IoTree: A Battery-free Wearable System with Biocompatible Sensors for Continuous Tree Health Monitoring

Tuan Dang[1], Trung Tran[2], Khang Nguyen[1], Tien Pham[1], Nhat Pham[4], Tam Vu[3], and Phuc Nguyen[1]
[1] University of Texas at Arlington, [2] Sungkyunkwan University, [3] University of Colorado Boulder, and [4] University of Oxford
{tuan.dang, khang.nguyen, tienan.pham, vp.nguyen}@uta.edu,
tranquangtrungam@gmail.com, nhat.pham@cs.ox.ac.uk, tam.vu@colorado.edu

ABSTRACT

In this paper, we present a low-maintenance, wind-powered, battery-free, biocompatible, tree wearable, and intelligent sensing system, namely *IoTree*, to monitor water and nutrient levels inside a living tree. *IoTree* system includes tiny-size, biocompatible, and implantable sensors that continuously measure the impedance variations inside the living tree's xylem, where water and nutrients are transported from the root to the upper parts. The collected data are then compressed and transmitted to a base station located at up to 1.8 kilometers (approximately 1.1 miles) away. The entire *IoTree* system is powered by wind energy and controlled by an adaptive computing technique called block-based intermittent computing, ensuring the forward progress and data consistency under intermittent power and allowing the firmware to execute with the most optimal memory and energy usage. We prototype *IoTree* that opportunistically performs sensing, data compression, and long-range communication tasks without batteries. During in-lab experiments, *IoTree* also obtains the accuracy of 91.08% and 90.51% in measuring 10 levels of nutrients, NH_3 and K_2O, respectively. While tested with Burkwood Viburnum and White Bird trees in the indoor environment, *IoTree* data strongly correlated with multiple watering and fertilizing events. We also deployed *IoTree* on a grapevine farm for 30 days, and the system is able to provide sufficient measurements every day.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; **Sensor networks**.

KEYWORDS

Sensor, Embedded Systems, Intermittent Computing

ACM Reference Format:
Tuan Dang, Trung Tran, Khang Nguyen, Tien Pham, Nhat Pham, Tam Vu, and Phuc Nguyen. 2022. *IoTree*: A Battery-free Wearable System with Biocompatible Sensors for Continuous Tree Health Monitoring . In *The 28th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '22), October 17–21, 2022, Sydney, NSW, Australia.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3495243.3567652

ACM MobiCom '22, October 17–21, 2022, Sydney, NSW, Australia

ACM ISBN 978-1-4503-9181-8/22/10.
https://doi.org/10.1145/3495243.3567652

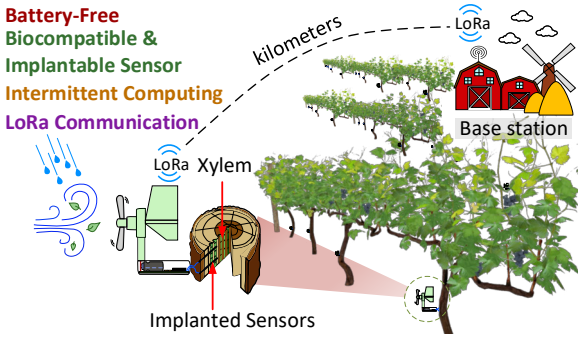

Figure 1: *IoTree*'s Concept [14].

1 INTRODUCTION

Global agriculture will need to produce more food in the next 50 years than in the previous ten thousand years to feed the growing population [1]. In modern agriculture, a few crops known to have high yields are selected to grow intensively [2]. Growing a few varieties of trees makes our food supply vulnerable to pests and diseases, leading to the overuse of fertilizers. Such chemicals make our farmland less productive and the food we grow less nutritious [3–5]. In the U.S., more than 1,700 trillion BTU (about $17B) of energy was for agriculture annually [6, 7], nearly 30% of this was for fertilizer production, yet the nitrogen fertilizer efficiency is 33% globally [8, 9]. Understanding how crops grow will help reduce the use of fertilizers, chemicals, and water and design novel and sophisticated growing techniques like intercropping [10, 11] and cover cropping [12, 13] to restore soil fertility and increase crop productivity.

Advanced soil sensors [15–17], camera-based sensors [18], and sap sample analyses [19] have been proposed to derive water and nutrient uptake behaviors of trees. However, soil sensors provide unreliable insights since the data are not derived from the tree body. Moreover, as these sensors are required to be buried at 40 cm below the ground [20], the deployment cost is high, and the soil-to-air wireless communication is also unreliable [21]. Second, camera-based solutions cannot provide tree's internal signals while performing poorly under low-light, rainy, and dusty conditions. Last but not least, extracted sap samples can be analyzed offline later to infer the water and nutrient contained inside the tree [19], but these techniques are costly [22], complex, time-consuming, and labor intensive [23]. As the measurements are not obtained from the living tree, and the real-time impact of the environments are also unclear [24, 25].

In this project, we develop wind-powered, low maintenance, battery-free, biocompatible, implantable, tree wearable, and intelligent sensing system, namely *IoTree*, that continuously captures

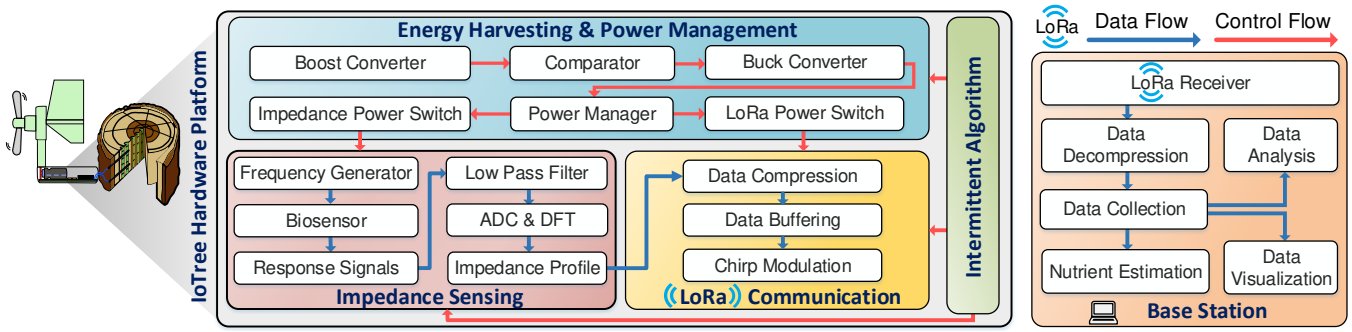

Figure 2: *IoTree*'s system overview.

signals inside a living tree to infer the tree's water and nutrient levels (Fig. 1). *IoTree* system measures impedance variations inside the tree's xylem, where nutrients and water from the root are transported to the upper parts, before compressing the collected data and transmitting to a base station at up to 1.8 km (approximately 1.1 miles) of distance. The entire *IoTree* system is powered by wind and controlled by an adaptive block-based intermittent computing algorithm. Such a battery-free sensing system for trees will provide an opportunity for ultra-long multi-day missions, low maintenance, and a lower ecological impact since plugged-in or battery-powered devices are unsuitable for a dense and long-term deployment. However, realizing this vision is difficult due to the following challenges:

- Many sensors have been developed to monitor human physiological signals, but very few target tree health monitoring. Building robust and biocompatible sensors for tree monitoring faces multiple challenges from design to implementation as the differences between human epidermis composition and tree bark composition.
- Since unpredictable weather conditions (i.e., sunny, rainy, or windy), especially on the farm-settings, could heavily impact the system's performance, building battery-free and low-power sensing hardware and software that can to provide reliable measurements under outdoor settings is a difficult task.
- Just like humans, trees grow over night [26, 27]. Hence, solar-based battery-free computing systems are not well-suited for tree monitoring. We found that wind is a great resource, but developing a wind-based battery-free system is challenging due to frequent power failures caused by unpredictable wind availability.
- *IoTree*'s tasks (sensing, computing, and long-range communication) consume different power levels; the computing technique needs to adaptively update its run-time scheduler to fully utilize the harvested energy and complete the user requirements.

Contributions. In this paper, we make the following contributions. (1) We develop a biocompatible fiber-based sensor to monitor water and nutrient levels inside the tree body. (2) We derive a battery-free and low-power sensing algorithm to measure signals inside the tree body reliably under multiple environmental conditions. (3) We design and implement a block-based intermittent computing algorithm allowing *IoTree* to fully utilize the harvested energy and perform its task with the optimal memory and energy requirements. (4) We prototype *IoTree* that opportunistically performs sense, data compression, and long-range communication without requiring any battery and maintenance. (5) We evaluate the system with in-lab and show that *IoTree* obtains 91.08% and 90.51% of accuracy in measuring 10 levels of nutrients, NH_3 and K_2O, respectively. When tested with Burkwood Viburnum and White Bird trees in the indoor environment, *IoTree* data strongly correlate with multiple soil stimuli (watering and fertilizing) events. We also deployed *IoTree* in the wild for 30 days, and the results show that *IoTree* is able to provide sufficient measurements every day. The system reports 558 measurements a day with a distance of up to 1.8 kilometers without requiring any batteries or maintenance.

Potential Applications. *IoTree* platform has potential to be used in precision agriculture [28], pH monitoring [29], global warming [30], crop monitoring [31, 32], plant physiology research [33], plant disease monitoring and pest control [34], forest monitoring [35], to name a few. Beside tree monitoring applications, the proposed system can be used as a soil sensor to monitor pH level and other information in a forest or contaminated areas [29, 35]. Additionally, the current single-hop network architecture can be extended to multi-hop network architecture to be used larger farms or in national forests, typically in cases where the deployment area is relatively large (approximately a few hundred kilometers squared [36]). The hardware design, firmware, and software libraries of *IoTree* will be made publicly available. The open 3D design, hardware, and software libraries can be used as an inexpensive educational tool, allowing graduate students, phytologists, bioengineers, and hobbyists to access signals inside the tree body in real-time.

2 FUNDAMENTALS OF TREE SENSING

IoTree relies on the unique chemoelectrical relationship between water, nitrogen (N), and potassium (K) ion levels inside the xylem sap of trees and the measured impedance levels. In this section, we will provide the background on ion ratio inside the living tree and why water/nutrient level changes will create impedance variations.

Ion Ratio Inside Living Trees. This study focuses on water, nitrogen, and potassium monitoring since they are the most important substance in tree development [37–39]. Other mineral elements such as phosphorus (P), calcium (Ca), magnesium (Mg), sulfur (S), and sodium (Na) also have vital factors in facilitating metabolism [38]. However, we reserve these studies for future work. To be specific, tree roots take up nutrients from the soil by absorbing their ions. Transceptor proteins of the roots regulate the nutrient uptake through an absorption process, causing by the difference between inside and outside ion concentrations of the roots [40]. After being taken by the

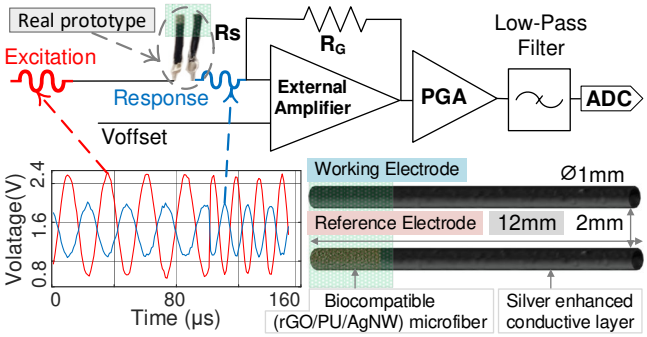

Figure 3: The developed sensors and overview working principle of the impedance sensing circuit.

root through the nitrogen fixation process, nitrogen is transported inside the tree under the form of nitrate (NO_3^-), nitrite (NO_2^-), or ammonium (NH_4^+) [40]. The nitrogen for healthy trees is maintained from 3% to 4% in their tissues [41], while the concentration of potassium (K^+) at 80-100 mM inside the tree cells [42].

Impedance Variations Created by Ion Changes. As the roots uptake nutrients from the soil and transport their ions through the xylem, the impedance changes accordingly to each ion's concentration [43]. Previous vitro experiments [44, 45] showed a strong correlation between the measured impedance spectra (from 1 Hz to 1 MHz) and a wide range of NO_3^- ion concentrations. They also confirmed that the resistance-capacitance parallel circuit model can be used to explain the behavior of ionic charges in ionic conductors [46]. Further studies [47, 48] also found similar results on the relationship between the concentration of nutrient ions (K^+, NH_4^+) and impedance spectra of its medium. This relationship will be later confirmed in our experiments in Fig. 11 (Sec. 8.1).

To sum up, we can infer tree's health from ion levels by measuring the impedance level inside the xylem; however, multiple hinders need to be overcome to make such a reliable system. First, a biocompatible solution needs to be developed; indeed, biocompatible sensors have been developed for human [49, 50], their applicability for tree is unclear. Second, battery-powered devices are not good choices for large-scale and long-term deployment in farm settings. However, existing battery-free solutions are solar-based setups [51–53] and their concepts are unsuitable for overnight and continuous sensing. In following sections, we will describe the proposed system to address the above challenges, followed by the detailed design, implementation, and evaluation with in-lab and outdoor experiments.

3 *IOTREE* SYSTEM OVERVIEW

We design and implement *IoTree* with the following goals. First, the sensors should be implanted onto the tree's tissue to capture the signals inside the tree. Second, *IoTree* must require the least maintenance since it will be deployed on a large scale, and frequent charging or replacing batteries is not practical in agriculture settings. Thus, the system has to harvest the wind energy and use the collected energy optimally by intrinsically re-configuring the program into blocks. Last but not least, *IoTree* must be able to transmit data at a long distance, particularly at a distance of the size of typical farms. To meet these goals, we design *IoTree* as shown in Fig. 2. The details of each component are as follows:

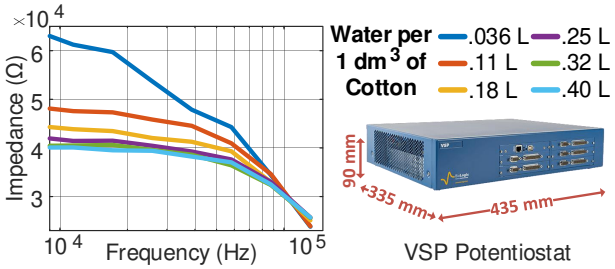

Figure 4: Impedance profile measured by VSP Potentiostat [61].

Biocompatible Sensors. We develop impedance-based sensors that consists of two fiber electrodes made by biocompatible materials (Sec. 4.1). The sensing circuit generates sweeping frequency signals and measures the responses to calculate the impedance. Multiple pairs of sweeping frequencies and the value calculated from Discrete Fourier Transform are used to form impedance profiles to infer the water and nutrient levels (Sec. 4.2).

Energy Harvesting & Power Management. *IoTree* system operates by the wind energy (Sec. 5). We design a lightweight wind energy harvester, including a custom-built wind indicator to maximize the harvested power. Additionally, we also carefully analyze the wind behaviors in the deployed area to ensure that the harvested energy is sufficient to perform *IoTree*'s important tasks.

Long-range Communication. Since *IoTree* will be deployed on-farm settings, it needs to transmit data on the farm to the base station reliably. The average farm size in the U.S. is 444 acres [54]; hence, the typical communicating distance requirement for the device is 0.59 miles (0.95 km). Therefore, we adopt LoRa communication to enable reliable, low-noise, and low-power communication. *IoTree* reports data continuously to the base station at 1.8 kilometers (Sec. 5).

Block-based Intermittent Computing. We propose a block-based intermittent computing approach that is able to intrinsically change the number of blocks and the size of each block to minimize the memory overhead and wasted energy compared to state-of-the-art intermittent computing solutions (i.e., task-based [55, 56] and checkpoint-based [57–60] approaches). The proposed block-based approach adaptively changes the number of blocks and the size of each block to minimize the memory overhead and wasted energy. After rebooting, *IoTree* can recover from the last executed block in the task instead of restarting from the beginning of the task (Sec. 6).

4 SENSOR DEVELOPMENT

4.1 Sensor Design & Fabrication

As *IoTree*'s sensors are implanted inside the tree, it needs to be tiny-sized and biocompatible. We propose a fiber-based impedance sensor that meets our design requirements. We fabricate conductive fibers and use them to manufacture fiber-based impedance sensors. The details of fabrication process are as follows:

Fabrication of Conductive Microfibers: The conductive microfiber are made from reduced graphene oxide (rGO), polyurethane (PU), and silver nanowires (Ag-NWs) via a wet spinning method [62, 63]. Firstly, PU pellets was dissolved dimethylacetamide (DMAC) to form PU solution. AgNWs and GO dispersed in DMAC were mixed with PU solution to form a homogeneous mixed solution. Hence,

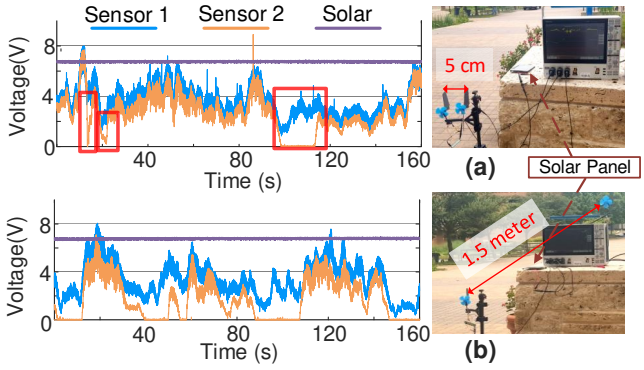

Figure 5: Wind energy distribution at different distances in compared to solar energy.

this solution was injected into a DI water coagulation bath through a single steel nozzle to form the GO/PU/AgNWs gel microfibers. Next, the gel is reduced by the ascorbic acid solution to generate conductive soft rGO/PU/AgNWs microfibers. Lastly, the substance is taken out and dried at $120°C$ for two hours in a vacuum oven.

Fabrication of Impedance Sensors: The sensor includes two conductive soft rGO/PU/AgNW microfiber electrodes as shown in Fig. 3. The microfibers are directly used as a working electrode (WE). To develop the reference electrode (RE), we coat the rGO/PU/AgNW microfibers in Ag/AgCl. Then, the WE and RE microfiber electrodes are aligned in parallel with a distance of 2 mm between each other to form a soft impedance sensor. The rGO and AgNWs were embedded in PU. The PU is a segmented polymer with soft segment to provide high flexibility and hard segment to provide high strength and good resistance to water and chemical [50, 64, 65] . Moreover, PU also presents unique chemistry with a molecular structure similar to that of human proteins. Thus, the impedance sensors made from rGO/PU/AgNWs microfibers will provide high biocompatibility, and a great facility for living inside trees [49, 50].

In-lab Sensors Sensitivity Validation. We validate sensors using VSP Potentiostat device [61]. Fig. 4 shows the impedance profiles measured by the device, showing the visual distinguishable differencé among amounts of water applied to the same volume, yielding that the sensors are sensitive and reliable. However, this VSP Potentiostat is heavy (9 kg), big (95 mm × 435 mm × 335 mm), and expensive ($4k). It is not suitable for outdoor tree monitoring.

4.2 Sensing Technique

Sensing Principle. We develop an inexpensive, low-power, and small-sized sensing circuit to replace VSP Potentiostat [61] with the working principle shown in Fig.3. In particular, we apply an excitation voltage signal U on one electrode to measure impedance, and the resulting current I is measured at another electrode. Assuming an alternating current with frequency f applied to the first electrode (the applied voltage is $v(t) = U_0 cos(\omega t)$) creates current $i(t) = I_0 cos(\omega t + \varphi)$ in the second electrode, where (φ) is the phase shift between $v(t)$ and $i(t)$; $\omega = 2\pi f$. Impedance Z can be presented in polar form: $Z = |Z|e^{j\varphi} = |Z|(cos(\varphi) + jsin(\varphi))$, where $|Z|$ is the ratio between voltage amplitude and current amplitude. Another form representing impedance uses the Cartesian form: $Z = R + jX$, where

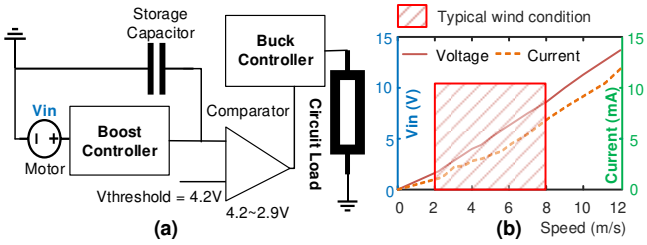

Figure 6: (a) Harvesting circuit diagram (b) Voltage and current of energy harvested at different wind speed.

R is real value, and X is imaginary value. The transformation from polar form to Cartesian form is calculated by $R = |Z|cos(\varphi)$, and $X = |Z|sin(\varphi)$, while the transformation from the Cartesian form to the polar form is calculated by $|Z| = \sqrt{R^2 + I^2}$, and $\varphi = tan^{-1}(I/R)$.

Signal Processing. Due to the high impedance of the monitoring object (from 200 KΩ to 1 MΩ), the response signals are feeble. Therefore, these signals are amplified by the chain of two amplifiers before being filtered and sampled. This chain includes one external amplifier and one on-chip PGA. Both amplifier gains can be configured by adjusting the value of R_G and G_A. Consequently, the signals at the input pin of the Low-Pass filter can be calculated by $V_r = (R_G/R_S) \cdot G_A \cdot V_x$ before being sampled by ADC, where R_G is gain resistor value, R_S is object resistance value, G_A is PGA gain factor, and V_x is excitation signals. For each frequency, the impedance $Z(f)$ is calculated by DFT using equation: $Z(f) = \sum_{n=0}^{1023}(x(n)(cos(n) - j \cdot sin(n)))$, where $Z(f)$ is the impedance at frequency f, $x(n)$ is the sampling result at the time n. As a result, we can calculate the real and imaginary values from $Z(f)$. Multiple-frequency sweeping technique is used to ensure the diverse characteristics of the substances retrieved from the response signals. The schematic of the impedance circuit is shown in Fig. 3.

Sensing Circuit. We build the circuit based on impedance analyzer chip AD5933 [66]. We select this Integrated Circuit (IC) because of its reliability and low power consumption (11 mA in operation mode). The IC includes a frequency generator that modulates 1 Hz to 1 MHz excitation signals. The response signals will be sampled by 12-bit ADC and calculated by Discrete Fourier Transform (DFT) supported by hardware. We also design an external amplifier that works along with an internal programmable gain amplifier. After sweeping one specific frequency, we can get four bytes (two bytes for real value and two for imaginary value) resulting from DFT processing. Every four bytes are stored in the FRAM and managed by the block-based computing algorithm (Sec. 6).

No.Measurements	No.Bytes	LZW Ratio	Huffman Ratio
1	64	0.72	1.55
2	128	0.61	1.27
3	192	0.56	1.09
4	256	0.55	1.02

Table 1: Compression ratio for *IoTree*'s data

Data Compression. Due to the high energy consumption of the communication task, data compression is needed for *IoTree* communication. We adopt LZW [67] for data compression thanks to its simplicity and reliability. LZW avoids the overhead of sending

a dictionary used to decode data like Huffman coding [68]. More importantly, LZW can work well if the input data is sufficient. Our implementation of impedance profile measurement includes 64 bytes, therein 16 sweeping frequencies, each occupying four bytes. Our observation on data collection is that one measure may be slightly different from others if the data are collected close in time. If we send raw bytes data, the compression algorithm cannot work well because each byte entry value distributes in a range of from 0 to 255. However, most differential data between the two measures are distributed from 0 to 50, so the room for data compression is intensified. We define the compression ratio as output size divided by input size, and Table 1 shows that LZW outperforms Huffman with *IoTree*'s sensing data. The more data we compress, the better performance we obtain. We choose the number of measurements based on application need and MCU's memory size.

5 WIRELESS COMMUNICATION & BATTERY-FREE DESIGN

5.1 Long-Range Wireless Communication

IoTree supports long-range communication to ensure that the system meets the requirement of agriculture deployment. We examine various solutions including NB-IoT [69] and LTE-M [70], Wi-Fi [71], Backscatter [72], LoRa backscatter [73], and LoWAN [74]. NB-IoT and LTE-M provide promising concepts of low-powered and long-range communication allowing the device to upload data directly to the internet. However, current 5G NB-IoT and LTE-M prototypes are only available from a few vendors, and they consume high power and require a power outlet supply. More importantly, they do not support custom-built implementation and are not free of charge. Backscatter and LoRa backscatter are promising concepts. They are also considered as the most suitable designs for *IoTree*. However, Backscatter supports a short distance of communication while LoRa-backscatter faces challenges in concurrent communications [73, 75]. Wi-Fi and Bluetooth 4.0 [76] are two other popular techniques that can be used for *IoTree*. However, Wi-Fi consumes high power while Bluetooth only provides a short distance of communication.

To confirm the correctness and feasibility of designing *IoTree* 's wireless communication, we conduct experiments to evaluate two commodities of wireless communication technologies, including Bluetooth and Wi-Fi, by two metrics: *communication distance* and *energy consumption*. We set up the communication experiments in the open space with the light of sight (LOS) condition. The Wi-Fi module can communicate up to 50 meters and consume peak power of 800 mW, while the Bluetooth module reaches 10 meters and consume the power of 180 mW. Therefore, we decide to use LoRa as the main communication protocol for *IoTree* to ensure the system's usability, reliability, and practicality. While LoRa consumes more power than state-of-the-art low-power long-range communication such as LoRa Backscatter [73] or modified Bluetooth 4.0 [77], our hardware design can be easier integrate with LoRa Backscatter or Bluetooth 4.0 hardware if they are available in the market. We implement Lora module using RFIC SX1276 from SEMTECH [78]. The IC operates at frequency 915 MHz and consumes 100 mW at a maximum transmitting power of 20 dBm. In an ideal environment with the LOS conditions, LoRa communication should support communication distance up to 3 kilometers.

5.2 Battery-free Design

Rechargeable batteries can be used to power the *IoTree* for long-term deployment (e.g., a 500 mAh LiPo battery can power the entire system for up to 34 hours). However, batteries have some common limitations, such as limited charging cycles, substantial heat generation, and degradation over time. They must be replaced after roughly three years, even with careful and regular maintenance. We examined multiple energy harvesting methods including solar-based [79], piezoflag-based [80], heat-based [81], wind-based [82]. The solar-based method provides reliable and constant power, but its performance is reduced when the light is blocked by the tree. In addition, it does not work during the night. Heat-based energy-harvesting prototype with cm^2 size does not provide sufficient energy to power the circuit [81]. The flag-based approach does not provide efficient power (average of 3.1 mW) within the same period of time compared to the turbine-based approach (average of 8.0 mW). Furthermore, the piezoelectric flags ($160) are much more expensive than the motor for the wind turbine ($2). Hence, wind energy is the most suitable and practical solution.

We develop an energy harvesting circuit as shown in Fig. 6 (a). The harvesting circuit includes a boost controller and a buck controller, optimizing the conversion performance from observed wind power to usable energy. The boost controller harvests energy from the lowest power (2.2 V, 1.1 mA at 2.2 m/s) up to the highest power (12 V, 13.7 mA at 12 m/s) of the motor to charge a storage capacitor while typical wind speed in the testing area is from 2 m/s to 8 m/s as shown in Fig. 6 (b). This capacitor plays a role as a static energy buffer to power the circuit. When the storage capacitor's voltage reaches the defined threshold, the comparator triggers the buck controller to power the primary computing circuit.

Wind energy is unpredictable and non-uniformly distributed in any geographical area. To validate this theory, we operate two harvesting circuits at a distance of 5 cm and 1.5 m and measure voltage from these two circuits. We also measure the solar energy at these locations. Fig. 5 (a) and Fig. 5 (b) show the capacitor voltages from two wind harvesters that are 5 cm and 1.5 m apart, respectively. As can be seen from these figures, the harvested wind energy is different in both cases, while the energy obtained from solar power is almost constant. In Sec. 6, we will discuss why that is the greatest hinders in designing the battery-free wind-powered system and describe our proposed solution to well-address it.

6 BLOCK-BASED COMPUTING

Motivation. A main objective of designing a battery-free system is to maintain forward progress and data consistency [55]. Although efforts have been made to deal with intermittent characteristics of harvested energy in battery-free systems, existing works focus mostly on solar-based systems [83, 84]. However, these solar-based techniques are not well-suited to tree health monitoring as trees grow overnight exactly when solar energy is unavailable. Moreover, existing intermittent computing techniques are limited in terms of energy consumption. First, the checkpoint-based approach takes snapshots of the entire system, including stack functions, local variables, global variables, and others before power failure, and then restores these snapshots after the power returns. While this technique is extremely efficient in energy utilization since the code will be resumed at the

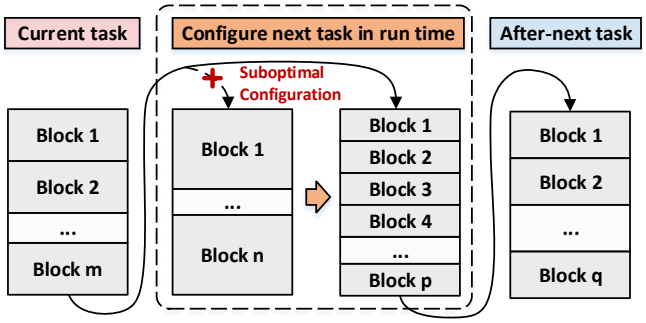

Figure 7: Adaptive block size illustration: a task that has unideal block numbers and block size will be changed to optimal configuration, which minimizes wasted energy.

precise location, they result in much memory overhead due to storing, restoring, and rebooting efforts and some lack of consideration of data consistency caused by idempotent violation [56, 85, 86]. Second, the task-based approach stores the states of executed tasks before power failure; the unfinished task has to be re-executed from the beginning of the task. By doing so, the system only needs to store/restore the states of the completed tasks instead of the entire system. This method is efficient in minimizing the memory overhead. It is, however, inefficient in energy utilization because if the power fails in the middle of the task, that task is going to be re-executed from the beginning. Lastly, most of the state-of-the-art intermittent focus on optimizing energy buffer for MCU (computing), while other tasks use separated energy buffers, causing a lack of synchronization between energy buffers, and energy is hard to manage [86].

HarvOS [86] – a checkpoint-based technique – strives to minimize wasted energy by seeking out the minimum number of checkpoints for an arbitrary program. The optimal checkpoints are the ones that are closest to the last practical checkpoints. Undoubtedly, HarvOS well-performs in systems with consistent energy consumption, but it does not guarantee the most optimal location for checkpoints in a program that requires different energy levels to execute a code segment iteratively (i.e., sensing). In *IoTree*, the energy consumption varies when performing multiple sweeping frequencies in the sensing task. In these scenarios, many energy levels are required to execute tasks, making HarvOS challenging in locating where the best checkpoints should be. Therefore, HarvOS is not well-suited for a device performing sensing tasks, particularly in the *IoTree* use cases. Also, HarvOS places the trigger calls for checkpoints based on Control Flow Graph (CFG) when analyzing the program during compilation which cannot be applied in the *IoTree* program, especially when sensing tasks requires different energy levels.

CleanCut [87] – another recent intermittent computing technique – investigates to prevent a program from trapping in non-terminated tasks. CleanCut continually compares the available energy and energy needed to perform a task in order to prevent one task from looping forever, a common issue of checkpoint-based and task-based approaches. Establishing task boundaries [55] ensures data consistency and forward progress, but lacks of awareness of multiple energy levels within a task. Also, the CleanCut's energy modeling does not correspond to the actual energy requirement of each task. Due to static code analyses, the developer must turn in the energy-related parameters when porting to different hardware platforms.

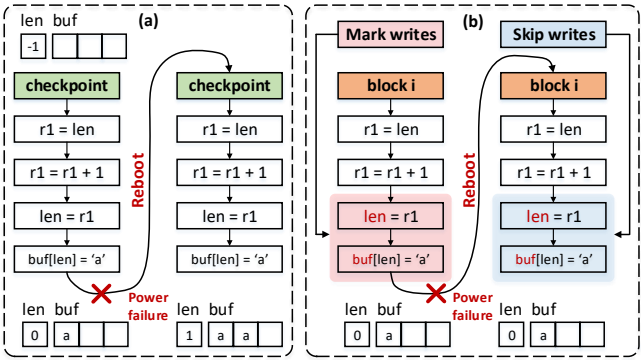

Figure 8: Non-volatile buffer manager avoids idempotent violation.

Consequently, it does not well-match the *IoTree* system since the energy level varies at different sweeping frequencies while performing impedance profile measurements. In other application use cases, the frequency set may alter corresponding to specific requirements, influencing the accurateness of CleanCut's energy modeling.

IoTree's block-based approach and Coala [88] both aim to divide over-fitted tasks into smaller sub-tasks, or blocks, to ensure the executed code size fits well to residual energy. However, Coala attempts to minimize the transitional overhead energy between tasks while *IoTree* tries to preserve sensing results and transmitting data state as the transition energy consumption is insignificant compared to the required energy in sensing and transmitting tasks in *IoTree* use cases. When dividing a task into blocks, each block must complete a work portion (i.e., portion of sensing data, portion of transmitting data). Thus, avoiding re-executing a code segment associated with the same sensing or communication tasks is the primary goal of the *IoTree* system. Moreover, Coala prevents data inconsistency over coalesced tasks using the blend of Virtual Memory Management (VMM) and Direct Memory Access (DMA), which cannot be applied an unsupported DMA hardware platforms like *IoTree*. In contrast, each block in the block-based approach shares the task buffer data that is committed only when a block completes its execution. The buffer manager controls the read/write operations from a block into the task buffer, ensuring data consistency when the execution flow transits from one block to another. Each block accesses the task buffer (non-volatile memory) and only copies related data to the volatile local buffer, and the volatile local buffer will be updated to the task buffer after a completed block execution (Fig. 8).

Block-based Computing. We propose a blocked-based approach to overcome the limitations mentioned above. The key idea is to boost energy utilization by adaptively changing the amount of codes to be executed before power failure, and to minimize overhead memory by check-pointing only replicas of the system's snapshots (i.e., pointers). In particular, we divide the task into multiple blocks whose sizes can be adaptively updated during run-time depending on the energy availability. Fig. 7 illustrates an example of how block-based approach works during run-time. A block-based program comprises tasks, blocks, and tasks containing many blocks within themselves. Blocks are built and assembled to form tasks. Thus, the input and output buffers for each task are allocated in order to create communication between the tasks. Note that any block in the block-based program neither allows users to use non-volatile or

Algorithm 1: Block Size Anticipation

Input : E_a := available energy in budget
tid := identification of the next task
Output : block_size := block size of next task
nblock := number of blocks of next task
E_r := remaining energy

```
function look_ahead(E_a, tid)
    block_size, nblock ← get_task_configuration(tid)
    E_t ← estimate_energy(tid)
    if E_a < E_t then
        if E_t/E_a = ⌊E_t/E_a⌋ + 1 then
            nblock ← E_t/E_a
        else nblock ← ⌊E_t/E_a⌋ + 1
        block_size ← E_t/nblock
        if nblock ≠ (get_number_block(tid)) then
            E_r ← E_a − nblock ·E_o − E_o
        else E_r ← E_a − nblock ·E_o
    else E_r ← E_a − E_t
return block_size, nblock, E_r
```

Algorithm 2: Adaptive Block Size using Anticipation

Input: E_c := buffer energy in capacitor
Output: $(n_1, n_2, ..., n_i)$:= Tasks' optimal configuration

```
(T_1, T_2, ..., T_i) ← compute_task_execution_time()
(E_1, E_2, ..., T_i) ← estimate_task_energy(T_1, T_2, ..., T_i)
E_a ← E_c
while E_a > 0 do
    tid ← get_next_task_id()
    block_size, n_tid, E_r ← look_ahead(E_a, tid)
    configure_task(tid, nblock, block_size)
    execute_task(tid)
    E_a ← E_r
```

global variables. There are two types of blocks: iterative and single blocks. An iterative block is equivalent to a loop over a function with different parameters (i.e., sweeping over different frequencies), while a single block is equivalent to a code sequence (i.e., task initialization or returning results). Still, both are atomic sequences [55] meaning that the code outside a block cannot observe its effects.

IoTree computing system manages to obtain the most optimal number of blocks for the tasks in terms of wasted energy. Specifically, we build a sequential pipeline, including a theoretical optimization in compile-time and an empirical optimization in run-time. For the theoretical optimization stage (i.e., compiling time phase), the hypothesized solution is the trivial solution based on the power consumption ratio among three tasks computing from the machine code distribution of each task on FRAM. Afterward, the hypothesized solution is served as initial block numbers in the empirical optimization stage, which only occurs when the system is deployed (i.e., during its run-time phase). The empirical optimization stage uses Alg. 1 as a configuration method to adjust the hypothesized parameters to be a more precise solution (i.e., near-optimal block size and the number of blocks) in the system's run-time (Alg. 2). This process returns the most optimal set-of-integer of block size and the number of blocks

The approach is feasible if and only if (i) there is *a finite set of solutions* for the number of blocks within a task and (ii) the algorithm finding the optimal block size converges. If these two conditions satisfy, the optimal block size and number of blocks (of the same size) will be obtained. For (i), the number of blocks solution set for a task cannot be infinite in practice since the size of one block cannot subceed the overhead unit. Otherwise, the task is filled with overhead units and cannot be executed. Additionally, by the well-ordering principle [89], we can confirm that there always exists a set-of-integer solution in the finite set that makes the system waste the least energy in our particular case. For (ii), we observe that Alg. 2 always returns solutions that reach the optimal solution in run-time for multiple test cases, including varying the capacitor size and task sizes. From test cases, the convergence of wasted energy in a power cycle, $E_w^{(i)}$, can be modeled in terms of the residual energy of the current power cycle, $E_r^{(i)}$, the required energy for task "j" in current power cycle, $E_j^{(i)}$, and the capacitor size, E_c, as follows:

$$\lim_{i\to\infty}\left(\sum E_w^{(i)} \Big/ \sum E_c\right) = \text{const, where } E_w^{(i)} = \left(E_r^{(i)} - \frac{E_j^{(i)}}{\lfloor E_j^{(i)}/E_r^{(i)}\rfloor+1}\right)$$

Hence, (i) *the finite solution set* and (ii) *the improved solution obtained from the algorithm during run-time* are verified, we conclude that *IoTree* can find the optimal solution to minimize wasted energy. This approach's energy consumption will be evaluated in Sec. 8.2.

Forward Progress & Data Consistency. Block-based approach maintains forward progress by saving the task pointer whenever the system jumps into the task and saves the current block pointer of that task. To maintain data consistency, we introduce the buffer manager, preventing multiple writes into output buffers when a task resumes from the same block multiple times. Fig. 8 (a) shows a typical idempotent violation, which produces two different results when executing the same code segment. Fig.8 (b) illustrates how buffer manager effectively controls the "write" and "skip write" data, avoiding data inconsistency.

Block-based Library Implementation. We implement the block-based run-time library using the C programming language, the ideal language in embedded system development. The library aims to give users a set of Application Programming Interface (API) to develop an intermittent application effortlessly. These APIs include setting up tasks and blocks. The user initially needs to create tasks by using *create_task(name, &task_id)*, connect them by *set_transition(task_id1, task_id2)*, and set input-output buffers by *set_buffer(task_id, buffer, buffer_type)* for each task. Before setting buffer for a task, we need to allocate memory for that buffer by *alloc_buffer(size)*. Non-volatile memory is the crucial component to save a program state before a power failure so that the application can continue from that failure point. Creating buffers with help from the non-volatile memory manager provided by the run-time library enables programmers to write an intermittent program efficiently. The library allows them to focus on their algorithms and let the run-time library manage all global non-volatile variables, which are potential causes of data inconsistency. Next, the user must create block handlers to manage block control and set these block handlers as well as the number of blocks for each task by using *set_block_handler(task_id, block_func, number_blocks)*. The system will execute the setup code only once at the first power cycle, then run as scheduled unless programmers

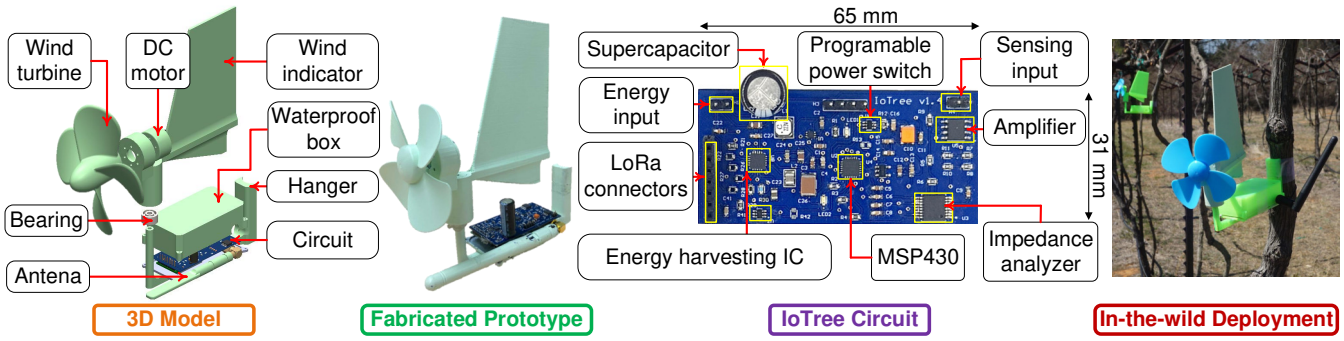

Figure 9: 3D Model, fabricated prototype, integrated circuit, and real-world deployment of *IoTree* system.

	Idle	Sensing	Computing	Transmitting
Power (mW)	1	62	6	245
Duration (ms)	0.3	170	80	150

Table 2: *IoTree*'s power consumption & typical execution time

want some changes at the run-time, such as re-wiring task transitions for special cases.

Block-based Implementation in *IoTree* system. *IoTree* includes three main tasks: sensing, data compression, and communication; each task is adaptively divided into blocks using the block-based run-time library. The power consumption and typical execution time of each task are shown in Table 2. A task must be accomplished before the system proceeds to the next task, and each of them reads the input from the previous task and produces the output for the subsequent one.

Due to the imperfection of electronic components, even the operational power of a task fluctuate over time, leading to many difficulties in obtaining optimal block size and number of block. Considering the sensing task as an example, our system captures the capacitance response created by sweeping 16 frequencies signals. Fig. 10 illustrates the energy consumed at frequency 1kHz, 10kHz, and 100kHz. The energy consumes by each frequency is different, and frequent power failures make it extremely difficult for the system to identify the best block size and number of blocks. Fortunately, the variation of the consumed power of each task is predictable. If the system obtains sufficient knowledge of how each task consumes power during its run-time, it will be able to identify the best block size and block number. The starting point of the energy ratio between tasks and the storage capacitor is provided by the user/programmer during compile time. Block size and number of blocks are then obtained automatically during the installation phase of the system.

7 SYSTEM IMPLEMENTATION

We design, fabricate, and implement *IoTree* system using COTS materials and electronic components as illustrated in Fig. 9. In the following subsections, we describe the 3D model, PCB, implant process, and sensor calibration. The process of design and fabricate the impedance sensor is described in Sec. 4.1. We also develop a software running on an MS Windows computer, equipped with a LoRa receiver, to continuously receive the signals from the *IoTree* nodes.

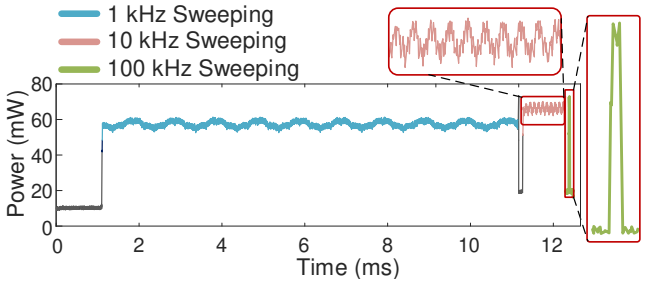

Figure 10: Sweeping frequency power consumption: 1 KHz, 10 KHz, and 100 KHz.

3D Model Design & Fabrication. *IoTree* 3D prototype is designed using SOLIDWORKS. We choose $\phi 24$ DC motor and wind turbine ($\phi 80$) for the energy harvester. We design a wind indicator that supports 180-degree freedom of rotation to steer the turbine towards the direction of the wind [14]. A hanger is designed at one end of the prototype to help attaching the device to the body of the tree. The vertical pole holding the turbine is a tube that allows the electrical wires to connect the motor and the circuit. This structure will help to avoid blockage caused by twisted wires around the prototype when the wind indicator rotates in one direction for certain cycles. A small bearing is used to reduce friction when the wind indicator rotates around the vertical axis. The whole system, including the circuit, weights 115.85 grams with the size of 145 mm $\times$ 170 mm.

PCB Design & Implementation. We design and fabricate *IoTree*'s circuit as shown in Fig. 9. The circuit has a size of 31.5 mm $\times$ 65 mm, weights 4 grams, includes two layers with 0.8mm thickness. MSP430FR2433 (126 μA/MHz, 16KB FRAM, 4KB SRAM) is the central control unit of *IoTree*'s circuit. To harvest energy at the low wind speed (2 m/s, 1.5 V), we use BQ25570, which supports a power boost charger to charge a storage capacitor up to 4.2 V from the lower input voltage. The energy harvester also allows the maximum power input of 510 mW that is able to sustain for maximum wind speed (12 m/s, 13.7 voltage, 12 mA, 164.3 mW) as the experiment depicted in Fig. 6. MCU communicates with impedance analyzer circuit via I2C and with RF components via UART. We use programmable power switch integrated circuits (TPS22919DCKT IC) to turn on or turn off different circuit components according to the requirement of the current task to force the component to go to sleep if they are not performing any task. The amount of energy buffer is calculated

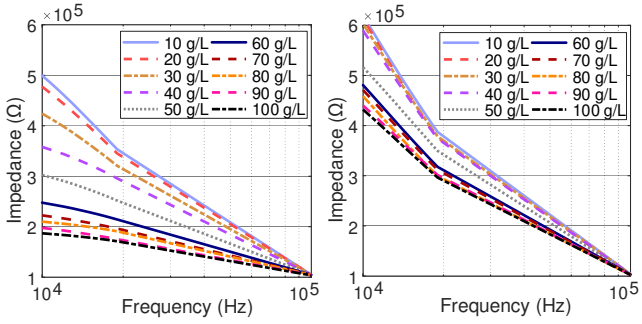

Figure 11: In-lab experiments with Nitrogen and Potassium: impedance profiles for different level of nutrients.

by $Ec = (1/2) \cdot C \cdot (V_{max}^2 - V_{min}^2)$, where V_{max} is the upper bound voltage to trigger starting of the circuit and V_{min} is the lower bound voltage where the circuit turns off.

Implanting Sensors into The Tree's Xylem. We first calibrate the sensors after they are manufactured to obtain the calibration constant since every pair of sensors might have a different distance due to the manufacturing imperfection. We hence drill two tiny holes on the tree with a depth of three millimeters, then we put a portion of mixture 500 mg catalase and 50 ml conductive gel into these holes to avoid the reduction in conductivity level [90]. The catalase prevents the defense mechanism of the tree, which will lead to compartmentalization and reduced sensor accuracy [91] while conductive gel increases conductivity and moisture between two electrodes and tree xylem [92]. We place two sensor electrodes into two holes and shield these holes with adhesive tape to make sure that no air can come into the holes.

Sensors Calibration. We develop a compact model to map the measured impedance profile to the corresponding water and nutrient levels. For each measurement, we obtain an impedance profile vector of $k+1$ elements, $\vec{x}_{raw} = (1, x_1, x_2, ..., x_k)$; each element from x_1 to x_k respectively represents for the impedance being measured at each frequency, ranging from f_1 to f_k ($k = 16$ in our implementation). Depending on each input impedance profile $\vec{x}_{raw}$, the corresponding predicted nutrient level, $\hat{y}$, will be calculated as follows:

$$\hat{y} = \sum_{i=0}^{k} \beta_i x_i + \sum_{i \neq j \neq 0}^{k} \beta_{i,j} x_i x_j + \sum_{i=1}^{k} \beta_{i,i} x_i^2 \quad (1)$$

Leveraging Eq. 1, our data set could be designed into the matrix form as: $\vec{y} = \mathbf{X}\vec{\beta} + \vec{\varepsilon}$, where $\mathbf{X}$ is a matrix composing of n vectors in the form of $\vec{x}$ (with $\vec{x} = (1, x_1, .., x_k, x_1x_2, x_1x_3, .., x_{k-1}x_k, x_1^2, x_2^2, .., x_k^2)$), $\vec{y}$ is the vector of nutrient levels corresponding to the impedance profiles, $\vec{\beta} = (\beta_0, \beta_1, .., \beta_k, \beta_{1,2}, \beta_{1,3}, .., \beta_{k-1,k}, \beta_{1,1}, \beta_{2,2}, .., \beta_{k,k})^\intercal$, and $\vec{\varepsilon} = (\varepsilon_1, \varepsilon_2, .., \varepsilon_n)^\intercal$ is the residual value vector for n measurements. To seek out the most suitable $\vec{\beta}$, we use the Iteratively Reweighted Least Squares (IRLS) algorithm that allows us to update $\vec{\beta}$ iteratively. By establishing an objective function in L_p spaces, Lebesgue spaces, for the quadratic regression as follows:

$$\text{argmin}_{\vec{\beta}} ||\vec{y} - \mathbf{X}\vec{\beta}||_p = \text{argmin}_{\vec{\beta}} \sum_{i=1}^{n} |y_i - \mathbf{X}_i\beta|^p \quad (2)$$

where $\text{argmin}_{\vec{\beta}}$ is the minimizer returning coefficient vector, $\vec{\beta}$, at

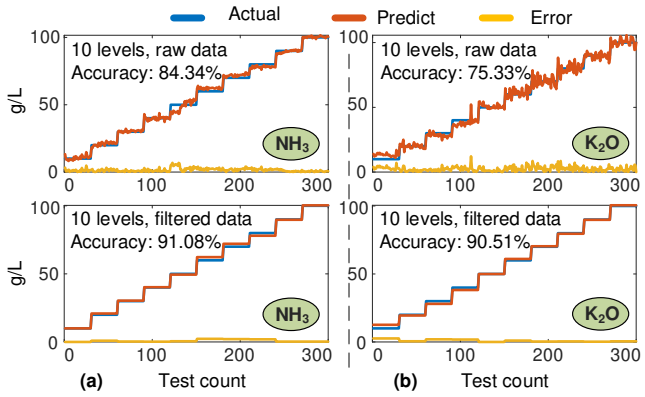

Figure 12: (a) The accuracies of sensing 10 levels of Nitrogen (NH_3) at 10 g/L resolution, and (b) The accuracies of sensing 10 levels of Potassium (K_2O) at 10 g/L resolution.

which the polynomial is minimum, $||x||_p$ is defined by $(\sum_{i=1}^{n} |x_n|^p)^{1/p}$ ($p \in \mathbb{R} \mid p \geq 1$), and $\mathbf{X}_i$ is the i^{th} row of $\mathbf{X}$.

To solve Eq. 2, we get initial values for $\vec{\beta}$ and $\mathbf{W}$. The initial $\vec{\beta}$, $\vec{\beta}^{(0)}$, is the resulting $\vec{\beta}$ from Weighted Least Squares algorithm running on our dataset. The initial diagonal weight matrix $\mathbf{W}$, $\mathbf{W}^{(0)}$, is assumed as the identity matrix $\mathbf{I}_n$. In the loop, $\vec{\beta}$ and $\mathbf{W}$ will be automatically updated by: $\vec{\beta}^{(k+1)} = \text{argmin}_{\vec{\beta}} \sum_{i=1}^{n} w_i^{(k)} |y_i - \mathbf{X}_i\beta|^2 = (\mathbf{X}^\intercal \mathbf{W}^{(k)} \mathbf{X})^{-1} (\mathbf{X}^\intercal \mathbf{W}^{(k)} \vec{y})$ and $\mathbf{W}_{i,i}^{(k)} = |y_i - \mathbf{X}_i\beta^{(k)}|^{p-2}$. Also, the weighted least-squares error is re-calculated as: $e = \sum_{i=1}^{n} w_i^{(k)} (y_i - \hat{y}_i)^2$. The loop stops when the error converges. Hence, we obtain the most suitable $\vec{\beta}$ with the minimum $\vec{\varepsilon}$. Note that the developed model is designed for a single substance (nitrogen or potassium); a more sophisticated learning algorithm is needed to estimate the nutrient levels when multiple nutrients are fertilized simultaneously.

8 PERFORMANCE EVALUATION

In this section, we conduct in-lab and in-the-wild experiments to validate the sensitivity and feasibility of *IoTree*. The experiments are designed to answer the following questions: (1) *Is it possible to monitor different water and nutrient levels using the developed sensors?*; (2) *What is the accuracy of the sensing technique?*; (3) *Is it possible to monitor how the tree react to the input water and fertilizer in real-time?*; (4) *Can the system operate completely battery-free?*; (5) *Is the implemented system reliable for long-term deployment?*

8.1 Indoor Experiments

In Situ Experiments. We first validate the accuracy of monitoring (NH_3 and K_2O) nutrient levels through in-lab experiments by producing different levels of concentrations for each substance ranging from 10 g/L to 100 g/L (i.e., amount of substance over the mixed water). At each level, the measurements are repeated 100 times; 70 measurements are used to build the model (finding $\vec{\beta}$ as in Eq. 2), and 30 measurements for testing our model as in Eq. 1.

Fig. 11 shows the measured impedance corresponding to 10 levels of NH_3 and K_2O at 10 g/L resolution with sweeping frequencies ranging from 10 kHz to 100 kHz, and the 10-level impedance profiles

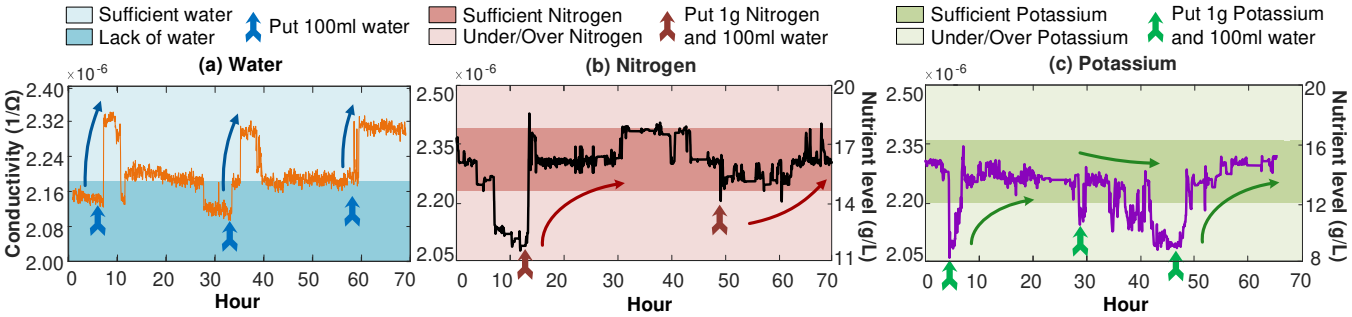

Figure 13: Water and nutrient sensing on living trees.

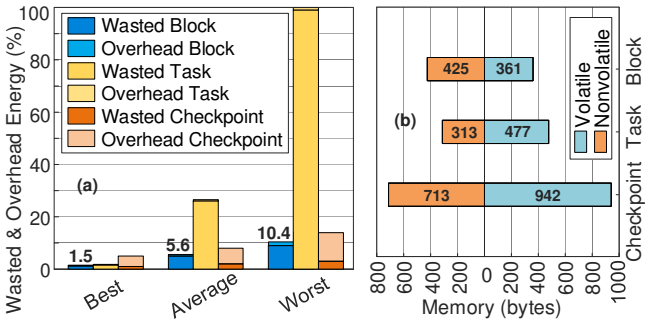

Figure 14: (a) Portion of wasted and overhead energy over consumed energy among three approaches in different cases (best-case, average-case, and worst-case scenarios). (b) Memory usage comparison among three approaches.

are also visually distinguishable. Fig. 12 illustrates that processing with the filtered data gives us higher accuracy compared to processing with the raw data: achieving the accuracy of 91.08% for 10 levels of concentration of NH_3 (from 84.34% with the non-filtered NH_3 data) and the accuracy of 90.51% for 10 levels of concentration of K_2O (from 75.33% with the non-filtered K_2O data). This confirms the reliability of the developed sensors and sensing circuit.

In Vivo Experiments. We deploy *IoTree* on Burkwood Viburnum and White Bird trees. We first put 100ml water into the soil and observe the measured impedance in the tree body over time. The measured impedance can be shown in Fig. 13 (a). As can be seen in the figure, the tree reaction to the input water can be clearly captured in real-time. In another experiment, we input 10 g/L of NH_3 into the tree's soil and observe the tree's reaction over time. Note that we did not fertilize the tree two weeks before this experiment. The tree's behaviors can be observed as in Fig. 13 (b). The conductivity increase after the tree is fertilized. This indicates that the sensor is sensitive to the nutrient levels change inside the tree body. At the 50th hour, when the conductivity reduces, we put another 10 g/L of NH_3 into the soil. The conductivity then increases to the normal range from 2.22x10^{-6} Ω^{-1} to 2.37x10^{-6} Ω^{-1} corresponding from 14 g/L to 18 g/L. We then follow the same procedure with K_2O nutrient, and similar reactions from the tree are observed as illustrated in Fig. 13 (c). The normal range from 2.20x10^{-6} Ω^{-1} to 2.32x10^{-6} Ω^{-1} corresponding from 12 g/L to 15 g/L. We use a fan to generate artificial wind for these experiments. These confirm the feasibility and reliability of the developed sensors and sensing system. In the next experiment, we present our observation in farm settings.

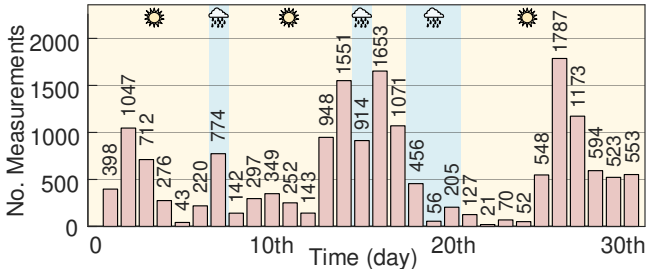

Figure 15: Average number of data transmissions of three prototypes deploying on three grapevine trees over a month.

8.2 On the Farm Experiments

Our deployment includes two components: IoTree prototypes and a base station, as shown in Fig. 17. We attach three prototypes on three grapevine trees on the field, while a base station, including a laptop interfacing with the LoRa receiver, is located at a distance of 0.8 km from the farm. Fig. 15 shows the average number of data transmissions of three prototypes we obtained every day. The number of measures varies each day depending on the weather condition. The total number of sensed measurements ranges from 21 to 1787, which is sufficient for our application. In particular, sensing data in Fig. 16 show that grapevines uptake nutrients during the day and consume nutrients at night. These measured data are matched with nutrient uptake behaviors described in literature [26]. Specifically, when grapevines uptake nutrients, the conductivity increases from 2.23x10^{-6} Ω^{-1} to 2.27x10^{-6} Ω^{-1}, corresponding from 14.4 g/L to 16 g/L for Nitrogen. According to prior work [93], the levels of Nitrogen fluctuate between normal range. We will continue to observe the measured data behaviors and report detailed results of full-season monitoring in future works.

Computing Performance. We compare the block-based approach with checkpoint and tasked-based approaches in terms of energy utilization, computation overhead, and memory utilization on the same hardware platform by performing the best-effort implementation of checkpoint approach [55] and task-based approach [57]. The source code for other relevant techniques [58, 86, 87]) are also unavailable. While Coala [88] provides open source implementation, the hardware incompatibility introduces issues in replicating the work. Regarding energy utilization, the block-based approach has similar wasted energy to the checkpoint approach but less wasted energy than the task-based approach. If the buffer energy is less than a task's required energy, the task-based approach likely wastes all

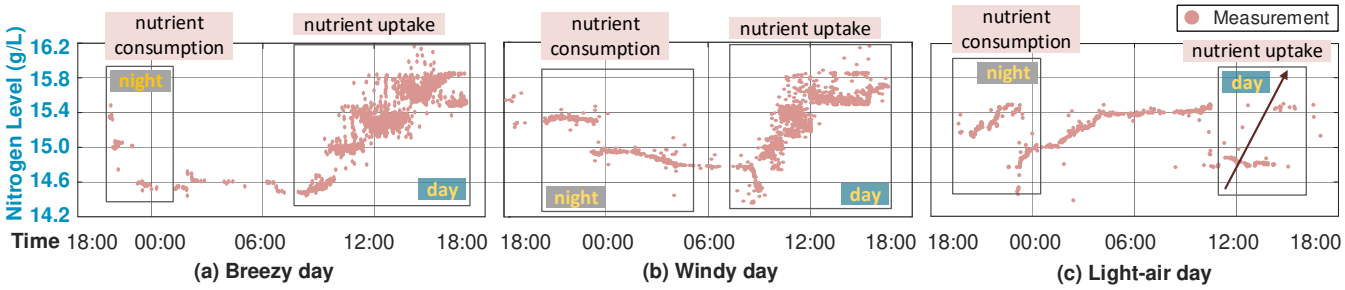

Figure 16: Sensing data in the wild deployment with different wind conditions.

power in the buffer energy, as shown in Fig. 14 (a). In the average case, our approach outperforms the task-based approach up to 4x. Our approach has similar performance to the task-based approach in terms of overhead while 5x improved compared to the checkpoint-based approach. As for memory utilization, our approach requires less memory than checkpoint up to 2x and has a similar memory footprint to the task-based approach as shown in Fig. 14 (b). We also conduct experiments with different capacitor sizes (i.e., 3 mF, 6 mF, 10 mF, and 15 mF) and let V_{max} be 4.2 V and V_{min} be 2.9 V while the required energy for three tasks is approximately 11 mJ, 0.5 mJ, and 37 mJ, respectively. The *IoTree* system executes stably for all those configurations and obtains optimal configuration after approximately ten power cycles.

System Reliability. We deploy *IoTree*'s prototype on grapevine trees farm for 30 days with an average number of measurements is 558. The prototypes perform reliably under multiple weather conditions, including windy, rainy, and sunny as illustrated in Fig. 15.

Communication Performance. We evaluate the performance and reliability of *IoTree* 's long-range communication on the field. The base station is placed at a fixed position, while *IoTree* wearable device is moved to different locations ranging from 0.3 km to 1.8 km from the base station. The system continuously sends data towards the base station. As illustrated Fig. 18, the communication works reliably up to 1.8 km with non-light-sight condition.

9 LESSONS LEARNED

In this section, we provide some of the insights and lessons learned from our experiences in designing, fabricating, deploying, and evaluating *IoTree* prototypes in the wild.

> ***Lesson 1.*** *Scheduling or sleeping is not the most optimal strategy as wind energy is unpredictable.*

Since the capacitor cannot hold the energy for an extended period due to leakage caused by the manufacturing imperfection, any operation should be performed as soon as the system accumulates enough energy by buffer size. Furthermore, the sensing system may miss important events while sleeping [85]. Additionally, the instant energy is entirely wasted if there is significant wind energy and the buffer is already full during the sleeping time.

> ***Lesson 2.*** *Choosing the capacitor size that perfectly fits the entire program is arduous in practice.*

The required energy for one cycle of the program, including sensing, data compression, and long-range communication, is impractical. This requires the system to measure the available energy at high precision. Such hardware is expensive and complex to control under intermittent power. Configuring the firmware to perfectly fit the size of capacitor leaves the programmers a difficult task when the stored energy in each power cycle might be different among each other due to the hardware imperfection. Additionally, the program may have updates and changes over deployment time; therefore, seeking out the best fit capacitor size is not a scalable method.

> ***Lesson 3.*** *Internal resistance of capacitors are crucial in understanding how voltage drop affects on the charging behavior.*

Regardless the capacitor size, we notice that internal resistance of capacitors are important in real-world deployment. Since the wind speed outdoor varies under multiple circumstances, the capacitor is not charged in most of the time if the internal resistance of the capacitor is high, eventually resulting in a voltage drop and an infinity loop of charging.

> ***Lesson 4.*** *Selecting big or small capacitors affects the responding time of the entire system since different capacitors need different times to be buffered.*

The bigger capacitors we use, the later sensing events we capture since the bigger capacitors require more time to charge. On the other hand, if we choose smaller capacitors, the system will respond quickly to the events but make a task more fine-grained, resulting in a block that does not cover at least one frequency in the sensing task. Due to the system's correlated behavior and responding time, the designing strategy in choosing the capacitor is needed; hence, we prefer selecting a capacitor whose energy is from one third to two third of the total required energy to run all tasks.

10 RELATED WORK

Nutrient Monitoring Technique. Measuring the ion quantities directly is the most accurate way to monitor nutrients inside the plants. Ion-selective electrodes are widely used in commercialized products [94, 95], and in literature [96, 97] to measure temperature, pH, sodium chloride, ammonium phosphatides, electrical conductivity, dissolved oxygen from soil or the hydroponic solution supplied to trees. However, these systems require humans to operate and are highly expensive, making it challenging to be used on an individual tree. Other approaches such as [98, 99] employ ultra-violet spectral absorption and wet chemical colorimetric reaction to measure individual ions such as ammonium, nitrate, and phosphate in the hydroponic solution with very high accuracy and fast response time. Nevertheless, these systems are costly (over $10k, [100]). Thus, it is impossible to scale these systems to monitor the nutrient of the

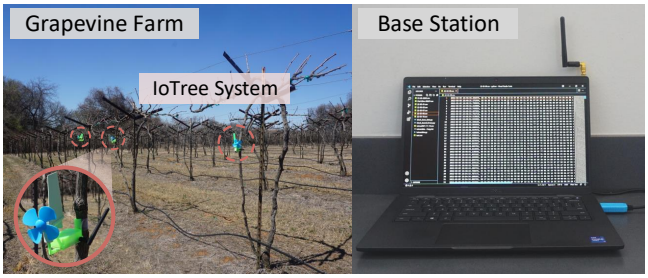

Figure 17: *IoTree*'s deployment setup.

Figure 18: *IoTree*'s long-range communication successful rate at different distances.

individual tree on a large scale. Another approach is to infer the nutrient level inside the tree base on the leaves' color [101]. However, this approach is inaccurate due to the lack of information retrieving directly from the inside of the trees and inapplicable during the defoliation period (when the tree has no leaves). *IoTree*, on the other hand, uses the novel biocompatible and implantable sensors that can be injected into the trees to measure the water and ion levels using the impedance sweeping technique.

Agricultural Sensing Platform. Automatic irrigation systems based on soil moisture [102], IoT-based systems based on sensors attached [103, 104], IoT-based greenhouse monitoring systems [59] are widely adopted in both literature and practice to improve crop yield and water efficiency effectively. However, they have several limitations, such as (1) the dependency on a power supply station and (2) invasive sensors that can harm trees. The common way to monitor nutrient deficiencies is based on the morphology and color of the leaves [105], which is impossible during the dormant stage. *Chai et al.* [106] designed the biocompatible senor for sap flow and water distribution inside the tree. Nevertheless, this sensor is not suitable for trees with large barks. A self-powered sensor for agriculture use was developed by *Lan et al.* [107]; however, this sensor is capable of Bluetooth communication only, which is impractical in farm settings since Bluetooth has a limited range of communications. To sum up, there is no previous continuous tree's health monitoring platform that integrates biocompatible sensors, battery-free design, and intermittent computing technique while being accurate, tree-wearable, reliable, and extendable to large-scale deployment.

11 DISCUSSION & FUTURE WORKS

Recycling Feasibility. As far as we know about recycling, *IoTree* can be recycled, one after another. Indeed, the *IoTree* prototype does not encounter any hardware or software issues after one-month deployment in various weather conditions, including sunny, rainy, and windy days. From our experiences on installing prototypes, we also learn that the prototype can be uninstalled from one tree and installed on another with minimal effort and without any maintenance.

Limitations. While the preliminary results are promising, the current implementation of *IoTree* has the following limitations. The sensor and sensing component is only sensitive to single substance measurement, either water or one type of fertilizer. Second, the current system only works with trees with a body size larger than 0.5 cm of radius. Third, the system is only validated with a small set of trees and one-month experiments. Last but not least, the system has only been validated at one geographical location. The tree-soil relationship is still an unexplored research problem since we reserve to explore a machine learning model to extract this relationship after substantial data are collected.

Future Works. We had only deployed *IoTree* prototypes on grapevine trees as a proof of concept as they are the only agricultural plants that are accessible within surrounding areas. However, we believe that *IoTree* would perform reliably on any kind of trees, we plan to continue optimizing the sensing system to make it more practical for long-term, full-season deployment on multiple trees at multiple geographical areas. In particular, we will develop an ion-selective and biocompatible sensing array to allow us to measure multiple nutrients simultaneously. Ion-selective membranes [108–112] and Organic Electrochemical Transistor (OECT) [113–116] are two promising approaches. We also plan to test the system with more types of trees (e.g., corn, sunflower, rice, and others). We will deploy the system on the smaller body size tree. We will look at multiple geographical locations to confirm the sensitivity and usability of the proposed solution. We will further optimize the block-based computing approach and apply it to other wind-based battery-free systems. While tree soil relationship has been explored in the past [117–120], we will explore this complex relationship by analyzing the data collected from the tree and soil simultaneously.

12 CONCLUSIONS

This paper presents *IoTree*, a battery-free wearable system with biocompatible and implantable sensors for a living tree to continuously monitor nutrient uptake. The system includes a biocompatible fiber-based sensor implanted inside the xylem of a living tree. The sensor is able to monitor 5 and 10 levels of NH_3 and K_2O nutrients with the accuracy of 91.08% and 90.51%, respectively, with in-lab experiments. With real-world experiments, the estimated nutrient and water level can be used to indicate normal, lack of water/nutrient, and over water/nutrient in real-time. The sensed data is wirelessly reported to a base station at kilometers of distance away. The entire system is powered by wind energy. To cope with the unpredictability of wind energy, we implement a block-based computing method to allow the system to fully utilize the energy with minimum memory overheads. *IoTree* is evaluated through in-lab and real-world deployment for 30 days. The experimental results show that *IoTree* is able to provide sufficient measurements every day. The system reports 558 measurements on average a day with a distance of up to 1.8 kilometers without requiring any batteries or maintenance. We include a real-time demo of the developed system [14]. The 3D model, hardware design, and software will be made publicly available to encourage adoption and reproducibility.

REFERENCES

[1] Feeding 9 Billion - National Geographic. http://www.nationalgeographic.com/foodfeatures/feeding-9-billion/, 2022.

[2] Acga Cheng, Sean Mayes, Gemedo Dalle, Sebsebe Demissew, and Festo Massawe. Diversifying crops for food and nutrition security – a case of teff. 92(1):188–198.

[3] William C. Wetzel, Heather M. Kharouba, Moria Robinson, Marcel Holyoak, and Richard Karban. Variability in plant nutrients reduces insect herbivore performance. 539(7629):425–427. Number: 7629 Publisher: Nature Publishing Group.

[4] Qingyun Zhao, Wu Xiong, Yizhang Xing, Yan Sun, Xingjun Lin, and Yunping Dong. Long-term coffee monoculture alters soil chemical properties and microbial communities. 8(1):6116. Number: 1 Publisher: Nature Publishing Group.

[5] 2019. Sustainable Agriculture Research & Education. "Rotations and Soil Organic Matter Levels.". https://www.sare.org/Learning-Center/Books/Building-Soils-for-Better-Crops-3rd-Edition/Text-Version/Crop-Rotations/Rotations-and-Soil-Organic-Matter-Levels, November 2021.

[6] Claudia Hitaj and Shellye Suttles. Trends in us agriculture's consumption and production of energy: Renewable power, shale energy, and cellulosic biomass. Technical report, 2016.

[7] University of Michigan Center for Sustainable Systems. U.S. Energy System. https://tinyurl.com/2pjvhvy7.

[8] John Miranowski et al. Energy consumption in us agriculture. *Agriculture as a Producer and Consumer of Energy*, 1:68–111, 2005.

[9] Lakesh K Sharma and Sukhwinder K Bali. A review of methods to improve nitrogen use efficiency in agriculture. *Sustainability*, 10(1):51, 2018.

[10] Bruce Horwith. A role for intercropping in modern agriculture. 35(5):286–291.

[11] K. Ann Bybee-Finley and Matthew R. Ryan. Advancing intercropping research and practices in industrialized agricultural landscapes. 8(6):80. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[12] Bhupinder S. Farmaha, Udayakumar Sekaran, and Alan J. Franzluebbers. Cover cropping and conservation tillage improve soil health in the southeastern united states. 114(1):296–316.

[13] A. Freidenreich, M. Bhat, and K. Jayachandran. Adoption and perception of cover crop implementation for tropical fruit growers. 77(2):158–171. Publisher: Soil and Water Conservation Society Section: RESEARCH SECTION.

[14] IoTree Demo. https://youtu.be/fWm2NQ6EFNE.

[15] Heyu Yin, Yunteng Cao, Benedetto Marelli, Xiangqun Zeng, Andrew J Mason, and Changyong Cao. Soil sensors and plant wearables for smart and precision agriculture. *Advanced Materials*, 33(20):2007764, 2021.

[16] Xueyan Zhang, Jianwu Zhang, Lin Li, Yuzhu Zhang, and Guocai Yang. Monitoring citrus soil moisture and nutrients using an iot based system. *Sensors*, 17(3):447, 2017.

[17] Ju Wang, Liqiong Chang, Shourya Aggarwal, Omid Abari, and Srinivasan Keshav. Soil moisture sensing with commodity rfid systems. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 273–285, 2020.

[18] Robert Chancia, Terry Bates, Justine Vanden Heuvel, and Jan van Aardt. Assessing grapevine nutrient status from unmanned aerial system (uas) hyperspectral imagery. *Remote Sensing*, 13(21):4489, 2021.

[19] Eduardo Esteves, Guilherme Locatelli, Neus Alcon Bou, and Rhuanito Soranz Ferrarezi. Sap analysis: A potential tool for monitoring plant nutrition. *Horticulturae*, 7(11):426, 2021.

[20] Abdul Salam, Mehmet C Vuran, and Suat Irmak. Di-sense: In situ real-time permittivity estimation and soil moisture sensing using wireless underground communications. *Computer Networks*, 151:31–41, 2019.

[21] Jian Ding and Ranveer Chandra. Towards low cost soil sensing using wi-fi. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.

[22] Soil Doctor. Soil Doctor Order Tests. https://www.soildoctorconsulting.com/order-tests.

[23] Daigard Ricardo Ortega Rodriguez, Hudson Wallace Pereira de Carvalho, and Mario Tomazello-Filho. Nutrient concentrations of 17-year-old pinus taeda annual tree-rings analyzed by x-ray fluorescence microanalysis. *Dendrochronologia*, 52:67–79, 2018.

[24] T Yoneyama, H Fujiwara, and JM Wilson. Variations in fractionation of carbon and nitrogen isotopes in higher plants: N metabolism and partitioning in phloem and xylem. In *Stable isotopes*, pages 99–109. Garland Science, 2020.

[25] Rubén Rellán-Álvarez, Hamdi El-Jendoubi, Gert Wohlgemuth, Anunciación Abadía, Oliver Fiehn, Javier Abadía, and Ana Álvarez-Fernández. Metabolite profile changes in xylem sap and leaf extracts of strategy i plants in response to iron deficiency and resupply. *Frontiers in plant science*, 2:66, 2011.

[26] Roman Zweifel, Frank Sterck, Sabine Braun, Nina Buchmann, Werner Eugster, Arthur Gessler, Matthias Häni, Richard L Peters, Lorenz Walthert, Micah Wilhelm, et al. Why trees grow at night. *New Phytologist*, 231(6):2174–2185, 2021.

[27] Jonas Hilty, Bertrand Muller, Florent Pantin, and Sebastian Leuzinger. Plant growth: the what, the how, and the why. *New Phytologist*, 232(1):25–41, 2021.

[28] Naiqian Zhang, Maohua Wang, and Ning Wang. Precision agriculture—a worldwide overview. *Computers and electronics in agriculture*, 36(2-3):113–132, 2002.

[29] María T Domínguez, Teodoro Marañón, José M Murillo, Rainer Schulin, and Brett H Robinson. Nutritional status of mediterranean trees growing in a contaminated and remediated area. *Water, Air, and Soil Pollution*, 205(1):305–321, 2010.

[30] Keith Skamp, Eddie Boyes, and Martin Stanisstreet. Beliefs and willingness to act about global warming: Where to focus science pedagogy? 97(2):191–217, 2013.

[31] Francisco M. Padilla, Michela Farneselli, Giorgio Gianquinto, Francesco Tei, and Rodney B. Thompson. Monitoring nitrogen status of vegetable crops and soils for optimal nitrogen management. 241:106356.

[32] T. M. Blackmer and J. S. Schepers. Techniques for monitoring crop nitrogen status in corn. 25(9):1791–1800, 1994.

[33] G Lawlor, DW Lawlor, Hans Mohr, and Peter Schopfer. *Plant physiology*. Springer Science & Business Media, 2012.

[34] Yun Shi, Zhen Wang, Xianfeng Wang, and Shanwen Zhang. Internet of things application to monitoring plant disease and insect pests. In *International conference on applied science and engineering innovation (ASEI 2015)*, pages 31–34, 2015.

[35] Nicholas C Coops, Piotr Tompalski, Tristan RH Goodbody, Alexis Achim, and Christopher Mulverhill. Framework for near real-time forest inventory using multi source remote sensing data. *Forestry: An International Journal of Forest Research*, 2022.

[36] United States Department of Agriculture. *Land Areas of the National Forest System*, volume 1 of *Fundamental Algorithms*. USDA Forest Service, 1st edition, 2015. (book).

[37] F. M. Del Amor and L. F. M. Marcelis. Regulation of nutrient uptake, water uptake and growth under calcium starvation and recovery. 78(3):343–349.

[38] Clain Jones, K Olson-Rutz, and C Dinkins. Nutrient uptake timing by crops. *Montana, USA: Montana State University*, 2011.

[39] Paula Ragel, Natalia Raddatz, Eduardo O. Leidi, Francisco J. Quintero, and José M. Pardo. Regulation of k+ nutrition in plants. 10:281.

[40] A. Gojon, G. Krouk, F. Perrine-Walker, and E. Laugier. Nitrate transceptor(s) in plants. 62(7):2299–2308.

[41] The Mosaic Company. *Nitrogen in Plants*, September 2021. https://www.cropnutrition.com/nutrient-management/nitrogen.

[42] Fernanda Campos Mastrotti Pereira, Reuben Tayengwa, Pedro Luis Da Costa Aguiar Alves, and Wendy Ann Peer. Phosphate status affects phosphate transporter expression and glyphosate uptake and transport in grand eucalyptus (eucalyptus grandis). *Weed Science*, 67(1):29–40, 2019.

[43] C. Day, S. Søpstad, H. Ma, C. Jiang, A. Nathan, S. R. Elliott, F. E. Karet Frankl, and T. Hutter. Impedance-based sensor for potassium ions. *Analytica Chimica Acta*, 1034:39–45, November 2018.

[44] Seyed Alireza Ghaffari, William-O Caron, Mathilde Loubier, Charles-O Normandeau, Jeff Viens, Mohammed S Lamhamedi, Benoit Gosselin, and Younes Messaddeq. Electrochemical impedance sensors for monitoring trace amounts of no3 in selected growing media. *Sensors*, 15(7):17715–17727, 2015.

[45] Rafael F Muñoz-Huerta, Antonio de J Ortiz-Melendez, Ramon G Guevara-Gonzalez, Irineo Torres-Pacheco, Gilberto Herrera-Ruiz, Luis M Contreras-Medina, Juan Prado-Olivarez, and Rosalia V Ocampo-Velazquez. An analysis of electrical impedance measurements applied for plant n status estimation in lettuce (lactuca sativa). *Sensors*, 14(7):11492–11503, 2014.

[46] Li Wang, Jishi Zhao, Xiangming He, Jian Gao, Jianjun Li, Chunrong Wan, and Changyin Jiang. Electrochemical impedance spectroscopy (eis) study of lini1/3co1/3mn1/3o2 for li-ion batteries. *Int. J. Electrochem. Sci*, 7(1):345–353, 2012.

[47] Javed Ahmad, SH Bukhari, M Qadeer Awan, M Ehsan Mazhar, and AR Makhdoom. Dielectric and impedance spectroscopy of k+-doped mgal2o4 nanoparticles. *International Journal of Modern Physics B*, 32(15):1850189, 2018.

[48] Li Meiqing, Li Jinyang, Mao Hanping, and Wu Yanyou. Diagnosis and detection of phosphorus nutrition level for solanum lycopersicum based on electrical impedance spectroscopy. *Biosystems Engineering*, 143:108–118, 2016.

[49] Qiyi Chen, Joey Dacula Mangadlao, Jaqueline Wallat, Al De Leon, Jonathan K Pokorski, and Rigoberto C Advincula. 3d printing biocompatible polyurethane/poly (lactic acid)/graphene oxide nanocomposites: anisotropic properties. *ACS applied materials & interfaces*, 9(4):4015–4023, 2017.

[50] Sangamesh Kumbar, Cato Laurencin, and Meng Deng. *Natural and synthetic biomedical polymers*. Newnes, 2014.

[51] Dong Ma, Guohao Lan, Mahbub Hassan, Wen Hu, Mushfika B Upama, Ashraf Uddin, and Moustafa Youssef. Solargest: Ubiquitous and battery-free gesture recognition using solar cells. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2019.

[52] Jasper De Winkel, Vito Kortbeek, Josiah Hester, and Przemysław Pawełczak. Battery-free game boy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–34, 2020.

[53] Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua R Smith. Battery-free cellphone. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–20, 2017.

[54] USDA. Farms and land in farms 2019 summary. https://www.nass.usda.gov/Publications/Todays_Reports/reports/fnlo0220.pdf.

[55] Brandon Lucia and Benjamin Ransford. A simpler, safer programming and execution model for intermittent systems. *ACM SIGPLAN Notices*, 50(6):575–585, June 2015.

[56] Jasper de Winkel, Vito Kortbeek, Josiah Hester, and Przemysław Pawełczak. Battery-Free Game Boy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):111:1–111:34, September 2020.

[57] Alexei Colin and Brandon Lucia. Chain: tasks and channels for reliable intermittent programs. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOPSLA 2016, pages 514–530, New York, NY, USA, October 2016. Association for Computing Machinery.

[58] Kiwan Maeng, Alexei Colin, and Brandon Lucia. Alpaca: intermittent execution without checkpoints. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA):96:1–96:30, October 2017.

[59] Josiah Hester and Jacob Sorber. Flicker: Rapid prototyping for the batteryless internet-of-things. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, pages 1–13, 2017.

[60] Kasım Sinan Yıldırım, Amjad Yousef Majid, Dimitris Patoukas, Koen Schaper, Przemyslaw Pawelczak, and Josiah Hester. Ink: Reactive kernel for tiny batteryless sensors. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 41–53, 2018.

[61] BioLogic. VSP Potentiostat. https://tinyurl.com/4dzrjvmw.

[62] Van Long Huynh, Tran Quang Trung, Montri Meeseepong, Han-Byeol Lee, Trong Danh Nguyen, and Nae-Eung Lee. Hollow microfibers of elastomeric nanocomposites for fully stretchable and highly sensitive microfluidic immunobiosensor patch. *Advanced Functional Materials*, 30(46):2004684, 2020.

[63] Shanshan Yao, Ping Ren, Runqiao Song, Yuxuan Liu, Qijin Huang, Jingyan Dong, Brendan T O'Connor, and Yong Zhu. Nanomaterial-enabled flexible and stretchable sensing systems: processing, integration, and applications. *Advanced Materials*, 32(15):1902343, 2020.

[64] James I. Wright. Using polyurethanes in medical applications, March 2006.

[65] Michael J Wiggins, Matt MacEwan, James M Anderson, and Anne Hiltner. Effect of soft-segment chemistry on polyurethane biostability during in vitro fatigue loading. *Journal of Biomedical Materials Research Part A*, 68(4):668–683, 2004.

[66] Analog. AD5933. https://www.analog.com/en/products/ad5933.html.

[67] Gran Badshah, Siau-Chuin Liew, Jasni Mohd Zain, and Mushtaq Ali. Watermark compression in medical image watermarking using lempel-ziv-welch (lzw) lossless compression technique. *Journal of digital imaging*, 29(2):216–225, 2016.

[68] Esra Satir and Hakan Isik. A huffman compression based text steganography method. *Multimedia tools and applications*, 70(3):2085–2110, 2014.

[69] Rapeepat Ratasuk, Benny Vejlgaard, Nitin Mangalvedhe, and Amitava Ghosh. Nb-iot system for m2m communication. In *2016 IEEE wireless communications and networking conference*, pages 1–5. IEEE, 2016.

[70] Rapeepat Ratasuk, Nitin Mangalvedhe, Amitava Ghosh, and Benny Vejlgaard. Narrowband lte-m system for m2m communication. In *2014 IEEE 80th vehicular technology conference (VTC2014-Fall)*, pages 1–5. IEEE, 2014.

[71] Evgeny Khorov, Ilya Levitsky, and Ian F Akyildiz. Current status and directions of ieee 802.11 be, the future wi-fi 7. *IEEE Access*, 8:88664–88688, 2020.

[72] Nguyen Van Huynh, Dinh Thai Hoang, Xiao Lu, Dusit Niyato, Ping Wang, and Dong In Kim. Ambient backscatter communications: A contemporary survey. *IEEE Communications surveys & tutorials*, 20(4):2889–2922, 2018.

[73] Vamsi Talla, Mehrdad Hessar, Bryce Kellogg, Ali Najafi, Joshua R Smith, and Shyamnath Gollakota. Lora backscatter: Enabling the vision of ubiquitous connectivity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–24, 2017.

[74] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A survey of lorawan for iot: From technology to application. *Sensors*, 18(11):3995, 2018.

[75] Vamsi Talla, Joshua Smith, and Shyamnath Gollakota. Advances and open problems in backscatter networking. *GetMobile: Mobile Computing and Communications*, 24(4):32–38, 2021.

[76] Joe Decuir et al. Bluetooth 4.0: low energy. *Cambridge, UK: Cambridge Silicon Radio SR plc*, 16, 2010.

[77] Mehrdad Hessar, Ali Najafi, Vikram Iyer, and Shyamnath Gollakota. Tinysdr: Low-power {SDR} platform for over-the-air programmable iot testbeds. In *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, pages 1031–1046, 2020.

[78] Semtech. SX1276. https://www.semtech.com/products/wireless-rf/lora-core/sx1276.

[79] Vijay Raghunathan, Aman Kansal, Jason Hsu, Jonathan Friedman, and Mani Srivastava. Design considerations for solar energy harvesting wireless embedded systems. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 457–462. IEEE, 2005.

[80] Santiago Orrego, Kourosh Shoele, Andre Ruas, Kyle Doran, Brett Caggiano, Rajat Mittal, and Sung Hoon Kang. Harvesting ambient wind energy with an inverted piezoelectric flag. *Applied energy*, 194:212–222, 2017.

[81] Seok Woo Lee, Yuan Yang, Hyun-Wook Lee, Hadi Ghasemi, Daniel Kraemer, Gang Chen, and Yi Cui. An electrochemical system for efficiently harvesting low-grade heat energy. *Nature communications*, 5(1):1–6, 2014.

[82] Yen Kheng Tan and Sanjib Kumar Panda. Optimized wind energy harvesting system using resistance emulator and active rectifier for wireless sensor nodes. *IEEE transactions on power electronics*, 26(1):38–50, 2010.

[83] Jasper de Winkel, Carlo Delle Donne, Kasim Sinan Yildirim, Przemysław Pawełczak, and Josiah Hester. Reliable timekeeping for intermittent computing. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 53–67, 2020.

[84] Brandon Lucia, Vignesh Balaji, Alexei Colin, Kiwan Maeng, and Emily Ruppel. Intermittent computing: Challenges and opportunities. In *2nd Summit on Advances in Programming Languages (SNAPL 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[85] Mikhail Afanasov, Naveed Anwar Bhatti, Dennis Campagna, Giacomo Caslini, Fabio Massimo Centonze, Koustabh Dolui, Andrea Maioli, Erica Barone, Muhammad Hamad Alizai, Junaid Haroon Siddiqui, et al. Battery-less zero-maintenance embedded sensing at the mithræum of circus maximus. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 368–381, 2020.

[86] Naveed Anwar Bhatti and Luca Mottola. Harvos: Efficient code instrumentation for transiently-powered embedded sensing. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 209–220. IEEE, 2017.

[87] Alexei Colin and Brandon Lucia. Termination checking and task decomposition for task-based intermittent programs. In *Proceedings of the 27th International Conference on Compiler Construction*, pages 116–127, 2018.

[88] Amjad Yousef Majid, Carlo Delle Donne, Kiwan Maeng, Alexei Colin, Kasim Sinan Yildirim, Brandon Lucia, and Przemysław Pawełczak. Dynamic task-based intermittent execution for energy-harvesting devices. *ACM Transactions on Sensor Networks (TOSN)*, 16(1):1–24, 2020.

[89] Tom M Apostol. *Introduction to analytic number theory*. Springer Science & Business Media, 1998.

[90] Albert Ruiz-Vargas, Antoni Ivorra, and John William Arkwright. Design, construction and validation of an electrical impedance probe with contact force and temperature sensors suitable for in-vivo measurements. *Scientific reports*, 8(1):1–11, 2018.

[91] Gad Miller, Karen Schlauch, Rachel Tam, Diego Cortes, Miguel A Torres, Vladimir Shulaev, Jeffery L Dangl, and Ron Mittler. The plant nadph oxidase rbohd mediates rapid systemic signaling in response to diverse stimuli. *Science signaling*, 2(84):ra45–ra45, 2009.

[92] Ye Shi, Jun Zhang, Andrea M Bruck, Yiman Zhang, Jing Li, Eric A Stach, Kenneth J Takeuchi, Amy C Marschilok, Esther S Takeuchi, and Guihua Yu. A tunable 3d nanostructured conductive gel framework electrode for high-performance lithium ion batteries. *Advanced Materials*, 29(22):1603922, 2017.

[93] Kim D. Coder. Nitrogen & trees. *University of Georgia*, 2012.

[94] Nutradip. Growboss nutrient monitoring system. https://tinyurl.com/dxkhnudr.

[95] Hanna. Hi-9811-5n ph/ec/tds/°c portable meter. https://tinyurl.com/dvdnewem.

[96] Woo-Jae Cho, Hak-Jin Kim, Dae-Hyun Jung, Dong-Wook Kim, Tae In Ahn, and Jung-Eek Son. On-site ion monitoring system for precision hydroponic nutrient management. *Computers and electronics in agriculture*, 146:51–58, 2018.

[97] Dae Hyun Jung, Hak-Jin Kim, Gyeong Lee Choi, Tae-In Ahn, Jeong-Ek Son, and Kenneth A Sudduth. Automated lettuce nutrient solution management using an array of ion-selective electrodes. *Transactions of the ASABE*, 58(5):1309–1319, 2015.

[98] Kenneth S Johnson and Luke J Coletti. In situ ultraviolet spectrophotometry for high resolution and long-term monitoring of nitrate, bromide and bisulfide in the ocean. *Deep Sea Research Part I: Oceanographic Research Papers*, 49(7):1291–1305, 2002.

[99] Miles S Finch, David J Hydes, Charles H Clayson, Bernhard Weigl, John Dakin, and Pat Gwilliam. A low power ultra violet spectrophotometer for measurement of nitrate in seawater: introduction, calibration and initial sea trials. *Analytica Chimica Acta*, 377(2-3):167–177, 1998.

[100] Brian A Pellerin, Beth A Stauffer, Dwane A Young, Daniel J Sullivan, Suzanne B Bricker, Mark R Walbridge, Gerard A Clyde Jr, and Denice M Shaw. Emerging tools for continuous nutrient monitoring networks: Sensors advancing science and water resources protection. *JAWRA Journal of the American Water Resources Association*, 52(4):993–1008, 2016.

[101] Marcilene Machado dos Santos Sarah, Renato de Mello Prado, Gelza Carliane Marques Teixeira, Jonas Pereira de Souza Júnior, Robson Luis Silva de Medeiros, and Rafael Ferreira Barreto. Silicon supplied via roots or leaves

relieves potassium deficiency in maize plants. *Silicon*, 14(3):773–782, 2022.
[102] Sandra Millán, Jaume Casadesús, Carlos Campillo, María José Moñino, and Maria Henar Prieto. Using soil moisture sensors for automated irrigation scheduling in a plum crop. *Water*, 11(10):2061, 2019.
[103] Victor Matasov, Luca Belelli Marchesini, Alexey Yaroslavtsev, Giovanna Sala, Olga Fareeva, Ivan Seregin, Simona Castaldi, Viacheslav Vasenev, and Riccardo Valentini. Iot monitoring of urban tree ecosystem services: Possibilities and challenges. *Forests*, 11(7):775, 2020.
[104] Antonio Oliveira-Jr, Carlos Resende, André Pereira, Pedro Madureira, João Gonçalves, Ruben Moutinho, Filipe Soares, and Waldir Moreira. Iot sensing platform as a driver for digital farming in rural africa. *Sensors*, 20(12):3511, 2020.
[105] Yuanyuan Sun, Cheng Tong, Shan He, Ke Wang, and Lisu Chen. Identification of nitrogen, phosphorus, and potassium deficiencies based on temporal dynamics of leaf morphology and color. *Sustainability*, 10(3):762, 2018.
[106] Yangfan Chai, Chuyi Chen, Xuan Luo, Shijie Zhan, Jongmin Kim, Jikui Luo, Xiaozhi Wang, Zhongyuan Hu, Yibin Ying, and Xiangjiang Liu. Cohabiting plant-wearable sensor in situ monitors water transport in plant. *Advanced Science*, 8(10):2003642, 2021.
[107] Lingyi Lan, Jiaqing Xiong, Dace Gao, Yi Li, Jian Chen, Jian Lv, Jianfeng Ping, Yibin Ying, and Pooi See Lee. Breathable nanogenerators for an on-plant self-powered sustainable agriculture system. *ACS nano*, 15(3):5307–5315, 2021.
[108] Hak-Jin Kim, John W Hummel, and Stuart J Birrell. Evaluation of nitrate and potassium ion-selective membranes for soil macronutrient sensing. *Transactions of the ASABE*, 49(3):597–606, 2006.
[109] Mallika Bariya, Hnin Yin Yin Nyein, and Ali Javey. Wearable sweat sensors. *Nature Electronics*, 1(3):160–171, 2018.
[110] Huixin Liu, Zhen Gu, Qing Zhao, Shuai Li, Xi Ding, Xinxin Xiao, and Guangli Xiu. Printed circuit board integrated wearable ion-selective electrode with potential treatment for highly repeatable sweat monitoring. *Sensors and Actuators B: Chemical*, 355:131102, 2022.
[111] Yi Heng Cheong, Liya Ge, and Grzegorz Lisak. Highly reproducible solid contact ion selective electrodes: emerging opportunities for potentiometry–a review. *Analytica Chimica Acta*, 1162:338304, 2021.
[112] Yoshiki Soda, Daniel Citterio, and Eric Bakker. Equipment-free detection of k+ on microfluidic paper-based analytical devices based on exhaustive replacement with ionic dye in ion-selective capillary sensors. *ACS sensors*, 4(3):670–677, 2019.
[113] Luke J Currano, F Connor Sage, Matthew Hagedon, Leslie Hamilton, Julia Patrone, and Konstantinos Gerasopoulos. Wearable sensor system for detection of lactate in sweat. *Scientific reports*, 8(1):1–11, 2018.
[114] Jonathan Rivnay, Sahika Inal, Alberto Salleo, Róisín M Owens, Magnus Berggren, and George G Malliaras. Organic electrochemical transistors. *Nature Reviews Materials*, 3(2):1–14, 2018.
[115] Xenofon Strakosas, Manuelle Bongo, and Róisín M Owens. The organic electrochemical transistor for biological applications. *Journal of Applied Polymer Science*, 132(15), 2015.
[116] Enming Song, Jinghua Li, Sang Min Won, Wubin Bai, and John A Rogers. Materials for flexible bioelectronic systems as chronic neural interfaces. *Nature materials*, 19(6):590–603, 2020.
[117] P. R. Ryan, E. Delhaize, and D. L. Jones. FUNCTION AND MECHANISM OF ORGANIC ANION EXUDATION FROM PLANT ROOTS. 52:527–560.
[118] J. H. Xia and P. H. Saglio. Lactic acid efflux as a mechanism of hypoxic acclimation of maize root tips to anoxia. 100(1):40–46.
[119] Abraham J. K. Koo and Gregg A. Howe. The wound hormone jasmonate. 70(13):1571–1580.
[120] Astrid Ratzinger, Nadine Riediger, Andreas von Tiedemann, and Petr Karlovsky. Salicylic acid and salicylic acid glucoside in xylem sap of brassica napus infected with verticillium longisporum. 122(5):571–579.