude that the quadratic classifier performs better if we consider individual datasets. We can also see in Table 4.6 that the best performing features are the ASP with three occurrences, followed by variance, Hjorth complexity, DFA, and mean with one occurrence. Interestingly, the average accuracy over all seven datasets using their respective best classifier and feature is higher than the accuracy for using all datasets together ($0.6586 > 0.6036$). This is an indicator that well-discriminating features or classifiers can vary between all seven datasets as the merging of multiple datasets yields worse results.

4.3.2 Multivariate Classification

After investigating the discrimination power of single features, we additionally examine the multivariate classification using all eleven features. As shown in Table 4.7, using all features leads to better results for most of the datasets. Again, datasets 4 and 7 show the highest accuracies with 85.93% and 80.82%, also again with the quadratic classifier. Neither the diagLinear nor the diagQuadratic classifier had the best accuracy for any of the datasets, so the assumption of independent features is unlikely. To see if any of the used features are redundant, we perform a sequential feature selection on datasets 4 and 7. Sequential feature selection starts with an empty set of features and consecutively adds one feature until the addition of more features does not further improve the accuracy. It is a greedy algorithm, so it always makes the locally optimal decision in each iteration. We can see in Table 4.8 that we reach the maximum accuracy of 89.63% for dataset 4 with 8 features, and the maximum accuracy of 80.82% for dataset 7 (Table 4.9) with 10 features. The addition of only a second feature improves the accuracy for dataset 4 from 75.56% to 82.59%. Because of the unexplainable differences between all datasets, we can not decide on a single best performing-feature or a specific best-performing subset of features. In addition, the maximum accuracy for dataset 4 using only eight features is higher than using all eleven features ($0.8963 > 0.8593$). That means that the addition

of more features can lead to a worse accuracy. However, from the five tested classifier variants, QDA seems to generally achieve the best results.

feature	classifier	accuracy	sensitivity	specificity
mean	linear	0.5597	0.5311	0.5874
	quadratic	0.5597	0.5320	0.5866
	Mahalanobis	0.5597	0.5311	0.5874
variance	linear	0.5593	0.7786	0.3464
	quadratic	0.5415	0.8005	0.2900
	Mahalanobis	0.5597	0.7786	0.3472
skewness	linear	0.5041	0.5278	0.4812
	quadratic	0.5228	0.4049	0.6373
	Mahalanobis	0.4444	0.5278	0.3636
kurtosis	linear	0.5104	0.3476	0.6683
	quadratic	0.5050	0.3190	0.6855
	Mahalanobis	0.5104	0.3476	0.6683
IQR	linear	0.5638	0.7399	0.3930
	quadratic	0.5589	0.7879	0.3366
	Mahalanobis	0.5647	0.7399	0.3946
Hjorth mobility	linear	0.5095	0.4983	0.5204
	quadratic	0.5000	0.5816	0.4208
	Mahalanobis	0.5095	0.4983	0.5204
Hjorth complexity	linear	0.5070	0.5042	0.5098
	quadratic	0.5087	0.3788	0.6348
	Mahalanobis	0.5070	0.5042	0.5098
Hurst exponent	linear	0.5000	0.3359	0.6593
	quadratic	0.4399	0.4108	0.4681
	Mahalanobis	0.4395	0.2130	0.6593
DFA	linear	0.5029	0.6987	0.3129
	quadratic	0.4942	0.8232	0.1748
	Mahalanobis	0.4473	0.5859	0.3129
WPE	linear	0.5597	0.5311	0.5874
	quadratic	0.5601	0.5328	0.5866
	Mahalanobis	0.5597	0.5311	0.5874
ASP	linear	0.6024	0.7500	0.4592
	quadratic	0.5891	0.7719	0.4118
	Mahalanobis	0.6036	0.7492	0.4624

Table 4.5: Accuracies, sensitivities and specificities for univariate blue and red light experiments, using every combination of features and classifiers.

dataset	feature	classifier	accuracy	sensitivity	specificity
1	variance	quadratic	0.6618	0.7731	0.5404
2	Hjorth complexity	linear	0.5707	0.5657	0.5758
3	ASP	quadratic	0.6984	0.8254	0.5714
4	ASP	quadratic	0.7556	0.8810	0.6458
5	DFA	linear	0.5852	0.3333	0.8056
6	ASP	quadratic	0.5873	0.6984	0.4762
7	mean	quadratic	0.7509	0.8296	0.6771
all	ASP	Mahalanobis	0.6036	0.7492	0.4624

Table 4.6: Best individual features for all collected datasets based on accuracy. The average accuracy for all seven datasets is 0.6586, which is higher than the actual accuracy for all datasets together.

dataset	classifier	accuracy	sensitivity	specificity
1	quadratic	0.7367	0.6759	0.8030
2	quadratic	0.6212	0.5859	0.6566
3	Mahalanobis	0.6667	0.6032	0.7302
4	quadratic	0.8593	0.8571	0.8611
5	Mahalanobis	0.6111	0.7540	0.4861
6	linear	0.6508	0.6349	0.6667
7	quadratic	0.8082	0.8182	0.7982
all	Mahalanobis	0.6235	0.6944	0.5547

Table 4.7: Best classifier for every dataset using multivariate classification with all eleven features. The average accuracy for all seven datasets is 0.7077, which is higher than the actual accuracy for all datasets together.

feature count	features used	accuracy	sensitivity	specificity
1	ASP	0.7556	0.8810	0.6458
2	+ IQR	0.8259	0.9365	0.7292
3	+ Hurst exponent	0.8407	0.9365	0.7569
4	+ DFA	0.8704	0.9286	0.8194
5	+ Hjorth mobility	0.8852	0.9444	0.8333
6	+ Hjorth complexity	0.8852	0.9286	0.8472
7	+ skewness	0.8926	0.9444	0.8472
8	+ kurtosis	0.8963	0.9444	0.8542
9	+ mean	0.8963	0.9127	0.8819

Table 4.8: Selected feature subsets using sequential feature selection and their accuracies, sensitivities and specificities of dataset 4.

feature count	features used	accuracy	sensitivity	specificity
3	mean, skewness, Hjorth complexity	0.7509	0.8296	0.6771
4	+ Hurst exponent	0.7509	0.8333	0.6736
5	+ Hjorth mobility	0.7599	0.8074	0.7153
6	+ variance	0.7778	0.8704	0.6910
7	+ kurtosis	0.7832	0.8593	0.7118
8	+ IQR	0.7885	0.8926	0.6910
9	+ WPE	0.8065	0.7815	0.8299
10	+ ASP	0.8082	0.8370	0.7812

Table 4.9: Selected feature subsets using sequential feature selection and their accuracies, sensitivities and specificities of dataset 7.

4.4 Discussion

We wanted to recreate the work from Chatterjee et al. [7] to use their classification approach for WatchPlant. While working on it, we noticed discrepancies in the mentioned paper. Even after communication with the authors, we were not able to recreate the results, despite using their published datasets and methods. In their experiments, they researched plant electrical responses to the stimuli ozone, sulfuric acid and sodium chloride. After we first ran experiments with the same stimuli, we decided to discontinue them once it was clear that we could not recreate the results. We might later resume the experiments if they are more in the focus of WatchPlant.

The classification of individual datasets resulted in better accuracies than classifying all datasets together. Even if we only classify datasets with good individual accuracies together, e.g., datasets 4 and 7, the accuracy of them (0.6981) is still worse than for both individual datasets (0.7556 and 0.7509). The same is true for multivariate classification using these datasets. Comparing the results of sequential feature selection for datasets 4 and 7 (Tables 4.8 and 4.9) shows different chosen features for multivariate classification, and both have different best-performing features for univariate classification, as seen in Table 4.6. It seems like the plant produced very different signals for both experiments, even if we tried to follow the guidelines listed in Section 3.3.

After the classification of light stimuli, there are still unanswered questions, especially regarding sudden drops or underlying trends in the electro potential response. For example, we can see a relatively stable baseline in dataset 5 in Figure 4.4, but for approximately nine hours (marked in orange in the plot), the potential dropped while still retaining the periodical reactions to light. In dataset 4 in Figure 4.3, we still can see the periodic responses, but the signal has no stable potential baseline: We can see an underlying trend (approximated in orange). These phenomena also occurred during other experiments with different stimuli and plants. We do not know why that happens yet but established three different theories: *Known stimuli interference*, *unknown stimuli interference* and *internal*

plant process interference. The lab we do experiments in is not completely shielded against outer stimuli. Maybe some stimuli we know plants can react to are influencing the experiments because we could not sufficiently shield the lab against them. We did not find any correlation to other stimuli, but we also do not have complete information about all stimuli we could have in the lab. Another possibility would be a yet unknown stimulus that influences the plant. If this is the case, we obviously can not seal the lab against it but can only hope to identify this new stimulus. The last theory views the change in the baseline as part of an internal, biological process of the plant. We do not know enough yet about the correlation between internal plant processes and what electrical responses they trigger, so this is an open biological research topic.

Dataset 7 in Figure 4.4 shows an interesting and variant behaviour if compared to the other datasets: The responses to the alternating blue and red light are also alternating in their orientation. The other datasets show only alternating amplitude, but the responses have the same orientation. The reason for this is unknown. While this leads to a good discrimination between both stimuli, it worsens the accuracy of the classification when used in combination with other datasets that do not have an alternating orientation. If we compare the other datasets with noticeable periodic responses, we can see alternating strong and weak responses. E.g., datasets 4 and 5 (Figures 4.3 and 4.4) both start with a weak response, but the first stimulus applied to dataset 4 was red light, while the first stimulus applied to dataset 5 was blue light. The light intensity was neither changed for blue nor for red light between both experiments, so because of an unknown reason, the plant reacts sometimes more to blue light, and other times more to red light. The reattachment of electrodes seems to play an important role if we want to measure similar responses and has to be further investigated.

5

Conclusion and Outlook

The objectives of this thesis were the development of a living sensor network, collection of light-stimuli datasets and classification of the light stimuli based on the methods of Chatterjee et al. [7]. The living sensor network as described in Section 3 was successfully deployed both at UzL and at our collaboration partners in Zagreb. After the initial test phase, it worked without any problems and recorded for a total time of more than 15.000 hours on multiple biohybrid sensor nodes. The network is extensible to use an arbitrary number of biohybrid sensor nodes and can be adjusted to new experiments through the quick and simple addition of further sensors. It will be of further use in the future to test other stimuli and plants and collect datasets for machine learning approaches.

A dataset of 3.8 million datapoints or about 1.000 hours was collected with alternating red and blue light stimuli with Zamioculcas plants. Although all experiments were conducted under the same circumstances, we can see quality differences between the datasets and yet unexplainable phenomena. Further research is needed to investigate the cause for underlying trends and drops in the electro potential baseline. This will help us to get a better understanding of biological processes inside plants and be useful for collecting more stable datasets.

One of the most challenging aspects while running experiments was to find suitable conditions for collecting datasets with clear responses. Section 3.3 describes some guidelines, but they are only empirically verified and are not based on actual scientific evidence. Exploring the parts of plant physiology where the electro potential is the strongest and can be well measured could improve the electrode placement, therefore improving the measurements.

Developing a framework for feature extraction and classification proved to be difficult. To find the mathematical definitions for the used MATLAB-classifiers, we partially had to "reverse-engineer" them from the MATLAB source code. Controlling the correctness of our implementation by comparing our extracted features with the results of the PLEASED paper [7] was also not possible due to its discrepancies. Despite these difficulties, we implemented feature extraction and preprocessing in Python and classification in MATLAB. The scripts can be used in the future for other datasets and are easy to adapt to e.g., include other features or use other statistical classifiers.

The manual evaluation of the collected red and blue light datasets yields quite some variation. Some show easily recognizable periodic responses (e.g., dataset 7 in Figure 4.4), while others do not allow us to identify responses (e.g., dataset 2 in Figure 4.3, especially the second half). This evaluation matches the classification results: If we classify all datasets together, we can identify ASP as the best feature with an accuracy of 60.36%. Individual classification of the datasets results in accuracies between 57.07% and 75.56%. Using multivariate classification with all eleven features, we can improve these results to range from 61.11% to 85.93%. We used sequential feature selection on the best performing datasets, datasets 4 and 7, to find the most important features. The stimuli of these datasets can be classified with accuracies of 89.63% and 80.82% using eight respectively ten of eleven features. QDA performs best for both datasets, but we can not find a consistent, best discriminating feature or feature combination.

These results prove that we can successfully use feature extraction paired with simple classifiers to distinguish between red and blue light stimuli using the electrophysiological response of plants. However, every dataset behaves differently and the combination of multiple datasets for classification leads to worse results. We do not know the reason for the differences between the individual datasets as all of them were collected in the same way. Some datasets show additional electrical responses, e.g., the drop in dataset 5 in Figure 4.4, that we can not explain. They may be the reaction to stimuli we could not control during the experiments or stem from internal biological processes. To sum up, we can conclude that plants are still a black box for us: We understand some reactions, but further research is needed to really understand the electrophysiology of plants.

5.1 Outlook

As we still do not know much about the plants reactions, many additional experiments are conceivable. We can use different plant species and compare their reactions to find species that are more sensitive to specific stimuli that we want to measure. Right now all experiments take place in a lab environment. Testing different outdoor plants that naturally grow in urban environments can show us suitable candidates for WatchPlant.

Long-term experiments with different plants could give us information about potential electrical responses that are created through the process of growing or are the result of internal, non-visible processes. Other stimuli can be tested, e.g., temperature, air humidity, physical stimulation like wind or touch, magnetism, sound, etc. Coupled with long-term experiments, long-lasting stimuli that do not only have an immediate effect on the plant like ozone or carbon dioxide can be observed.

To measure the electro potentials, we had to insert silver electrodes inside the plant. Plants try to repair the damage caused by the insertion through lignification around the electrode, which renders the measurements unusable. Surface electrodes could be attached to the leaves without damaging the plant. The investigation of their usability for measuring plant electrical responses might show us a better alternative than using insertion electrodes.

Regarding classification methods, it may be interesting to compare the results to other classifiers and other features. Feature extraction can also be useful for machine learning approaches as a method of dimensionality reduction to optimize the needed processing power. A low amount of needed processing power is especially important for WatchPlant's self-sufficient, low powered sensor nodes.

In the context of energy consumption, we should try to replace the sensor nodes with something smaller and more energy-efficient. The MU and Raspberry Pi work well for research but for the final design of sensor nodes, a more minimalistic approach should be used. Depending on how successful the energy production from phloem sap is, solar energy is also conceivable for powering the sensor nodes.

Bibliography

[1] *WatchPlant project website*. Available online: https://watchplantproject.eu/ (accessed in December 2021).

[2] Whippo, C. W. and Hangarter, R. P. Phototropism: Bending towards Enlightenment. In: *The Plant Cell* 18(5):1110–1119, 2006.

[3] Volkov, A. G. and Markin, V. S. Phytosensors and Phytoactuators. In: *Plant Electrophysiology*. Springer, 2012, pp. 173–206.

[4] *Smart Biohybrid Phyto-Organisms for Environmental In Situ Monitoring*. Available online: https://cordis.europa.eu/project/id/101017899 (accessed in December 2021).

[5] Hamann, H., Bogdan, S., Diaz-Espejo, A., García-Carmona, L., Hernandez-Santana, V., Kernbach, S., Kernbach, A., Quijano-López, A., Salamat, B., and Wahby, M. WatchPlant: Networked Bio-hybrid Systems for Pollution Monitoring of Urban Areas. In: *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press. 2021.

[6] García-Carmona, L., Bogdan, S., Diaz-Espejo, A., Dobielewski, M., Hamann, H., Hernandez-Santana, V., Kernbach, A., Kernbach, S., Quijano-López, A., Roxhed, N., et al. Biohybrid systems for environmental intelligence on living plants: WatchPlant project. In: *Proceedings of the Conference on Information Technology for Social Good*. 2021, pp. 210–215.

[7] Chatterjee, S. K., Das, S., Maharatna, K., Masi, E., Santopolo, L., Mancuso, S., and Vitaletti, A. Exploring Strategies for Classification of External Stimuli Using Statistical Features of the Plant Electrical Response. In: *Journal of the Royal Society Interface* 12(104):20141225, 2015.

[8] Romano, D., Donati, E., Benelli, G., and Stefanini, C. A review on animal–robot interaction: From bio-hybrid organisms to mixed societies. In: *Biological Cybernetics* 113(3):201–225, 2019.

[9] Whitmire, E., Latif, T., and Bozkurt, A. Acoustic Sensors for Biobotic Search and Rescue. In: *SENSORS*. IEEE. 2014, pp. 2195–2198.

[10] Dirafzoon, A. and Lobaton, E. Topological Mapping of Unknown Environments using an Unlocalized Robotic Swarm. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 5545–5551.

[11] Bozkurt, A, Gilmour, R, Stern, D, and Lal, A MEMS based bioelectronic neuromuscular interfaces for insect cyborg flight control. In: *2008 IEEE 21st International Conference on Micro Electro Mechanical Systems*. IEEE. 2008, pp. 160–163.

[12] Zhang, C., Cao, F., Li, Y., and Sato, H. Fuzzy-Controlled Living Insect Legged Actuator. In: *Sensors and Actuators A: Physical* 242:182–194, 2016.

[13] Holzer, R. and Shimoyama, I. Locomotion Control of a Bio-Robotic System via Electric Stimulation. In: *Proceedings of the 1997 IEEE/RSJ International Conference on*

Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS'97. Vol. 3. IEEE. 1997, pp. 1514–1519.

[14] Li, G. and Zhang, D. Brain-Computer Interface Controlling Cyborg: A Functional Brain-to-Brain Interface Between Human and Cockroach. In: *Brain-Computer Interface Research*. Springer, 2017, pp. 71–79.

[15] Kobayashi, N., Yoshida, M., Matsumoto, N., and Uematsu, K. Artificial control of swimming in goldfish by brain stimulation: Confirmation of the midbrain nuclei as the swimming center. In: *Neuroscience Letters* 452(1):42–46, 2009.

[16] Wenbo, W., Ce, G., Jiurong, S., and Zhendong, D. Locomotion Elicited by Electrical Stimulation in the Midbrain of the Lizard Gekko gecko. In: *Intelligent Unmanned Systems: Theory and Applications*. Springer, 2009, pp. 145–153.

[17] Wang, Y., Lu, M., Wu, Z., Tian, L., Xu, K., Zheng, X., and Pan, G. Visual Cue-Guided Rat Cyborg for Automatic Navigation. In: *IEEE Computational Intelligence Magazine* 10(2):42–52, 2015.

[18] Halloy, J., Mondada, F., Kernbach, S., and Schmickl, T. Towards Bio-hybrid Systems Made of Social Animals and Robots. In: *Conference on Biomimetic and Biohybrid Systems*. Springer. 2013, pp. 384–386.

[19] Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tâche, F., Saïd, I., Durier, V., Canonge, S., Amé, J. M., et al. Social Integration of Robots into Groups of Cockroaches to Control Self-Organized Choices. In: *Science* 318(5853):1155–1158, 2007.

[20] Taylor, R. C., Klein, B. A., Stein, J., and Ryan, M. J. Faux frogs: multimodal signalling and the value of robotics in animal behaviour. In: *Animal Behaviour*, 2008.

[21] Abaid, N., Marras, S., Fitzgibbons, C., and Porfiri, M. Modulation of risk-taking behaviour in golden shiners (Notemigonus crysoleucas) using robotic fish. In: *Behavioural Processes* (100):9–12, 2013.

[22] Benelli, G., Romano, D., Rocchigiani, G., Caselli, A., Mancianti, F., Canale, A., and Stefanini, C. Behavioral asymmetries in ticks–Lateralized questing of Ixodes ricinus to a mechatronic apparatus delivering host-borne cues. In: *Acta Tropica* 178:176–181, 2018.

[23] Stewart, C. N., Abudayyeh, R. K., and Stewart, S. G. Houseplants as home health monitors. In: *Science* 361(6399):229–230, 2018.

[24] Laura, S. B. and Şumălan, R Plant growth and development monitoring through integrated sensor systems. In: *Journal of Horticulture, Forestry and Biotechnology* 17(1):307–311, 2013.

[25] *World Population Projected to Reach 9.8 Billion in 2050, and 11.2 Billion in 2100*. Available online: `https://www.un.org/development/desa/en/news/population/world-population-prospects-2017.html` (accessed in December 2021).

[26] *flora robotica project website*. Available online: `https://www.florarobotica.eu/` (accessed in December 2021).

[27] Heinrich, M. K., Mammen, S. von, Hofstadler, D. N., Wahby, M., Zahadat, P., Skrzypczak, T., Soorati, M. D., Krela, R., Kwiatkowski, W., Schmickl, T., et al. Constructing living buildings: a review of relevant technologies for a novel application of biohybrid robotics. In: *Journal of the Royal Society Interface* 16(156):20190238, 2019.

[28] Hamann, H., Wahby, M., Schmickl, T., Zahadat, P., Hofstadler, D., Stoy, K., Risi, S., Faina, A., Veenstra, F., Kernbach, S., et al. Flora Robotica-Mixed Societies of Symbiotic Robot-Plant Bio-Hybrids. In: *2015 IEEE Symposium Series on Computational Intelligence*. IEEE. 2015, pp. 1102–1109.

[29] Hamann, H., Soorati, M. D., Heinrich, M. K., Hofstadler, D. N., Kuksin, I., Veenstra, F., Wahby, M., Nielsen, S. A., Risi, S., Skrzypczak, T., et al. Flora robotica–An Architectural System Combining Living Natural Plants and Distributed Robots. In: *Living Architectures Workshop 2017: A workshop of ECAL, the European Conference on Artificial Life*. MIT Press. 2017.

[30] Wahby, M., Hofstadler, D. N., Heinrich, M. K., Zahadat, P., and Hamann, H. An Evolutionary Robotics Approach to the Control of Plant Growth and Motion: Modeling Plants and Crossing the Reality Gap. In: *2016 IEEE 10th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE. 2016, pp. 21–30.

[31] Hofstadler, D. N., Wahby, M., Heinrich, M. K., Hamann, H., Zahadat, P., Ayres, P., and Schmickl, T. Evolved Control of Natural Plants: Crossing the Reality Gap for User-Defined Steering of Growth and Motion. In: *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 12(3):1–24, 2017.

[32] *PLEASED project website*. Available online: `https://pleased-fp7.eu/` (accessed in December 2021).

[33] Chatterjee, S. K., Malik, O., and Gupta, S. Chemical Sensing Employing Plant Electrical Signal Response-Classification of Stimuli Using Curve Fitting Coefficients as Features. In: *Biosensors* 8(3):83, 2018.

[34] Teng, H., Kok, B., Uttraphan, C., and Yee, M. A Review on Energy Harvesting Potential from Living Plants: Future Energy Resource. In: *International Journal of Renewable Energy Research* 8(4):2598–2614, 2018.

[35] Oy, S. A. and Özdemir, A. E. Usage of Piezoelectric Material and Generating Electricity. In: *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE. 2016, pp. 63–66.

[36] Calogero, G. and Di Marco, G. Red Sicilian orange and purple eggplant fruits as natural sensitizers for dye-sensitized solar cells. In: *Solar Energy Materials and Solar Cells* 92(11):1341–1346, 2008.

[37] Strik, D. P., Hamelers, H., Snel, J. F., and Buisman, C. J. Green electricity production with living plants and bacteria in a fuel cell. In: *International Journal of Energy Research* 32(9):870–876, 2008.

[38] Patel, N. Screening of Mutant Arabidopsis Thaliana and Chlamydomonas Reinhardtii for their Potential Use as Phytosensors in 2, 4, 6 Trinitrotoluene (TNT) Contaminated Environments. MA thesis. University of Tennessee, 2003.

[39] Stewart Jr, C. N. and Liu, W. Deployable Phytosensors for Plant Pathogen Detection. In: *ISB News Report*, 2013.

[40] Wiersma, G. B. *Environmental Monitoring*. CRC press, 2004.

[41] Zhang, Y., Gu, Y., Vlatkovic, V., and Wang, X. Progress of smart sensor and smart sensor networks. In: *Fifth World Congress on Intelligent Control and Automation*. Vol. 4. IEEE. 2004, pp. 3600–3606.

[42] Frank, R. *Understanding Smart Sensors*. Artech House, 2013.

[43] Martinez, K., Hart, J. K., and Ong, R. Environmental Sensor Networks. In: *Computer* 37(8):50–56, 2004.

[44] *CORDIS Environmental Intelligence programme*. Available online: https://cordis.europa.eu/programme/id/H2020_FETPROACT-EIC-08-2020 (accessed in December 2021).

[45] *I-Seed project website*. Available online: https://iseedproject.eu/ (accessed in December 2021).

[46] Hjorth, B Time domain descriptors and their relation to a particular model for generation of EEG activity. In: *CEAN-Computerized EEG analysis*:3–8, 1975.

[47] Hjorth, B. EEG analysis based on time domain properties. In: *Electroencephalography and Clinical Neurophysiology* 29(3):306–310, 1970.

[48] Hurst, H. E. Long-Term Storage Capacity of Reservoirs. In: *Transactions of the American society of Civil Engineers* 116(1):770–799, 1951.

[49] Mandelbrot, B. B. and Wallis, J. R. Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence. In: *Water Resources Research* 5(5):967–988, 1969.

[50] Peng, C.-K., Buldyrev, S. V., Havlin, S., Simons, M., Stanley, H. E., and Goldberger, A. L. Mosaic organization of DNA nucleotides. In: *Physical Review E* 49(2):1685, 1994.

[51] Lee, J.-M., Kim, D.-J., Kim, I.-Y., Park, K.-S., and Kim, S. I. Detrended fluctuation analysis of EEG in sleep apnea using MIT/BIH polysomnography data. In: *Computers in Biology and Medicine* 32(1):37–47, 2002.

[52] Ting, W., Guo-Zheng, Y., Bang-Hua, Y., and Hong, S. EEG feature extraction based on wavelet packet decomposition for brain computer interface. In: *Measurement* 41(6):618–625, 2008.

[53] Welch, P. T h e Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms. In: *IEEE Transactions on Audio and Electroacoustics* 15(2):70–73, 1967.

[54] Theodoridis, S., Pikrakis, A., Koutroumbas, K., and Cavouras, D. *Introduction to Pattern Recognition: A MATLAB Approach*. Academic Press, 2010.

[55] Krzanowski, W. *Principles of Multivariate Analysis*. Vol. 23. OUP Oxford, 2000.

[56] Seber, G. A. *Multivariate Observations*. Vol. 252. John Wiley & Sons, 2009.

[57] Alpaydin, E. *Introduction to Machine Learning*. MIT press, 2020.

[58] Hunter, J. D. Matplotlib: A 2D graphics environment. In: *Computing in Science & Engineering* 9(3):90–95, 2007.

[59] Liscum, E., Askinosie, S. K., Leuchtman, D. L., Morrow, J., Willenburg, K. T., and Coats, D. R. Phototropism: Growing towards an Understanding of Plant Movement. In: *The Plant Cell* 26(1):38–55, 2014.

[60] Hogewoning, S. W., Wientjes, E., Douwstra, P., Trouwborst, G., Van Ieperen, W., Croce, R., and Harbinson, J. Photosynthetic Quantum Yield Dynamics: From Photosystems to Leaves. In: *The Plant Cell* 24(5):1921–1935, 2012.

[61] Wahby, M., Heinrich, M. K., Hofstadler, D. N., Neufeld, E., Kuksin, I., Zahadat, P., Schmickl, T., Ayres, P., and Hamann, H. Autonomously shaping natural climbing plants: a bio-hybrid approach. In: *Royal Society Open Science* 5(10):180296, 2018.