IOP Publishing Bioinspir. Biomim. 18 (2023) 025003 https://doi.org/10.1088/1748-3190/acbad2

Bioinspiration & Biomimetics

OPEN ACCESS

RECEIVED
30 September 2022

REVISED
5 February 2023

ACCEPTED FOR PUBLICATION
9 February 2023

PUBLISHED
7 March 2023

PAPER

Stimulus classification with electrical potential and impedance of living plants: comparing discriminant analysis and deep-learning methods

Eduard Buss[1,2,*], **Till Aust**[1,2], **Mostafa Wahby**[1], **Tim-Lucas Rabbel**[1], **Serge Kernbach**[3] and **Heiko Hamann**[2]

[1] Institute of Computer Engineering, University of Lübeck, Lübeck, Germany
[2] Department of Computer and Information Science, University of Konstanz, Konstanz, Germany
[3] CYBRES GmbH, Research Center of Advanced Robotics and Environmental Science, Stuttgart, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: eduard.buss@uni-konstanz.de

Keywords: electrophysiology, electrical potential, tissue electrical impedance, phytosensing, discriminant analysis, artificial neural networks, time series classification

Abstract

The physiology of living organisms, such as living plants, is complex and particularly difficult to understand on a macroscopic, organism-holistic level. Among the many options for studying plant physiology, electrical potential and tissue impedance are arguably simple measurement techniques that can be used to gather plant-level information. Despite the many possible uses, our research is exclusively driven by the idea of phytosensing, that is, interpreting living plants' signals to gather information about surrounding environmental conditions. As ready-to-use plant-level physiological models are not available, we consider the plant as a blackbox and apply statistics and machine learning to automatically interpret measured signals. In simple plant experiments, we expose *Zamioculcas zamiifolia* and *Solanum lycopersicum* (tomato) to four different stimuli: wind, heat, red light and blue light. We measure electrical potential and tissue impedance signals. Given these signals, we evaluate a large variety of methods from statistical discriminant analysis and from deep learning, for the classification problem of determining the stimulus to which the plant was exposed. We identify a set of methods that successfully classify stimuli with good accuracy, without a clear winner. The statistical approach is competitive, partially depending on data availability for the machine learning approach. Our extensive results show the feasibility of the blackbox approach and can be used in future research to select appropriate classifier techniques for a given use case. In our own future research, we will exploit these methods to derive a phytosensing approach to monitoring air pollution in urban areas.

1. Introduction

One key to understanding living organisms, and life as such, is to analyze their physiology. A good understanding of plant physiology can help us grow healthier crop plants (i.e. by providing all the required resources) more efficiently (i.e. by providing only the necessary resources) [1]. For example, Fromm and Fei 1998 [2] report that root to shoot electric communication includes information about soil moisture. Our main motivation, however, is driven by an engineering perspective, as we ultimately want to use measurements of plant physiology to implement a so-called phytosensing approach [3, 4]. Phytosensing is the idea of using living plants as sensors by interpreting their internal signals (more details below). In plant science one would hope for fundamental insights once plant physiology is fully understood. As in all natural sciences, an essential step of the experimental approach is measurement. There are many options for measuring a plant's physiology. Here, we focus on electrical potential and tissue electrical impedance for a number of reasons. Plants generate different types of extracellular

electrical responses in reaction to environmental stresses [5, 6].

In electrophysiology a differential electrical potential of the plant is measured [7, 8]. Changes in a plant's electrical potential can be detected for different stimuli, such as heat, light, humidity, CO_2 level, herbivory, and salt stress [3, 7]. The measured electric potential in the plant is subject to an uneven distribution of ions that are also actively and passively transported through the plant (see section 2 for a few more details). This method can be intrusive as electrodes are inserted into the plant's tissue (as done here). Less intrusive techniques are also established by recording surface potentials of plant leaves [9].

To measure tissue electrical impedance, a small alternating current is applied to probe, in our case, the plant tissue's resistance and reactance characteristics [10]. An advantage of both electrical potential and impedance is their relatively simple application because only two electrodes and simple electronics are required to measure and process the signal. We measure the differential potential using electrodes that are inserted into plants a few centimeters apart. This macroscopic property complicates the interpretation of the signal. While previous experiments have shown that there are clearly measurable effects, such as a reaction of the plant to light, the scientific interpretation of the obtained data in terms of the underlying molecular mechanisms is still challenging [11]. Hence, the interpretation of data obtained using plant electrophysiology is complex, and methods to improve our understanding of the signals need to be further explored. Here, we assume the plant to be a blackbox and we automatically interpret the electrical potential or impedance signal using methods of statistics and machine learning.

Essential for our approach is to gather training data in controlled plant experiments that can explicitly be labeled by a respectively applied stimulus (e.g. heat or light color). The required amount of data and plant experiments is generally unknown, but especially for methods of deep learning, more data usually means more accuracy [12]. This poses challenges due to the relatively high costs of plant experiments (hardware/space requirements of grow box setups and plant health maintenance overheads, see [13]) compared to data collection in other applications of machine learning. Although we gather data from many plant experiments, at least a few of our tested methods may still suffer from lack of data (also see discussion in section 4.3).

In previous work, we designed an energy-efficient sensor node called *PhytoNode* for plant electrical potential (not used here). It can run classification models on board and interact, for example, with handheld devices of close-by citizens via Bluetooth Low Energy [14]. In this previous work, we exposed *Zamioculcas Zamiifolia* plants to four stimuli: wind, heat, blue light, and red light for data collection. We trained ten different artificial neural network (ANN) classifiers based on the machine learning framework by Fawaz *et al* [15]. Our classifiers based on convolutional neural networks (CNNs) achieved the best accuracies. Our binary classifier (ANN architecture: Residual Network) achieved 98.1% accuracy and our five-class network (ANN architecture: Encoder) achieved up to 86.0% accuracy. We ran our best classifiers on the embedded hardware and performed on-board classifications, which were disseminated to a mobile phone. Our main goal is to find accurate energy-efficient stimuli classifiers that can run on our PhytoNodes. Therefore, we extend the previous work by training five discriminant analysis classifiers (linear, quadratic, linear naive Bayes, quadratic naive Bayes, and Mahalanobis). We also retrain the ten deep-learning classifiers on the raw electrical potential data of only stimulus application intervals (see section 4.3). We compare all these approaches in terms of stimulus classification accuracy.

Our main motivation for this research is to exploit our results presented here to implement a phytosensing network. In our EU-funded project 'WatchPlant' [16, 17] (2021–2024), we use phytosensing to measure environmental features via living plants. We develop a self-powered, energy-efficient sensor network of electronic devices attached to plants [14, 18]. This forms a so-called bio-hybrid system of strongly coupled technological devices and living organisms. Such a bio-hybrid network can implement, for example, long-term monitoring and prediction of environmental conditions (e.g. air pollution) in urban settings. By measuring the physiology of living plants in cities, we may be able to infer causing stimuli (e.g. particulate matter, increased ozone concentrations). Measured stress reactions in plants over periods of weeks or months in a given urban area may even be correlated with the health data of citizens. A distributed bio-hybrid system powered by electrical potential, tissue impedance, and possibly additional means of measuring plant physiology may be used in the future to implement an early warning system about urban health hazards.

2. Related work

Research on electrical activity in plants started in 1873 by Burdon-Sanderson [19]. He conducted experiments on the carnivorous Venus flytrap (*Dionaea muscipula*), which is now a well-known example of action potentials [20]. Once an insect touches the sensitive hair of the trap leaves twice, action potentials are triggered, prompting an electrical impulse that leads to one of the fastest movements known in plants (100 ms), that is, the trap leaves close quickly. Electrical signals allow for rapid transmission of information compared, for example, to chemical signals. They are responsible for triggering plant-specific processes, including respiration, water uptake, leaf movement,

biotic stress response, turgor pressure reduction, or photosynthesis changes. They also form the initial plant reaction to external stimuli (e.g. environmental changes) [21], and allow for internal communication between plant cells or organs. Depending on the species, plants have a resting potential of between -80 mV and -200 mV, which is caused by a higher negative ion concentration inside the cell [22]. This unbalanced distribution of K^+, Na^+, Ca^{2+}, and Cl^- ions is separated by a plasma membrane. Variations of the potential are caused by active (e.g. H^+-ATPase) and passive (e.g. diffusion) transport mechanisms.

As discussed above, raw measured electrical potential time series are not easily interpretable. However, we are able to automatically interpret those signals by exploiting methods of pattern recognition and time series classification (TSC) [23, 24]. In the common approach of discriminant analysis, discriminative statistical features (e.g. variance, skewness) are calculated to form a feature space that is used to distinguish between different classes of stimuli. For example, Najdenovska *et al* [25] recorded the electrical potential response of tomato plants before and after infestation by spider mites. They extracted up to 34 different features from measured time series and classified them with 80% accuracy using the gradient-boosted tree algorithm in a binary classification problem.

Chatterjee *et al* [4] implemented a similar approach by exposing tomato plants to various stimuli. They measured the electrical potentials of tomato plants in response to sodium chloride (5 ml and 10 ml, 3 mol), sulfuric acid (5 ml, 0.05 mol), and ozone (16 ppm min^{-1}). The authors extracted eleven features from the data and used them to train five discriminant analysis classifiers, including linear, quadratic, naive Bayes, and Mahalanobis classifiers. They trained classifiers for different classification problems (e.g. one-vs-one and one-vs-rest) with varying numbers of features. Using quadratic discriminant analysis (QDA), they achieved up to 95% accuracy for a binary classification of sulfuric acid and ozone. Our discriminant analysis approach for classification is based on this work.

In another approach, Reissig *et al* [26] classified the ripening stage of tomatoes based on the electrical activity of the fruit. They classified the ripening stages based on three levels: mature green, breaker (<10% of the fruit surface is not green), and light red. They used principal component analysis based on features gained by approximate entropy, Fourier transformation, PSD, and wavelets. The gathered principal components were used to train decision trees, support vector machines, Gaussian processes, k-neighbors, random forests, Gaussian naive Bayes, and dummy classifiers. The dummy classifier serves as the baseline for the aforementioned classifiers as it uses several unintelligent ways to classify, for instance, assigning all new samples to the class with the highest prior probability. The three-class support vector machine classifier achieved the highest accuracy of 74.4%.

Another approach for such classification problems is training ANN classifiers. Generally, one has two options. The ANN classifiers can either be trained using the raw electrical potential data or statistical features can be extracted that serve as input to the ANN. Qin *et al* [27] used raw electrical potential data and investigated the salt tolerance of wheat seedlings. As mentioned above, training ANNs requires a large amount of data for training the weights [12], which can be challenging in plant science [13]. Facing this challenge, Qin *et al* used a conditional generative adversarial network to generate 200 artificial samples based on 127 living plant experiments. Using this data they trained a 1D CNN that achieved 92.3% accuracy.

In another example, Pereira *et al* [28] exposed soybean plants to three different stimuli: cold, low light, and osmotic stress. They followed two approaches to train five machine learning classifiers: ANNs, CNNs, optimum-path forest, k-nearest neighbors, and support vector machine. In the first approach (called interval arithmetic), they separate the raw data into intervals of certain length. They calculate each interval's minimum, average, and maximum as features. They obtain a chronological array of triplets as a training dataset. In the second approach, they used visual rhythm encoding to generate images from the raw data that were then used to train CNN classifiers. The authors reached accuracies of up to 90.7% with support vector machines in a binary class classification and up to 71.2% in multi-class classifications with k-nearest neighbors. Interestingly, the authors concluded that deep learning techniques are less efficient (i.e. lower accuracies).

3. Methods

Our main goal is to interpret living plant responses to various environmental stimuli. Therefore, we attach our electrical potential sensors to *Zamioculcas zamiifolia* and impedance sensors to *Solanum lycopersicum* in controlled experiment setups, as described in section 3.1. We expose the plants to a single stimulus at a time in multiple data collection plant experiments (see section 3.2). We use the collected data to train stimulus classifiers following two different approaches: discriminant analysis (see section 3.3) and deep learning (see section 3.4).

3.1. Bio-hybrid experiment setups

In our first indoor experimental setup, we measure the electrical potential of 'ZZ' plants (*Zamioculcas zamiifolia*, family Araceae), which are stemless forest plants with dark green glossy leaves [29] (see figure 1(a)). To avoid stressing the plants, we alternate the experiments among two 'ZZ' plants. The 'ZZ' plants show fast electrical responses

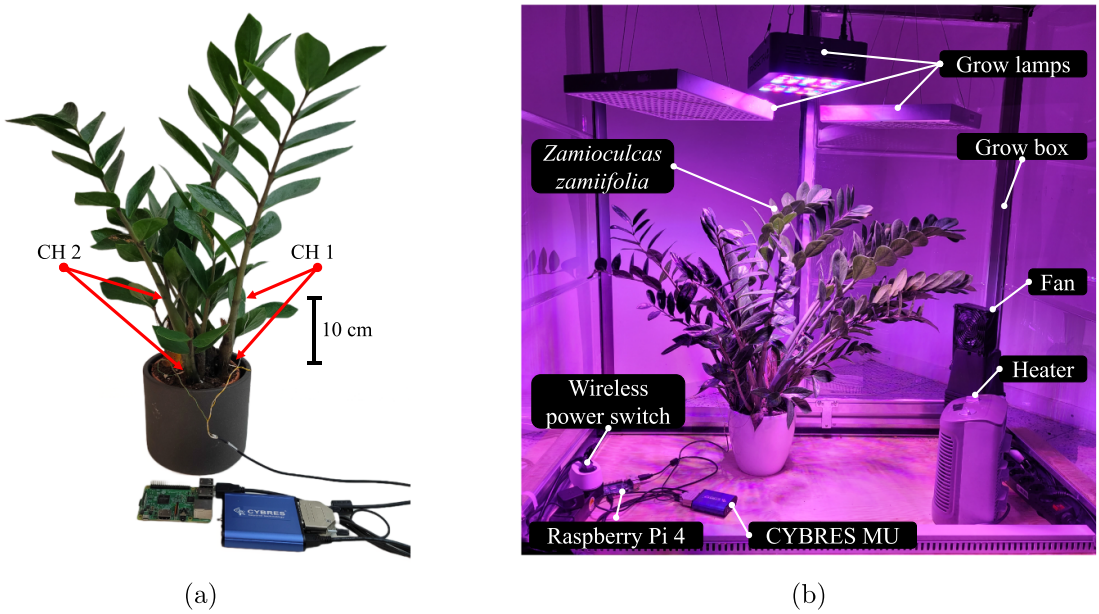

Figure 1. Experimental setup for electrical potential measurements. (a) Electrode placement of the two electrical potential channels (CH1 and CH2) of the CYBRES phytosensor on the *Zamioculcas zamiifolia*. (b) Complete experimental setup with the CYBRES phytosensor, attached to 'ZZ' plant, grow lamps, a fan, a heater, a wireless power switch, a Raspberry Pi 4, and a power supply.

to various stimuli and are robust to grow in technical laboratories without excessive maintenance (i.e. non-greenhouse condition). We minimized external environmental influences by conducting our experiments in a grow box of size 120 cm × 60 cm × 160 cm (L × W × H), see figure 1(b) and (fully) controlled the light conditions (no natural light). Our setup consists of two 'ZZ' plants to which we can apply four different stimuli: wind, heat, red light, and blue light. The wind stimulus is generated by a fan (Nidec V34809) placed 60 cm from the plant pot and at a height of 30 cm, generating a wind speed of approximately 9.2 $\frac{m}{s}$. We placed a heater (ROWENTA Mod.S02220F0) at a distance of 10 cm from the plant pot to increase the temperature inside the grow box (25.9 °C on average) by 2.3 °C on average during stimulus application. For both red and blue light stimuli, we use a total of six 45 W 'Erligpowht' LED grow lamps and place them around the plant. An 'Erligpowht' grow lamp contains 225 LEDs (165 red and 60 blue) with peak emissions λ_{max} at wavelengths 650 nm and 465 nm, respectively. We concealed the blue LEDs of three lamps for the red light stimulus. Similarly, we concealed the red LEDs of the remaining three lamps for the blue light stimulus. Two lamps are placed 30 cm above the plant. Two of each wavelength are placed on both sides 30 cm from the plant. In summary, we have one blue and one red lamp on top as well as two red and two blue lamps on the sides. To minimize human intervention, we use programmable TP-Link HS 100 power sockets to automate the experiments. The electrical activity of the plant is recorded using the CYBRES phytosensor [30] in combination with a Raspberry Pi 4, which stores and uploads the measured data. We insert the two silver electrodes (Ag99) of each electrical potential channel of the CYBRES phytosensor 10 cm apart into the lower end of one thickened petiole of the 'ZZ' plant, see figure 1(a). Preliminary control experiment results indicate negligible inference on the measured signals from all used electrical devices.

In our second experimental setup, we measure the tissue impedance of three tomato plants (*Solanum lycopersicum*, family Solanaceae) in response to light (combination of red/blue LEDs as used in the first experimental setup for electrical potential measurements, placed above the plants at about 0.5 m). To avoid stressing the plants, we alternate the experiments among three tomato plants. We apply the tissue impedance in the stem areas which is an example of an electrochemical sap flow measurement. Given that sap flow depends on many different parameters (e.g. irrigation cycles, leaf transpiration, cyclic hour-day rhythms, etc [31]), we automatically trigger the environmental stimuli (e.g. light and irrigation) at fixed times with one experiment per day. Here, we use the two impedance channels of the CYBRES phytosensor to measure the sap flow of the plant [32], see figure 2. The electrodes of each channel are inserted 2 mm into phloem/xylem vascular tissue [33] of the low stem area at distances of about 1 cm between electrodes respectively, see figure 2. Typical tissue RMS-impedance is between 40 kΩ and 60 kΩ under these conditions. Due to electrochemical measurements, the variation of penetration depth and position is

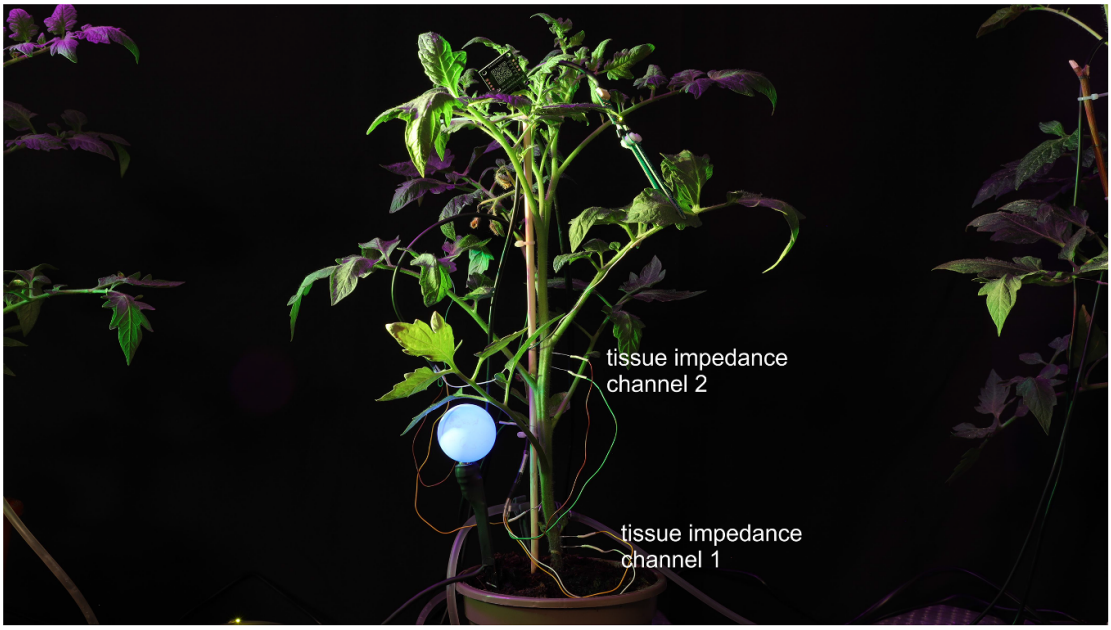

Figure 2. Setup with tissue impedance measurements (two independent channels) with tomato plants *Solanum lycopersicum*.

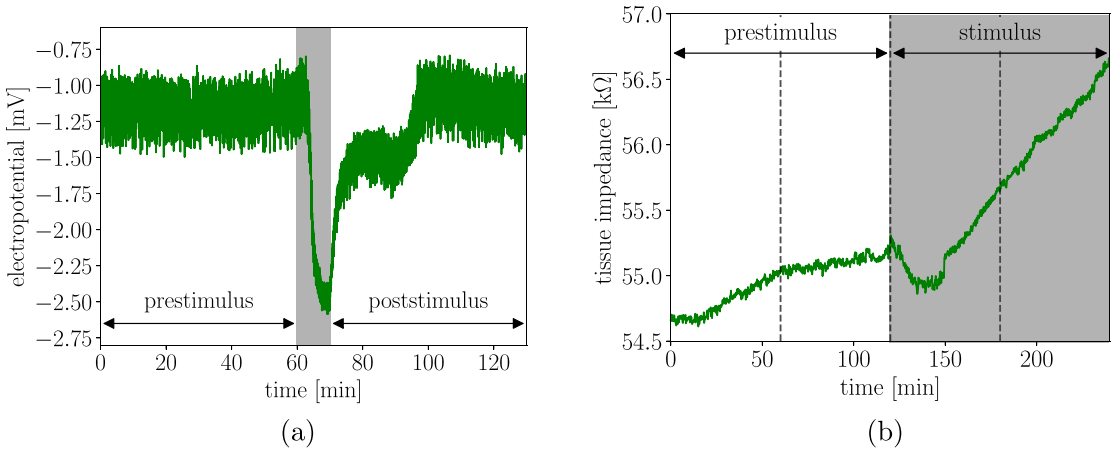

Figure 3. Example measurements of the electrical potential and tissue impedance where the respective stimulus application time is marked with a gray background. (a) Example measurement of the electrical potential change caused by a wind excitation using our experimental protocol. (b) Example measurement of the tissue impedance change caused by a light stimulus using our experimental protocol. The black dotted lines indicate interval boundaries which are used to differentiate between the two classes light and no stimulus, see section 3.3.

reflected in a variation of initial impedance in the prestimulus area. Therefore, all calculations should be performed in relation to this signal level.

3.2. Data collection

We use the experiment setups as described in section 3.1 to run our experiments for automatic data collection. We run two types of data collection experiments: electrical potential and impedance experiments. The collected data is labeled according to the applied stimulus and used to train and evaluate both the discriminant and deep learning classifiers. Given the labeled data, we can directly use the raw data or explore stimulus-specific features that lead to optimal class separation.

All electrical potential experiments follow the same experimental protocol and are divided into three phases. Following the nomenclature of Chatterjee *et al* [4], an experiment starts with a *prestimulus* phase of 60 min followed by a *stimulus* application phase of 10 min, and the experiment is concluded with a *poststimulus* phase of 60 min (see figure 3(a)). No stimulus is applied during either *prestimulus* and *poststimulus* phase to allow a stable initial plant state before the next stimulus application. The CYBRES phytosensor measures the plant's electrical potential responses at a frequency of 0.58 Hz.

We collected 544 measurements with wind stimulus, 504 measurements with heat stimulus, 134 measurements with blue light stimulus, and

138 measurements with red light stimulus. These result in a dataset of 1320 univariate time series of approximately 4500 samples each. We aim to train classifiers that are capable of identifying whether any of the previously mentioned stimuli is currently applied to the plant. Therefore, our classifiers should also be trained on electrical potential measurements where no stimulus is applied. For this reason, we extract a total of 544 twenty minute intervals from randomly selected prestimulus phases, which corresponds to the largest number of measurements from one class (i.e. wind stimulus). Now, our final dataset is composed of 1864 univariate time series.

Since tissue impedance has a slower reaction time than electrical potentials [34], we implement a different experiment protocol for impedance experiments with a longer stimulus phase. The protocol is divided into a 120 min prestimulus phase followed by a 120 min stimulus application phase (see figure 3(b)). We conducted 20 experiments where we applied a light stimulus. The tissue impedance is sampled with a frequency of approximately 0.08 Hz. From these experiments, we obtain 40 univariate time series of approximately 1250 samples, because of the two impedance channels of the CYBRES phytosensor. We use the collected impedance data to train classifiers that are capable of distinguishing between only two classes: *no stimulus* and *light*. Therefore, from each experiment, we extract the 120 min prestimulus intervals and label them as no stimulus. This results in a dataset of 80 univariate time series.

3.3. Discriminant analysis classification

To perform the discriminant analysis, statistical features are first calculated from the time series and then used for classification. In general, we are interested in classifying the plant responses during active stimulus application phases. That's why, for electrical potential-based classification, we extract the features for the first 340 samples (≈9.8 min) of the stimulus phase and use the last 340 samples of the prestimulus phase for background subtraction (see figure 3(a)). We follow the same procedure for each stimulus class (wind, heat, blue light, and red light). In the case of no stimulus class, we split the 20 min time series into two halves. Similar to the other classes, we now extract the last 340 samples from the first half for background subtraction and the first 340 samples from the second half for classification.

In preprocessing the impedance dataset, we are again only interested in the change caused by a stimulus. For this reason, we extract the features for the first 295 samples (≈60 min) of the stimulus phase and use the last 295 samples of the prestimulus phase for the background subtraction for the class light, see dashed lines in figure 3(b). The time series of the no stimulus dataset is again split in half and processed as for the no stimulus class in the electrical potential measurements. We use the same division of data into test and training in both classification approaches. Here, 70% of all data is used for training, while the remaining 30% is used for testing. Since the datasets per stimulus are of different sizes, the respective proportions in the testing and training sets are kept constant.

Discriminant analysis begins by computing nine statistical features from each univariate time series. We use a subset of the statistical features defined by Chatterjee *et al* [4] including the mean (μ), variance (σ^2), skewness (γ), kurtosis (β), interquartile range (IQR), Hjorth mobility (HM), Hjorth complexity (HC), wavelet packet entropy (WPE), and average spectral power (ASP). The first four moments of distributions μ, σ^2, γ, β describe basic properties of a distribution [23]. The first moment μ is the expected value or mean. The second moment σ^2 gives the dispersion of the data around the mean. The third moment γ indicates the asymmetry around the mean. β is the fourth moment and indicates whether the data have a heavy or light tail compared to a normal distribution. These moments of distributions are calculated as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i,\ \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2,$$
$$\gamma = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma}\right)^3,\ \beta = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma}\right)^4, \quad (1)$$

where x_i represents the samples and n the total number of samples. IQR is another descriptive characteristic that describes the dispersion for the middle 50% of the data that lie between the first quartile Q_1 and the third quartile Q_3 of the observed data [35]. It is particularly useful for skewed distributions because it is unaffected by outliers. The IQR is defined as

$$\mathrm{IQR} = Q_3 - Q_1. \quad (2)$$

Hjorth parameters [36] are commonly used to analyze signals in the time domain and for feature extraction in electroencephalography signals. The parameters are based on the variance ${\sigma_1}^2$ of the signal and its first ${\sigma_2}^2$ and second ${\sigma_3}^2$ derivatives. For discrete series [37], the variances can be calculated as

$$\sigma_1^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2, \quad \sigma_2^2 = \frac{1}{n-1}\sum_{i=2}^{n}\left(x_i - x_{i-1}\right)^2,$$
$$\sigma_3^2 = \frac{1}{n-2}\sum_{i=2}^{n-1}\left(x_{i+1} - 2x_i + x_{i-1}\right)^2. \quad (3)$$

Based on the these variances, HM and HC can be calculated as follows

$$\mathrm{HM} = \frac{\sigma_2}{\sigma_1}, \qquad \mathrm{HC} = \sqrt{\frac{\sigma_3^2}{\sigma_2^2} - \frac{\sigma_2^2}{\sigma_1^2}}. \quad (4)$$

Wavelets are used to decompose signals arising from multiscale processes such as climatology or

neuroscience [38]. These wavelets are decaying wave like oscillations with zero mean that enable us to analyze complex signals in the frequency and time domain in different resolution. For instance, we get a high frequency and low time resolution in lower frequency ranges since the time signal barley changes. While increasing the frequency we increase the time resolution and decrease the frequency resolution since changes in the time domain occur more often. Through the discrete wavelet transformation, we obtain wavelet coefficients $\mathcal{W}_\psi(f)(j,k)$ that are defined by the convolution of the signal f with a wavelet ψ:

$$\mathcal{W}_\psi(f)(j,k) = \langle f, \psi_{j,k} \rangle \text{ where } \psi_{j,k}(t) = \frac{1}{a^j}\psi\left(\frac{t-kb}{a^j}\right). \tag{5}$$

ψ is also called the mother wavelet while $\psi_{j,k}$ represents the wavelet family. It is a set of scaled and translated versions of the mother wavelet ψ. We use the Daubechies 1 mother wavelet in our approach. a denotes a scaling while b is used to translate the wavelet across the signal. j and k denote the scaling and translating index while moving a scaled wavelet through the signal. Based on the gained wavelet coefficients and the Shannon entropy we are able to calculate the wavelet entropy. A low WPE indicate more organized data while a high WPE indicate disordered data and is defined as

$$\text{WPE} = -\sum_{\mathcal{W}\in\mathcal{W}_\psi} \mathcal{W}^2 \log\left(\mathcal{W}^2\right). \tag{6}$$

The ASP is the integral over the power spectral density (PSD), also called the power spectrum. The PSD is a measure of the energy variation in a signal distributed over the measured frequencies. We calculate the PSD using the estimate provided by Welch [39]. The signal is first divided into k overlapping segments, then a Hann window is applied to each segment. We use a segment size of 50 samples in our approach, with 25 samples overlapping between segments. After windowing, the periodogram of each segment is calculated using the discrete Fourier transform. Then, all k periodograms are averaged. The resulting function $\text{PSD}(f)$ is the PSD as a function of frequency f. Finally, the discrete ASP is defined as the sum of all PSDs:

$$\text{ASP} = \sum_{f\in F} \text{PSD}(f), \tag{7}$$

where F denotes all frequencies contained in the signal.

After feature calculation, we apply background subtraction by subtracting each prestimulus time series feature from the stimulus time series features to analyze only the change in signal. The resulting incremental features $\mathbf{x'}_i \in \mathbb{R}^n$ of the test and training set are min-max normalized based on the training data where $i \in [\mu, \gamma, \beta, \text{IQR}, \text{HM}, \text{HC}, \text{ASP}, \text{WPE}]$ and $j \in [1, \ldots, n]$:

$$x'_{i,j} = \frac{x_{i,j} - \min(\mathbf{x}_{\mathbf{i},\text{train}})}{\max(\mathbf{x}_{\mathbf{i},\text{train}}) - \min(\mathbf{x}_{\mathbf{i},\text{train}})}. \tag{8}$$

The normalized training set is now used to train five discriminant analysis classifiers. These include linear discriminant analysis (LDA), QDA, naive Bayes LDA and QDA, and Mahalanobis. We use the implemented classifier of the Matlab framework *classify*[4]. In this classification approach, we assume that each class can be approximated by a Gaussian distribution [23]. We start with estimating the mean $\mu = E[\mathbf{x}]$ and the covariance matrix $\Sigma = E\left[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^{\text{T}}\right]$ to build the probability density function $p(\mathbf{x}|\theta)$ of each class based on the known observations $\mathbf{x}$ in a l-dimensional feature space:

$$p(\mathbf{x}|\theta) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right) \quad \text{with } \theta = \{\mu, \Sigma\}. \tag{9}$$

Based on the probability density function $p(\mathbf{x}|\theta)$ and the prior probability $P(\theta)$ we assign new observations $\mathbf{x}$ to the class $\hat{\theta}$ with the highest a posterior $P(\theta|\mathbf{x})$ probability

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\, P(\theta|\mathbf{x}) = \underset{\theta}{\text{argmax}}\, p(\mathbf{x}|\theta)P(\theta). \tag{10}$$

All mentioned classifiers follow a similar methodology, but differ in the estimated covariance matrix. In LDA, we assume that all classes have the same covariance matrix Σ. However, since the covariances of the classes vary, the pooled covariance matrix of all classes is determined. In contrast, the quadratic and Mahalanobis classifiers use stratified covariance estimation. However, the prior probabilities are not considered by the Mahalanobis classifier. In both naive Bayes classifiers, we assume independent features whereby only the diagonal entries of the covariance matrix are estimated.

We train all five classifiers in three different settings with an increasing number of classes. The first setting involves a binary classification between no stimulus and wind. We increase the complexity to a three-class classification by adding the heat class. The final setting includes all stimuli, consisting of wind, heat, red light, blue light, and no stimulus. If we use all nine available features, the difficulty of the classification problem could increase significantly. The classification might be inefficient and computationally demanding. In the worst case, the classification accuracy may decrease as the dimensionality of the feature space increases with each additional feature. To counter this problem, sequential forward selection (SFS) is applied before each training session. SFS is

[4] https://de.mathworks.com/help/stats/classify.html.

a greedy algorithm that starts with an empty feature set. It sequentially expands the feature set with individual features based on the training set and accuracy improvement. SFS terminates when no improvement in accuracy can be achieved by adding more features. We use the Matlab framework *sequentialfs*[5] for SFS. All collected datasets and trained classifiers, as well as the code for the experiment and data collection setup, are available online [40].

3.4. Deep learning-based classification

Given the success of deep learning in recent years in many fields, such as image processing [41], natural language processing [42], drug discovery [43], and also time series prediction [15], we want to compare deep learning approaches with current state-of-the-art discriminant analysis in the application of stimulus classification. Therefore, we are using a deep learning framework for TSC developed by Fawaz *et al* [15]. This framework includes ten different deep end-to-end classifiers, which take an electrophysiological plant signal time series as an input and output a likelihood for each possible class, i.e. applied external stimulus. All approaches are implemented using the machine learning library Tensorflow 2.0[6] (with Python 3). We trained the networks on a Nvidia DGX A100 with eight cores (two in use). Similar to the discriminant analysis approach, we are interested in classifying the plant's reaction to the applied stimulus. Therefore, we extract the first 340 samples (≈9.8 min) of the stimulus phase. Next, we normalize them to remove any shifts or data scaling due to the inherent bias of various plants [44] and electrode placement. We apply the Z-score normalization: $\boldsymbol{x}_{\text{norm}} = (\boldsymbol{x} - \mu_x)/\sigma_x$, with μ_x the mean and σ_x the standard deviation of the extracted series $\boldsymbol{x}$.

The deep learning framework contains three different fundamental structures: ANNs, CNNs, and echo state networks (ESNs). In the following, we explain the general idea of each fundamental structure and the remarkable differences in each implementation.

The ANN structure is the simplest and most traditional architecture for deep learning. The architecture consists of artificial neurons, which are stacked up to several layers connected in series. Each neuron has a nonlinear activation function and receives the output of all neurons from the previous layer (fully connected) as input. The first layer receives one time series as one input vector[7]. The final layer contains a softmax activation function for providing a likelihood distribution over the available classes. The cross-entropy loss function is used in all approaches, if not stated otherwise, because it is a good measure of how distributions differ. The foremost disadvantage of this architecture in TSC tasks is that the architecture does not incorporate spatial invariance [15], meaning that each time step of the time series has its own weights and is considered individually, leading to loss of temporal information, i.e. initially the network does not know that time step $t + 1$ comes after time step t. The sole implementation of this approach is the multilayer perceptron (MLP) [45]. Each layer is preceded by a dropout operation which helps to prevent overfitting [46]. In this implementation, the AdaDelta optimizer is used.

CNNs are common used architectures because of their success in many other domains, such as image [47] or natural language processing [42]. In the context of TSC tasks, convolutional layers (main ingredient of CNNs) can be considered as filters, which learn invariant discriminative features of time series [15]. In comparison to the aforementioned ANNs, the weight sharing property of convolutional layers allows for invariance because multiple time steps share the same weights, i.e. the network *knows* the coherence of the time steps initially. The general CNN structure has two stages: first learning a feature representation using convolutional layers, which can be augmented to convolutional blocks by adding pooling operations for further invariance. In the second stage, the output of the convolutional layers or blocks are forwarded to one or more fully connected layers for classifying the learned features. Finally, similar to the ANN approach, the last layer usually contains a softmax activation function that provides a likelihood distribution over the available classes.

The first implementation of the framework using this architecture is the time convolutional neural network (TCNN) [48] with two convolutional blocks. It is the only implementation, which does not use a softmax activation function but a sigmoid activation function and thus a mean squared error loss function (with Adam as an optimizer). The second implementation is a fully convolutional neuronal network (FCN) [45], which has three convolutional blocks where a batch normalization operation instead of pooling operation is applied [49]. Furthermore, a global average pooling layer instead of fully connected layers is used. The subsequent implementation is the Encoder [50] with three convolutional blocks, where an instance normalization operation [51] and a dropout operation are applied in each block. The major difference compared to the FCN is that an attention layer replaces the global average pooling layer [52]. The implementation of time Le-Net (t-LeNet) [53] follows the structure of a classical CNN. In this approach, additional preprocessing of the time series is done, that is, window warping (WW) and window slicing (WS), for data augmentation. As WW and WS are also applied

[5] https://de.mathworks.com/help/stats/sequentialfs.html.

[6] www.tensorflow.org; A list of further required Python 3 packages can be found at [40].

[7] A detailed structure of the architecture, e.g. dimensions of layers, of the various implementations can be found at [40].

during inference, the classification result is a majority vote over all window slices generated from one time series. The multi-scale convolutional neural network (MCNN) [54] also has a traditional CNN structure. The main difference lies in the heavy data preprocessing. First WS is used for data augmentation. Second, the slices are transformed using identity mapping, down sampling, and smoothing before being passed to the network. Similar to the t-LeNet for classification, a majority voting over all window slices of the time series is applied. The multi-channel deep convolutional neural network (MCDCNN) [55] is originally implemented for multivariate time series, as it splits them into multiple univariate time series. The network collapses to a traditional CNN structure in our application because we only consider univariate time series. It is the only approach using a stochastic gradient descent as an optimizer paired with the categorical cross-entropy loss function. The implemented Residual Network (ResNet) [45] has the deepest architecture with nine convolutional blocks. To prevent the vanishing gradient problem skip connections are added every three convolutional blocks. The vanishing gradient problem is caused by the deep architecture, as during backpropagation the gradient for updating the weights becomes close to zero and therefore the weights do not change during training. The last implementation using the CNN structure is the Inception Neuronal Network [56]. It consists of six inception modules, which are characterized by multiple convolutions in parallel—following the idea of making the network wider, not deeper—instead of the classical convolutional blocks. In addition, similar to the ResNet, skip connections are added every three blocks.

Finally, we want to present the ESN which is a special form of recurrent neuronal networks designed for time series prediction [15]. As recurrent neuronal networks are hard to train due to the vanishing gradient problem and no convergence because of cyclic dependencies, in ESNs, only the output weights are changed during learning. The hidden layers, also called reservoir, and input weights are randomly initialized and kept constant. The framework implements the time warping invariant echo state network (TWIESN) [57].

We train all previously mentioned deep classifiers for the same three classification settings, as described in section 3.3, and these are: no stimulus and wind; no stimulus, wind and heat; and no stimulus, wind, heat, red light, and blue light. For each classification setting we do five repetitions (iterations) to minimize the influence of the random initialization of the weights. Similar to the discriminant analysis we split the data into 70% training and 30% testing sets. Given a time series of plant electrical potential measurements as input, the networks are trained to find the likelihood within the stimulus classes mentioned above. We use three metrics to evaluate the performance of the ten deep classifiers: accuracy, precision, and recall. The accuracy is the ratio of correct classified samples and is defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{11}$$

with TP true positives, TN true negatives, FP false positives, and FN false negatives. With similar nomenclature we can define precision, which describes the proportion of actually correct classified positive samples, as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{12}$$

and recall, which describes the proportion of actual correctly identified positive samples,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{13}$$

The precision and recall is calculated per class and then averaged over all classes.

4. Results

In this section, we report results from both the discriminant analysis classifiers (see section 4.1) and the deep learning time series classifiers (see section 4.2), followed by a discussion (see section 4.3).

4.1. Discriminant analysis

The nine computed statistical features (as described in section 3.3) of each class are presented in violin plots[8] in figure 4. Each violin is divided into two halves. The left side shows the distribution of the normalized training set, while the right side reflects the normalized test set. Especially in classification tasks, we can use these plots to get a first impression of a possible separability of the data. On the one hand, high classification accuracy can be expected if the distributions of the classes do not overlap on the y-axis. On the other hand, high accuracy can be achieved if the training and test data of one class do not differ significantly, that is, both subsets represent the same data distribution.

This is confirmed for the binary classification of no stimulus (first violins in figure 4) and wind (second violins in figure 4). Any one of the features, mean, variance, or ASP, is required to achieve 100% accuracy with each of the five classifiers (see table 1). Regardless of the classifier, we also achieve accuracies above 99% with the other features: kutosis, IQR, and WPE. The lowest accuracy of 89.5% is achieved with feature HC in combination with LDA, naive Bayes LDA, and Mahalanobis.

[8] Violin plots combine boxplots with smoothed histograms and visualize more features of the data's distribution [58].

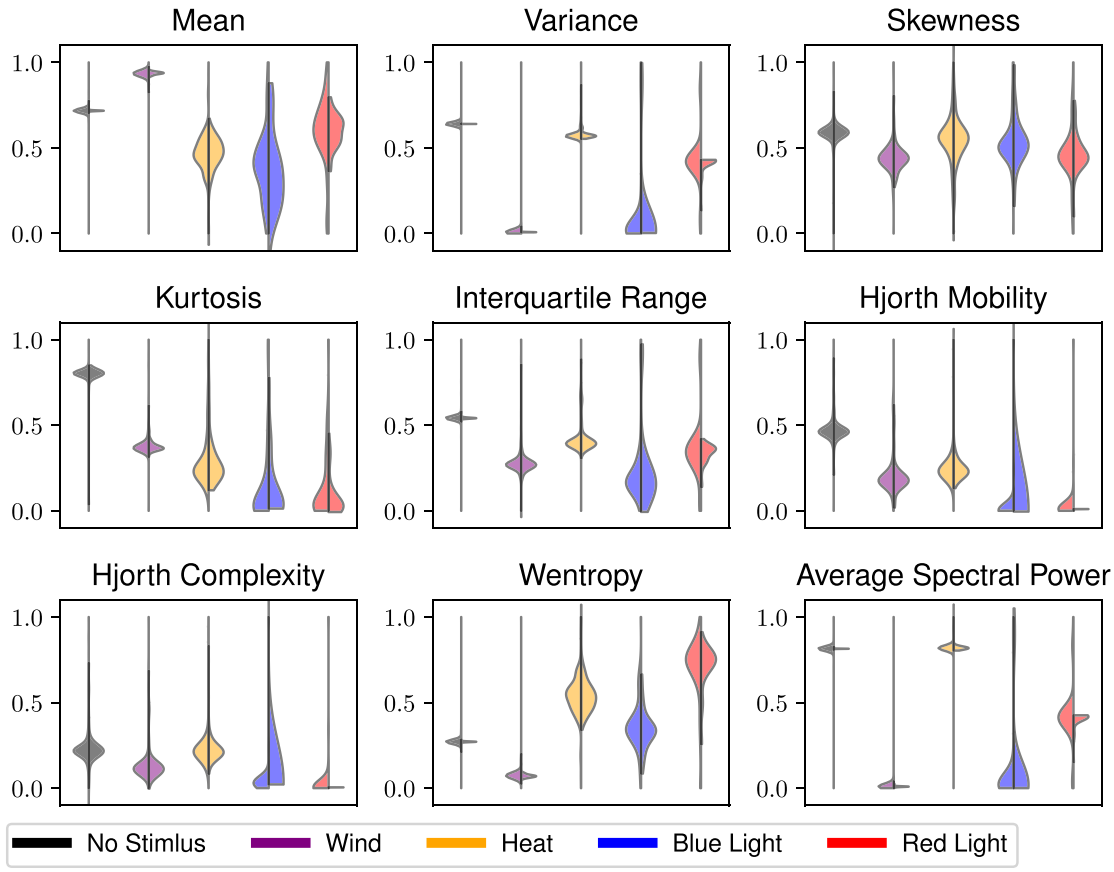

Figure 4. Electrical potential dataset, violin plots of the distribution of all nine normalized features: mean μ, variance σ^2, skewness γ, kurtosis β, interquartile range (IQR), Hjorth mobility (HM), Hjorth complexity (HC), wavelet packet entropy (WPE, Wentropy), and average spectral power (ASP). Each violin represents one of the classes, from left to right: no stimulus, wind, heat, blue light or red light. The left half of each violin shows the distribution of the training set, the right side that of the test set.

Table 1. Electrical potential dataset, accuracies for all five discriminant analysis classifiers in percent, first with two (wind and no stimulus), then three (+ heat) and finally all five (+ red and blue light) classes. In addition, the subset of features selected by SFS. The order of features shown corresponds to that of SFS as it added features sequentially.

Classifier	Setting	Accuracy (%)	Feature set
LDA	two classes	100.0	[ASP]
	three classes	100.0	[WPE, μ, ASP]
	five classes	98.6	[σ^2, WPE, ASP, IQR, HM, μ]
Naive Bayes LDA	two classes	100.0	[ASP]
	three classes	99.6	[WPE, σ^2, HM]
	five classes	97.8	[σ^2, μ, HM]
QDA	two classes	100.0	[ASP]
	three classes	100.0	[WPE, μ]
	five classes	99.1	[σ^2, HM, ASP, IQR, β, HC, WPE, μ]
Naive Bayes QDA	two classes	100.0	[ASP]
	three classes	99.8	[WPE, σ^2]
	five classes	98.4	[σ^2, μ]
Mahalanobis	two classes	100.0	[ASP]
	three classes	100.0	[μ, WPE]
	five classes	98.9	[σ^2, μ, WPE, ASP]

As more classes are added, the classification task becomes more difficult. This leads to a slight decrease in the highest accuracy to 99.8% when only one feature is used in the case of three classes. This is achieved when the mean is used in combination with the Mahalanobis classifier. However, accuracies above 99% are achieved with QDA, naive Bayes QDA, and Mahalanobis using the WPE and the mean (table 1). Already in figure 4, a good separation of the three classes with WPE or mean can be seen,

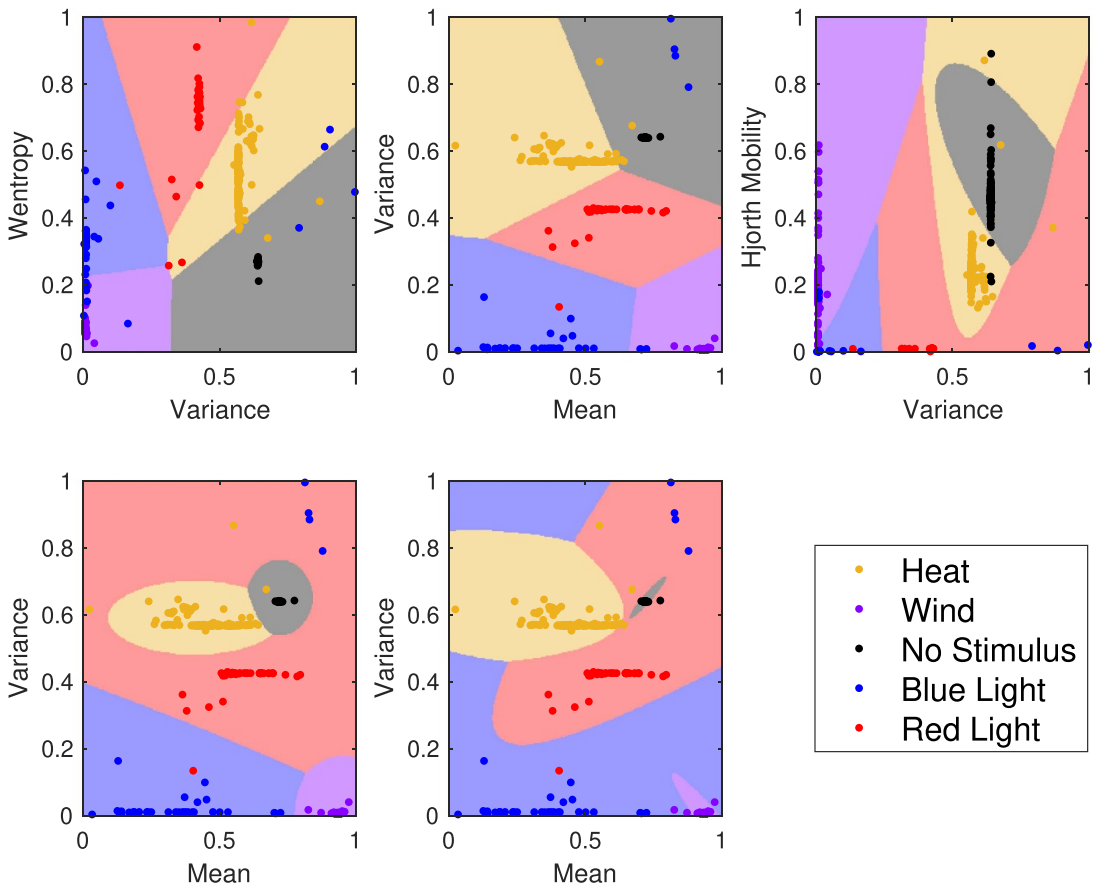

Figure 5. Electrical potential dataset, class ranges and test set samples of the five-class classifiers using the first two features selected with SFS, upper row from left to right: LDA, naive Bayes LDA, QDA; lower row from left to right: naive Bayes QDA and Mahalanobis.

since most of the data does not overlap on the vertical axis. Other promising features are variance, kurtosis, and IQR, as they achieve over 90% accuracy with all classifiers. The feature HC performs the worst, achieving an accuracy of ≈60% with any classifier.

In a next step, we investigate whether higher accuracies can be achieved if more than one feature is used. We select features using SFS as described in section 3.3. The accuracy of QDA and Mahalanobis increased to 100% with a feature combination of mean and WPE. Adding ASP as a third feature, LDA also achieves 100% accuracy. Naive Bayes LDA achieves 99.6% accuracy with variance, HM, and WPE, while naive Bayes QDA achieves 99.8% with variance and WPE.

The highest accuracy of all classifiers using one feature (variance) is decreased to ≈90% in our five-class case. The lowest accuracy of ≈50% with all classifiers is achieved with the feature skewness. Again, we use SFS to increase the accuracy of the classifiers. In order to get an intuitive understanding of the task difficulty and the quality of the classifiers, we show the classification bounds and test data for all five classes and each classifier after adding the second feature in figure 5. According to SFS, the highest accuracy of 99.1% is achieved with QDA and all features except skewness. The second highest accuracy of 98.9% is achieved with the Mahalanobis classifier and the features mean, variance, skewness, WPE, and ASP. After adding IQR and HM to the last set of features, we obtain the highest accuracy of 98.6% with LDA. The naive Bayes classifiers require the fewest features but have the lowest accuracy. Naive QDA requires only the variance and mean to achieve its highest accuracy of 98.4%. After adding HM, we obtain the required feature subset of the naive Bayes LDA classifier to achieve an accuracy of 97.8%

We conclude this study on discriminant analysis with the results for the impedance dataset. The nine computed statistical features of the two classes, light and no stimulus, based on the impedance dataset, are shown in figure 6. In this binary classification, we achieve a maximum accuracy of 96% with each classifier when ASP or kurtosis is taken as a single feature (see table 2). The accuracies of QDA and Mahalanobis can be increased to 100% by SFS when the features WPE and mean are used for classification. However, SFS decreased the accuracy of the naive Bayes QDA when WPE was chosen instead of ASP or Kurtosis.

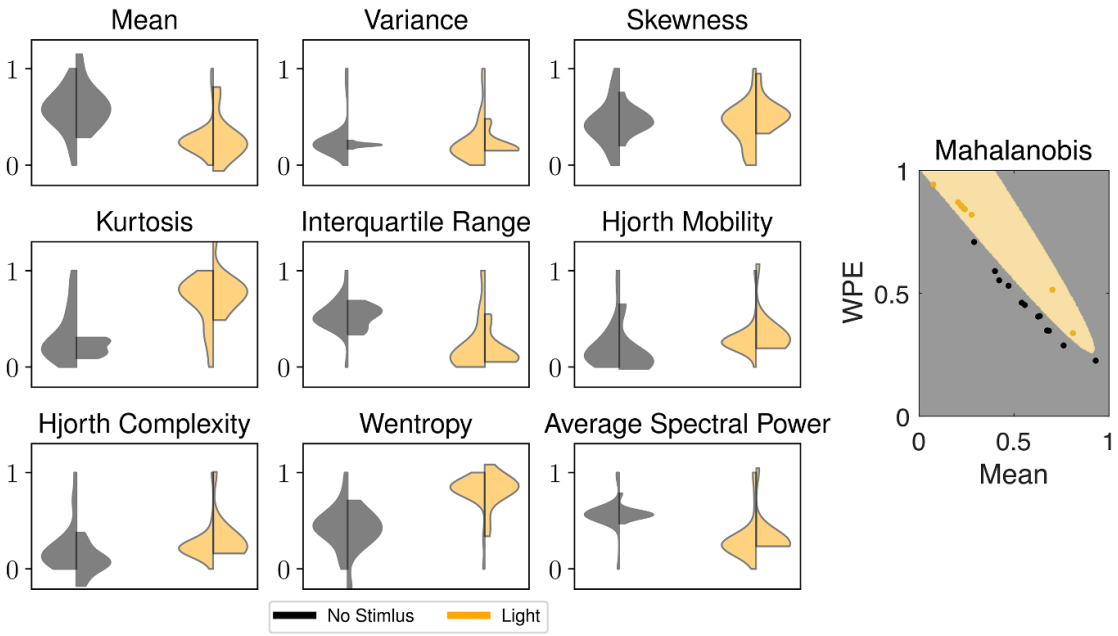

Figure 6. Impedance dataset, violin plots of the distribution of all nine normalized features. Each violin represents one of two classes: no stimulus or light. The left half of each violin shows the distribution of the training set, while the right side shows the distribution of the test set. On the far right, the class ranges and test set samples of the Mahalanobis classifier using the features mean and WPE.

Table 2. Impedance dataset, accuracies for all five discriminant analysis classifiers in percent and the subset of features selected by SFS. The order of features given is that of SFS as it adds features sequentially.

Classifier	Accuracy (%)	Feature set
LDA	96.0	[ASP]
Naive Bayes LDA	96.0	[ASP]
QDA	100.0	[WPE, μ]
Naive Bayes QDA	88.0	[WPE]
Mahalanobis	100.0	[WPE, μ]

This is because SFS selects the subset of features based on the training data.

4.2. Deep learning-based classification

We trained ten deep end-to-end classifiers, as described in section 3.4, on the electrical potential measurement training dataset (70% of all data). We did five iterations per classifier to minimize the influence of weight initializations. The averaged results, obtained using the test dataset (the remaining 30% of all data), for accuracy, precision, and recall for nine out of ten classifiers, can be seen in table 3. The MCNN could not learn an appropriate representation of the data, so we do not consider it in further analysis.

For the two-class classification task, the Inception model has the highest accuracy with 89.7%, followed by the ResNet (89.4%) and the FCN (89.3%). The TWIESN performs worst (57.8%) in differentiating wind and no stimulus. The average accuracy over all classifiers is 80.8%. The precision is within 1% deviation from the accuracy for all classifiers except for MCDCNN ($\Delta = 1.2\%$) for this setting. As described in section 3.4 the recall is calculated using the true positives and false negatives per class. Only looking at one class at a time implies that there are only positive samples, and thus there are no true and false negatives. Hence, in a perfectly balanced classification problem, i.e. in our case, the recall is similar to the accuracy.

By adding the heat stimulus, we have a three-class classification task, where the Inception model again has the highest accuracy with 92.2%, followed by the ResNet (92.0%) and the FCN (91.4%). The TWIESN performs worst with 53.2%. Besides the three best models, the Encoder and the MCDCNN achieve higher accuracies compared to the two-class setting, and the average accuracy of all approaches drops slightly to 80.3%. The precision in the three-class classification task of all classifiers is within 1% deviation from the accuracy except for MCDCNN ($\Delta = 1.4\%$) and TWIESN ($\Delta = 2.6\%$). The third class, heat stimulus, is slightly smaller than the other two classes, but the average recall for all classifiers is within 1% deviation from the accuracy.

Finally, we present the five-class classification task by adding the blue light and red light stimulus. In this setting, the ResNet and the FCN have the highest accuracy with 83.5%, followed by the Inception model (83.0%). Similar to the previous settings, the TWIESN has the lowest accuracy (43.2%), leading to an average accuracy of 70.1% for all approaches. Due to the unbalanced classes, the precision deviates significantly from the accuracy. The largest deviation can be observed in the TWIESN with $\Delta = 14.3\%$

Table 3. Electrical potential dataset, average accuracies, precisions and recalls for all ten deep classifiers in percent. We averaged over five iterations to minimize the influence of the random initialization of the weights. We present three different settings: two classes (no stimulus and wind stimulus); three classes (no stimulus, wind, and heat stimulus); and five classes (no stimulus, wind, heat, red light, and blue light stimulus).

	Two classes			Three classes			Five classes		
Classifier	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)
MLP	73.2	73.3	73.2	68.6	68.7	68.6	57.4	50.8	47.8
TCNN	80.5	80.6	80.5	75.7	76.1	75.9	62.5	53.0	50.3
FCN	89.3	89.3	89.3	91.4	91.6	91.6	83.5	75.6	71.3
Encoder	86.6	86.6	86.6	89.6	89.8	89.8	80.0	74.5	67.0
t-LeNet	82.5	83.1	82.5	80.8	81.5	81.0	70.3	62.5	53.5
MCDCNN	78.4	79.6	78.4	78.7	80.1	79.0	67.5	59.0	55.6
ResNet	89.4	89.5	89.4	92.0	92.2	92.2	83.5	74.9	70.2
Inception	89.7	89.8	89.7	92.2	92.3	92.3	83.0	73.0	68.6
TWIESN	57.8	57.9	57.8	53.2	55.8	53.2	43.2	29.0	30.3

Table 4. Electrical potential dataset, the average confusion matrix over all classifiers for three different settings: wind and no stimulus (left); wind, heat, and no stimulus (middle); and wind, heat, red light, blue light and no stimulus (right). The different stimuli are: no stimulus (no), wind stimulus (wind), heat stimulus (heat), blue light stimulus (blue) and red light stimulus (red). The rows are the true values, which sum up to 163 for no stimulus, 163 for wind stimulus, 151 for heat stimulus, 40 for blue light stimulus, and 41 for red light stimulus. The columns are the assigned values by the classifiers.

Stimuli	No	Wind	No	Wind	Heat	No	Wind	Heat	Blue	Red
No	126.1	36.9	126.5	32.8	3.7	121.5	30.4	3.6	3.6	3.8
Wind	29.2	133.8	25.5	125.0	12.5	28.7	118.5	12.9	1.5	1.4
Heat			8.9	10.8	131.3	10.4	10.2	129.9	0.4	0.1
Blue						8.3	13.1	1.6	10.6	6.3
Red						8.7	14.1	2.0	5.5	10.7

followed by the Inception model ($\Delta = 10.0\%$) and TCNN ($\Delta = 9.5\%$). The unbalanced dataset has a similar effect on the recall, where now the deviation from accuracy is, on average $\Delta = 12.9\%$, with t-LeNet having the highest deviation ($\Delta = 16.8\%$) and MLP having the lowest deviation ($\Delta = 9.7\%$).

Further, we provide a confusion matrix for each setting, see table 4, which summarizes the classification results from all classifiers. The individual confusion matrices for each classifier can be found in additional material [40]. These tables reveal among which stimuli the classifiers have difficulties discriminating. The general trend in the three-class classification setting is that all classifiers have problems correctly discriminating between wind and no stimulus, while differentiating between heat and wind seems less difficult. The classifiers are best in distinguishing between heat and no stimulus. We see similar trends for the five-class classification task compared to the three-class setting for the stimuli of wind, heat, and no stimulus. Furthermore, we notice that the blue and red light stimuli are often misclassified as wind.

4.3. Discussion

With our experiments, we have found that both the 'ZZ' plant and tomato exhibits electrical responses to our applied stimuli of wind, heat, blue light, and red light. Both the changes in electrical potential and impedance exhibit stimulus-specific characteristics that allow us to train classifiers that identify the respective stimulus with high accuracy. Although both methods, discriminant analysis and deep learning, achieve high accuracies, the discriminant analysis significantly outperforms the deep learning approach. One of the main differences between the two approaches is the preprocessing of the data. In discriminant analysis, statistical properties are first calculated from the raw signals and then classified assuming a Gaussian distribution. In contrast, the raw data is only Z-score normalized before being presented to the neural networks, increasing the task difficulty, probably considerably. Depending on their architecture, the networks directly classify the time series (ANN), autonomously learn a representation of the time series before classification (CNN), or randomly map the time series to another dimension before classification (ESN). The property of autonomously learning a representation of the data is advantageous because less *a priori* knowledge and engineering for finding appropriate features is required. Therefore, the deep learning method can readily be adapted to other classification tasks without the overhead of reconsidering which features need to be selected.

For the binary classification task, the discriminant analysis achieves an average accuracy of 100% while the deep learning approach only achieves an average accuracy of 80.8%. On average, more samples of the no stimulus class (36.9) are incorrectly assigned than samples of the wind class (29.2).

Next, we discuss the three-class task. By adding the heat stimulus, we add 504 time series to the dataset. The dataset sizes are now unequal but can still be considered balanced. To achieve 100% accuracy, we need at least two features. The deep learning approach achieves a lower average accuracy of 80.3%. Nevertheless, the accuracy increases with the increased number of classes for the FCN, Encoder, MCDCNN, ResNet, and Inception model. This counterintuitive effect can be explained by the training data getting bigger while the number of trainable parameters in the classifiers increases less. The confusion matrices indicate that the classifiers struggle more to discriminate wind from no stimulus than wind from heat. This may be caused by the shapes of the different electrical potential signals.

A total of 272 samples is added to our dataset by including the two remaining stimuli: red light (134 samples) and blue light (138 samples). Now we have an unbalanced dataset. The discriminant analysis is largely unaffected by unbalanced datasets (red and blue light correctly classified). On average, only five samples of the blue light stimulus class and one sample of the red light stimulus class are misclassified. We again observe a decrease in average accuracy to 98.56%. The average accuracy of the deep learning approach decreases to 70.1%. It is possibly affected by the unbalanced dataset as shown by the decrease in precision (average 61.4%) and recall (average 57.2%). This is also confirmed by the confusion matrices for all classifiers. As mentioned in section 4.2, the confusion matrix shows that most red and blue light stimuli are classified as wind stimulus. This causes the decrease in precision and recall (about 20% compared to the three-class task).

Another issue with the deep learning approach to TSC is the lack of sufficient training data. In our experiments, we obtained approximately 1300 time series as a training dataset which is probably not sufficient for networks with larger numbers of trainable weights. In previous work [14], the same dataset was used. However, the time window used for classification was slightly larger (about 11.5 min compared to our 9.8 min), including about 1.5 min from the poststimulus phase. The larger time window allowed for more information, including a part of the steep negative slope at the beginning of the poststimulus phase, see figure 3(a), resulting in better performance (on average, 11.3% for two classes, 7.5% for three classes, and 6.5% for five classes). The results suggest that with more information, i.e. more samples or longer time windows, the performance of the classifiers can be increased. However, we shortened the time window length because we wanted to train classifiers that only rely on electrical potential changes during the stimulus application period. An advantage of deep-learning approaches is that they do not require additional information for background subtraction. We mitigated the possibility of overfitting for all classifiers by using unseen data for obtaining the results.

In addition to the electrical potential experiments, we also attempted training classifiers for tissue impedance, to test whether it can also be a valid method for stimuli classification. We chose to train discriminant analysis classifiers because they showed better classification accuracies than the deep learning classifiers during our electrical potential experiments. The average accuracy of all classifiers in combination with SFS is 96%. QDA and Mahalanobis achieve an accuracy of 100% with the features mean and WPE. This suggests that the tissue impedance data also encode information about externally imposed stimuli and is a promising candidate for further investigation.

5. Conclusion and future work

Our objective was to compare methods of discriminant analysis and deep learning for the application of classifying electrical potential and impedance signals of living plants. We have studied the efficiency and accuracy of these methods. We gathered the required data in many plant experiments with *Zamioculcas zamiifolia* ('ZZ' plant) and *Solanum lycopersicum* (tomato), measuring the electrical potential and tissue impedance response for five and two different stimuli, respectively, in a controlled environment.

Using 1864 electrical potential time series, we trained five discriminant analysis classifiers and ten classifiers based on deep learning. We have achieved high accuracies for many of them, depending on the difficulty of the classification task. We trained the classifiers in three different settings with increasing number of classes. The discriminant analysis classifiers outperformed the classifiers based on deep learning for all tasks. The first task was a binary classification between wind and no stimulus that allowed for accuracies of up to 100%. The second task was a three-class case with wind, heat, and no stimulus. Also for that task, we achieved 100% accuracy. The third task includes all five stimuli, including red and blue light, where the maximum accuracy was slightly reduced to 99.1%. Based on the 40 impedance time series, we trained five discriminant analysis classifiers for a single binary classification task (light stimulus, no stimulus). With QDA and Mahalanobis we achieved accuracies of up to 100%. Thus, tissue impedance is considered a promising method for measuring plant physiology. However, this measurement method requires more energy than the electrical potential because an alternating current must be applied during the measurement. This challenges our goal of an energy-efficient, self-sufficient measurement unit, so this method is only applicable when sufficient energy is available. Given these results, and also comparing them to our previous work [14] where we achieved good results with deep learning

methods for a qualitatively different dataset, we conclude that the decision regarding which classification technique to use, needs to be done on a case-to-case basis. Depending on the amount of available data and the feature selection, the simple statisitical methods can outperform the deep learning techniques. The comparison is arguably unfair because, for the statistical approach, we used hand-crafted feature-based inputs, while the deep learning approach was required to operate on the raw data. With increasingly more data being available, the deep learning methods may increase their accuracy and become competitive with the statistical methods. One advantage of deep learning models is that they can learn and interpret features from raw data. However, such deep structures involve increased model complexity and high resource requirements. These models typically require GPU-based devices to train the networks in a reasonable amount of time. In contrast, discriminant analysis computes known statistical features and assumes class-wise Gaussian distributions. As a result, this approach is less complex, easier to interpret, does not require GPU support, and is consequently better suited for sensor nodes with limited computational resources. Considering the simplest case of a fully connected network, each neuron of layer i with $n(i)$ neurons is assigned the output of all weighted neurons of layer $i-1$ [59]. Thus, we obtain $n(i-1)$ parameters per neuron, which totals $n(i-1) \times n(i)$ parameters in layer i. For example, our trained two-class MLP-based model contains 672 502 parameters. In contrast, discriminant analysis estimates only the covariance matrix and the mean of each class distribution and determines the features of the measured data, which significantly reduces the number of parameters and thus the amount of memory and computation required. Data collection in plant experiments is comparatively expensive and, hence, data availability is challenging. Methods of data augmentation for plant data may be a promising option. In our future work, we want to focus on applications for phytosensing within our project WatchPlant. While here we have used simple stimuli that are readily available, we will study other options to test classification techniques and phytosensing for monitoring urban areas, and especially air pollution.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.7126105.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 101017899.

ORCID iDs

Eduard Buss https://orcid.org/0000-0001-6993-5873
Till Aust https://orcid.org/0000-0003-2863-1341
Mostafa Wahby https://orcid.org/0000-0002-1903-3625
Tim-Lucas Rabbel https://orcid.org/0000-0003-4815-6250
Serge Kernbach https://orcid.org/0000-0002-5621-8979
Heiko Hamann https://orcid.org/0000-0002-2458-8289

References

[1] Garlando U, Bar-On L, Ros P M, Sanginario A, Peradotto S, Shacham-Diamand Y, Avni A, Martina M and Demarchi D 2020 Towards optimal green plant irrigation: watering and body electrical impedance *2020 IEEE Int. Symp. on Circuits and Systems* (*ISCAS*) pp 1–5
[2] Fromm Jorg and Fei H 1998 Electrical signaling and gas exchange in maize plants of drying soil *Plant Sci.* **132** 203–13
[3] Volkov A G 2012 *Plant Electrophysiology: Methods and Cell Electrophysiology* (Berlin: Springer)
[4] Chatterjee S K, Das S, Maharatna K, Masi E, Santopolo L, Mancuso S and Vitaletti A 2015 Exploring strategies for classification of external stimuli using statistical features of the plant electrical response *J. R. Soc. Interface* **12** 20141225
[5] Choi W-G, Toyota M, Kim S-H, Hilleary R and Gilroy S 2014 Salt stress-induced Ca^{2+} waves are associated with rapid, long-distance root-to-shoot signaling in plants *Proc. Natl Acad. Sci.* **111** 6497–502
[6] Farmer E E 2001 Surface-to-air signals *Nature* **411** 854–6
[7] Oyarce P and Gurovich L 2010 Electrical signals in avocado trees *Plant Signal. Behav.* **5** 34–41
[8] Beilby M J and Casanova M 2013 *The Physiology of Characean Cells* (Berlin: Springer)
[9] Mousavi S A R, Nguyen C T, Farmer E E and Kellenberger S 2014 Measuring surface potential changes on leaves *Nat. Protocols* **9** 1997–2004
[10] Jócsák I, Végvári G and Vozáry E 2019 Electrical impedance measurement on plants: a review with some insights to other fields *Theor. Exp. Plant Physiol.* **31** 359–75
[11] Yan X, Wang Z, Huang L, Wang C, Hou R, Xu Z and Qiao X 2009 Research progress on electrical signals in higher plants *Prog. Nat. Sci.* **19** 531–41
[12] Anthony M *et al* 1999 *Neural Network Learning: Theoretical Foundations* vol 9 (Cambridge: Cambridge University Press)
[13] Cap Q H, Uga H, Kagiwada S and Iyatomi H 2022 Leafgan: an effective data augmentation method for practical plant disease diagnosis *IEEE Trans. Autom. Sci. Eng.* **19** 1258–67
[14] Buss E, Rabbel T-L, Horvat V, Krizmancic M, Bogdan S, Wahby M and Hamann H 2022 Phytonodes for environmental monitoring: stimulus classification based on natural plant signals in an interactive energy-efficient bio-hybrid system *Proc. 2022 ACM Conf. on Information Technology for Social Good* (*GoodIT 2021*) pp 258–64
[15] Fawaz H I, Forestier G, Weber J, Idoumghar L and Muller P-A 2019 Deep learning for time series classification: a review *Data Min. Knowl. Discovery* **33** 917–63
[16] García-Carmona L *et al* 2021 Biohybrid systems for environmental intelligence on living plants: watchplant project *Proc. Conf. on Information Technology for Social Good* (*GoodIT 2021*) pp 210–15
[17] 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 2021 Watchplant: networked bio-hybrid systems for pollution

monitoring of urban areas *ALIFE 2022: The 2022 Conf. on Artificial Life, Volume ALIFE 2021: The 2021 Conf. on Artificial Life* p 37
[18] Križmančić M, Rabbel T-L, Buss E, Wahby M, Hamann H and Bogdan S 2022 Distributed connectivity control in bio-hybrid wireless sensor networks *Proc. 2022 ACM Conf. on Information Technology for Social Good* (*GoodIT 2022*) pp 250–7
[19] Burdon-Sanderson J S 1873 I. Note on the electrical phenomena which accompany irritation of the leaf of dionaea muscipula *Proc. R. Soc. London* **21** 495–6
[20] Forterre Yel, Skotheim J M, Dumais J and Mahadevan L 2005 How the venus flytrap snaps *Nature* **433** 421–5
[21] Li J-H, Fan L-F, Zhao D-J, Zhou Q, Yao J-P, Wang Z-Y and Huang L 2021 Plant electrical signals: a multidisciplinary challenge *J. Plant Physiol.* **261** 153418
[22] Fromm J and Lautner S 2007 Electrical signals and their physiological significance in plants *Plant Cell Environ.* **30** 249–57
[23] Theodoridis S and Koutroumbas K 2009 *Pattern Recognition* 4th edn (New York: Academic)
[24] Chaovalit P, Gangopadhyay A, Karabatis G and Chen Z 2011 Discrete wavelet transform-based time series analysis and mining *ACM Comput. Surv.* **43** 1–37
[25] Najdenovska E, Dutoit F, Tran D, Plummer C, Wallbridge N, Camps C and Raileanu L E 2021 Classification of plant electrophysiology signals for detection of spider mites infestation in tomatoes *Appl. Sci.* **11** 1414
[26] Reissig G N, Oliveira T F C, Costa A V L, Parise A G, Pereira D R and Souza G M 2021 Machine learning for automatic classification of tomato ripening stages using electrophysiological recordings *Front. Sustain. Food Syst.* **5** 696829
[27] Qin X-H, Wang Z-Y, Yao J-P, Zhou Q, Zhao P-F, Wang Z-Y and Huang L 2020 Using a one-dimensional convolutional neural network with a conditional generative adversarial network to classify plant electrical signals *Comput. Electron. Agric.* **174** 105464
[28] Pereira D R, Papa J P, Saraiva G F R and Souza G M 2018 Automatic classification of plant electrophysiological responses to environmental stimuli using machine learning and interval arithmetic *Comput. Electron. Agric.* **145** 35–42
[29] Thongkham L and Phavaphutanon L 2018 Effect of position and size of leaflets on rooting and rhizome formation of ZZ plant (Zamioculcas zamiifolia (lodd.) Engl.) leaflet cuttings *Agric. Nat. Resour.* **52** 246–9
[30] Kernbach S 2022 *Differential Impedance Spectrometer for Electrochemical and Electrophysiological Analysis of Fluids and Organic Tissues. Handbook and User Manual* (Stuttgart: CYBRES GmbH)
[31] Fernández J E, Alcon F, Diaz-Espejo A, Hernández-Santana V and Cuevas M 2022 Water productivity and economic analyses for super high density olive orchards *Acta Horticulturae* pp 395–402
[32] Kernbach S 2016 Device for measuring the plant physiology and electrophysiology *IJUS* **12-13** 138
[33] Sakurai G and Miklavcic S J 2021 On the efficacy of water transport in leaves. A coupled xylem-phloem model of water and solute transport *Front. Plant Sci.* **12** 615457
[34] Kisnieriene V, Lapeikaite I, Pupkis V and Beilby M J 2019 Modeling the action potential in characeae nitellopsis obtusa: effect of saline stress *Front. Plant Sci.* **10** 82
[35] Dekking F M, Kraaikamp C, Lopuhaä H P and Meester L E 2005 *A Modern Introduction to Probability and Statistics* (London: Springer)
[36] Hjorth B 1970 EEG analysis based on time domain properties *Electroencephalogr. Clin. Neurophysiol.* **29** 306–10
[37] Kugiumtzis D and Tsimpiris A 2010 Measures of analysis of time series (mats): a matlab toolkit for computation of multiple measures on time series data bases *J. Stat. Softw.* **33** 1–30
[38] Brunton S L and Kutz J N 2019 *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems and Control* (Cambridge: Cambridge University Press)
[39] Welch P 1967 The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms *IEEE Trans. Audio Electroacoust.* **15** 70–73
[40] Buss E, Aust T, Wahby M, Rabbel T-L, Kernbach S and Hamann H 2022 Stimulus classification with electrical potential and impedance of living plants: comparing discriminant analysis and deep-learning methods (Supplementary material online) (https://doi.org/10.5281/zenodo.7126105)
[41] Jiao L and Zhao J 2019 A survey on the new generation of deep learning in image processing *IEEE Access* **7** 172231–63
[42] Young T, Hazarika D, Poria S and Cambria E 2018 Recent trends in deep learning based natural language processing *IEEE Comput. Intell. Mag.* **13** 55–75
[43] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
[44] Rakthanmanon T, Campana B, Mueen A, Batista G, Westover B, Zhu Q, Zakaria J and Keogh E 2013 Addressing big data time series: mining trillions of time series subsequences under dynamic time warping *ACM Trans. Knowl. Discovery Data (TKDD)* **7** 1–31
[45] Wang Z, Yan W and Oates T 2017 Time series classification from scratch with deep neural networks: a strong baseline *2017 Int. Joint Conf. on Neural Networks* (*IJCNN*) (IEEE) pp 1578–85
[46] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58
[47] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* vol 25
[48] Zhao B, Lu H, Chen S, Liu J and Wu D 2017 Convolutional neural networks for time series classification *J. Syst. Eng. Electron.* **28** 162–9
[49] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *Int. Conf. on Machine Learning* (PMLR) pp 448–56
[50] Serrà J, Pascual S and Karatzoglou A 2018 Towards a universal neural network encoder for time series *CCIA* pp 120–9
[51] Ulyanov D, Vedaldi A and Lempitsky V 2016 Instance normalization: the missing ingredient for fast stylization (arXiv:1607.08022)
[52] Bahdanau D, Cho K and Bengio Y 2014 Neural machine translation by jointly learning to align and translate (arXiv:1409.0473)
[53] Arthur L G, Malinowski S and Tavenard R 2016 Data augmentation for time series classification using convolutional neural networks *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*
[54] Cui Z, Chen W and Chen Y 2016 Multi-scale convolutional neural networks for time series classification (arXiv:1603.06995)
[55] Zheng Y, Liu Q, Chen E, Yong G and Zhao J L 2014 Time series classification using multi-channels deep convolutional neural networks *Int. Conf. on web-age Information Management* (Springer) pp 298–310
[56] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*CVPR*)
[57] Tanisaro P and Heidemann G 2016 Time series classification using time warping invariant echo state networks *2016 15th IEEE Int. Conf. on Machine Learning and Applications* (*ICMLA*) (IEEE) pp 831–6
[58] Hintze J L and Nelson R D 1998 Violin plots: a box plot-density trace synergism *Am. Stat.* **52** 181–4
[59] Taghavi M and Shoaran M 2019 Hardware complexity analysis of deep neural networks and decision tree ensembles for real-time neural data classification *2019 9th Int. IEEE/EMBS Conf. on Neural Engineering* (*NER*) pp 407–10