

Review

Not peer-reviewed version

An Overview of Software Sensor Applications in Biosystem Monitoring and Control

[Nasem Badreldin](#) , [Xiaodong Cheng](#) , [Ali Youssef](#) *

Posted Date: 23 September 2024

doi: 10.20944/preprints202409.1722.v1

Keywords: Software sensors; Biosystems; Monitoring; Control; Machine-Learning; Digital Agriculture



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Review

An Overview of Software Sensor Applications in Biosystem Monitoring and Control

Nasem Badreldin ¹, Xiaodong Cheng ² and Ali Youssef ^{3,*}

¹ Department of Soil Science, University of Manitoba, 13 Freedman Crescent, Winnipeg, MB R3T 2N2, Canada

² Mathematical and Statistical Methods Group (Biometris), Department of Plant Science, Wageningen University & Research, 6700 AA Wageningen, The Netherlands

³ Adaptation Physiology Group, Wageningen University & Research, P.O. Box 338, 6700 AH Wageningen, The Netherlands

* Correspondence: ali.youssef@wur.nl

Abstract: This review article explores the innovative application of software sensors for monitoring and controlling biosystems, emphasizing their advantage in managing the complexities of various biological processes. Biosystems—from cellular interactions to ecological dynamics—are characterized by intrinsic nonlinearity, temporal variability, and uncertainty, posing significant challenges for traditional monitoring approaches. A critical challenge highlighted is that what is typically measurable may not align with what needs to be monitored. Software sensors bridge this gap by combining hardware sensor data with advanced computational modelling techniques to indirectly infer hard-to-measure target variables, such as stress level, animal and human health indicators, and chemical soil properties. The article outlines advancements in sensor technologies and their integration into model-based monitoring and control systems, leveraging the capabilities of Internet of Things (IoT) devices, wearables, remote sensing, and smart sensors. It provides an overview of common methodologies for designing software sensors, focusing on the modelling process. The discussion contrasts hypothetico-deductive (mechanistic) models with inductive (data-driven) models, illustrating the trade-offs between model accuracy and interpretability. Specific case studies are presented, showcasing software sensor applications such as the use of Kalman filter in greenhouse control, remote detection of soil organic matter, and sound recognition algorithms for early detection of respiratory infections in animals. Key challenges in designing software sensors, including the complexity of biological systems, inherent temporal and individual variabilities, and the trade-offs between model simplicity and predictive performance, are also discussed. This article positions software sensor as remarkable tool for advancing biosystem management, driving forward the potential for sustainable practices in various sectors such as agriculture, healthcare, and environmental monitoring.

Keywords: Software sensors; Biosystems; Monitoring; Control; Machine-Learning; Digital Agriculture

1. Introduction

In general, “Biosystems” refers to complex living systems that involve biological components (e.g., cells, tissues, organs, whole organisms, and ecosystems) interacting with each other and their surrounding (micro-)environment [1]. In other words, as per Ruth and Hannon [2], biosystems are systems in which the crucial part of their underlying processes is a living organism, serving as the protagonist that shapes and steers the processes defining these systems. Such systems are complex assemblages of interacting physical, chemical, and biological processes. A significant number of these processes have inherent nonlinearity, and there is substantial uncertainty about their interconnections and nature [3,4]. Additionally, these processes can span the spectrum of scale and complexity, from the microscopic intercellular biological processes like photosynthesis and cellular respiration within a single cell to the macroscopic intricate relationships between living organisms and their environment, such as predator-prey interactions and soil microbiome.

Comprehensive study, understanding, and control of biosystems necessitate watchful monitoring and close observation to unravel their complexity and harness their potential. Therefore, biosystem monitoring plays a decisive role in ensuring the optimal operation of contemporary industrial operations, encompassing agricultural and biological processes.

Biosystem monitoring involves systematically collecting and analysing data related to the living organisms involved and their underlying processes. This includes monitoring the living organism's physiological and behavioural responses. The ultimate goal of biosystem monitoring is to gain insights into the underlying mechanisms, identify potential issues, and optimize conditions for enhanced efficiency and productivity.

In recent decades, there has been a remarkable surge in the application of computational methods to monitor biosystems. This progress has been powered by revolutionary advancements in sensor and sensing technologies, providing us unprecedented access to biological and environmental measurements. Concurrently, significant steps have been made in computational methods and techniques for analysing and interpreting this wealth of biological data.

Monitoring and controlling biosystems variables, such as biomass [5], product concentrations, metabolic rate, and energy expenditure [6], soil organic matter and moisture contents [7–10], pest outbreak control [11], nutrient-plant management and greenhouse controlling [12–15], in real-time or online poses a real challenge. The difficulty stems from issues related to physical accessibility, making certain measurements too difficult. Moreover, the absence of cost-effective and dependable sensors renders some measurements too expensive to undertake. One way to surmount this challenge is to depend on some auxiliary measurable [16,17] variables to indirectly infer the unmeasurable or inaccessible process variables instead of the direct physical measurement of these variables [16]. Such a combination of these auxiliary measured variables (using hardware sensors) and estimation algorithm (software) is termed “software sensor,” soft sensor, inferential sensor, or virtual sensor [16–19].

This article represents the software sensor concept as a powerful instrument for monitoring and controlling/managing biosystems. The paper also explores the evolution of biosystem contexts, software sensors, and biosystems' functions and applications. The discussion will highlight potential challenges and limitations, such as technical challenges and calibration against physical measurements.

2. Model-Based Monitoring and Controlling of Biosystems Concepts

The fourth industrial revolution, also known as Industry 4.0, has brought significant advancements in sensor technologies (e.g., Internet-of-things (IoT) and smart sensors), which led to a substantial improvement in monitoring and control of biosystems [20]. That enables a more accurate, faster, smaller/compact, and notably cheaper means for data gathering and logging biological and environmental data. The advancement of downsizing has facilitated the creation of micro-sensors, which can be integrated into various environments. The utilization of enhanced materials has facilitated the development of sensors with increased levels of sensitivity and selectivity, allowing them to accurately detect even the most subtle changes in biological signals [20,21]. Moreover, with these cost-effective sensing advancements, deploying multiple sensors in the field and integrating multimodal data to make informed decisions becomes possible. For example, in agriculture, different sensors can gather data about soil conditions, weather, and crop health to optimize crop yields [22–24]. In many applications, sensor technologies are used with control systems, which provide continuous feedback of process variables to automate biosystem processes. For instance, in industrial biotechnology, advanced sensors can provide continuous feedback on the bioreactor conditions to the control system, optimizing the production of pharmaceuticals and other bioproducts [25].

With that said, in the context of monitoring biosystems, it is often noticed that what we can typically quantify using conventional hardware sensors may not align with what we truly desire to monitor and study. While hardware sensors can provide invaluable data on various physical, chemical, and physiological parameters, hereafter known as “measured variables,” such as

temperature, pH level, heart rate, etc., often capture a very limited snapshot of the holistic reality of the biosystem. We truly aspire to observe and monitor the nuanced behaviours and the emergent status of these systems, hereafter called “target variables” (e.g., performance, stress, health, wellbeing, ...), which transcend the capacity of traditional hardware sensors. Thus, bridging this gap between the measured variables (field data) and the target variable(s), remains a remarkable challenge to monitor and control many biosystems [16].

Furthermore, in many cases of biosystem applications, it is often that several vital biosystem-related variables are challenging to measure directly using hardware sensors due to their difficulty (e.g., plant biomass, chemical soil properties, plant nutrient dynamics), inaccessibility (e.g., intercellular signalling, pain and thermal perceptions, and ecological relationships in remote or extreme environments), or high cost (e.g., energy expenditure, eDNA, and proteomics).

Model-based monitoring of biosystems is a promising tool for overcoming the aforementioned challenges associated with the direct use of hardware sensors. Instead, it employs the software sensor approach, integrating mathematical models or computational algorithms with sensor data [16]. This approach can offer a more comprehensive and accurate estimation of biosystem target variables. Accordingly, this can enhance our ability to predict, control, and optimize biosystems, ultimately promoting better decision-making in fields such as medical and veterinarian diagnostics, biotechnology, and environmental and agricultural sciences. Figure 1 shows the general schema for the model-based monitoring and control (M_bMC) system [16]. This scheme comprises the biosystem component (block), where certain outputs, also referred to as bio-responses [1,26] can be directly measured using hardware sensors. The measured variables (e.g., acceleration, video/sound signals, satellite images, and soil temperature), also known as field sensor data, are integrated with the estimation algorithm component (block) to indirectly infer the target variable (e.g., infection, animal stress, growth rate, and soil organic matter), which is the variable directly related to the final objective of the monitoring system [27,28].

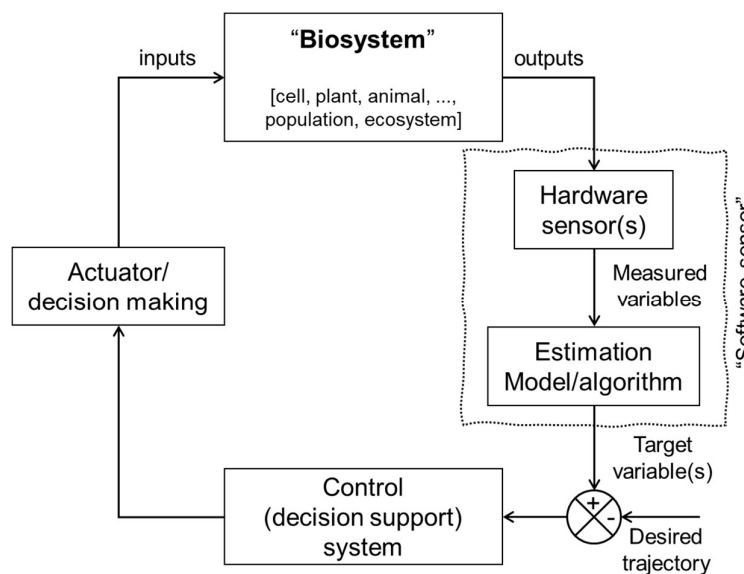


Figure 1. A general schema for model-based monitoring and control (M_bMC) of biosystems and the employment and software sensor concept.

Online monitoring of the target variable could be used to make informed decisions and optimize the output of biosystems (e.g., bioproducts). This can be achieved through the control or decision support system, which leverages the continuously updated data from the monitored variable to enhance the performance and output of the biosystems [29].

3. Software Sensors

The advent of the digital revolution has drawn attention to the need for a different type of sensor capable of performing computational inferences and estimating unready observable variables. This has given rise to the concept of software sensors, which are also known as soft, virtual, and inferential sensors. The concept of software sensors is based on the idea of inferring (estimating) essential biosystem variables (target variables) that are challenging to measure directly by fusing data acquired from hardware sensor technology with an estimation model or algorithm (i.e., software). As illustrated in Figure 1, the software sensor comprises two main components: the hardware sensor(s), responsible for generating the measured variable(s), and the software component, which encompasses an estimation algorithm for the target variable prediction.

3.1. The Emergence of Software Sensor

The concept of software sensing originated from the domain of systems and control theory [30,31]. In many control applications, we need to estimate some internal states of a dynamic system that are unmeasured or difficult to measure for control purposes. The particular estimation algorithm is known as a “state estimator” in the context of control theory, by which the future values of the system states are estimated or predicted based on the current measured states together with the dynamic model of the system [32]. Depending on how uncertainties in a process are treated, state estimation methods can be deterministic or stochastic. The former is usually referred to as “state-observers,” and the latter is called Bayesian filtering, a general statistical method for estimating an unknown probability distribution over time with sensor measurements and a mathematical model. A well-known Bayesian filtering is the Kalman filter, which is tailored for linear dynamic systems and Gaussian noises. For more detailed information on the concepts of state observers and the Kalman filter, please refer to Section 3.2.

Additionally, the principle of the software sensor aligns closely with the concept of *inferential control* proposed in the 1970s by Brosilow [33]. In the inferential control scheme (depicted in **Error! Reference source not found.**), the controlled variable (i.e., the primary process, or the target variable, y) is not measured directly but is instead estimated from secondary (inferential) measured variables. These inferential variables (z) are chosen to maintain a strong correlation with the unmeasured output variable (y) in response to changes in the process input (manipulated) variables (m). In this context, the software sensor is represented by the combination of the measured inferential variables and the estimator (inferential model), which, in this specific example [33], is formulated using the transfer functions G_p and G_s (as shown in **Error! Reference source not found.**). Here, the estimated value ($\hat{y}$) of the unmeasured output variable (y) is doing the job of the regular output as if it was measured in the feedback control loop. Consequently, the concept of the software sensor demonstrates significant potential not only in control applications but also in other fields.

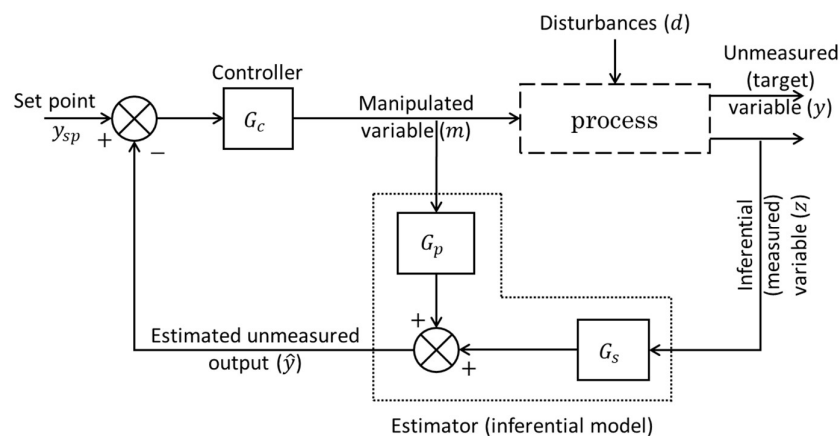


Figure 2. A schematic representation of the *inferential control* based on Brosilow's work [33].

3.2. State Observers and Kalman Filters

State observers and Kalman filter are widely recognized as the earliest and most commonly used forms of software sensors. Their foundational role in this domain stems from their robust capabilities in unmeasured state estimation and their extensive applicability across various fields. In the systems and control theory, a dynamic system is often described by an input-output state space model, which is a first-order differential equation about state variables x for continuous-time systems or a difference equation about x in the discrete-time case. The input variables u impart energy or force allowing to alter the state variables, and the measurement signals y are the system's outputs. A state observer estimates the internal states x of a dynamic system based on available measurements y and the input variables u . The diagram of the state observer is shown in Figure 3. The state observer utilizes a state space model of the system, incorporating the measurements y and the inputs u , and by comparing the predicted outputs $\hat{y}$ with the actual measurements y , the state observer provides the estimates $\hat{x}$ of the internal states and continuously refines the estimation over time. This iterative process enables the observer to provide real-time or near-real-time estimates of internal states, facilitating feedback control strategies and enhancing the overall performance of control systems in complex and dynamic environments.

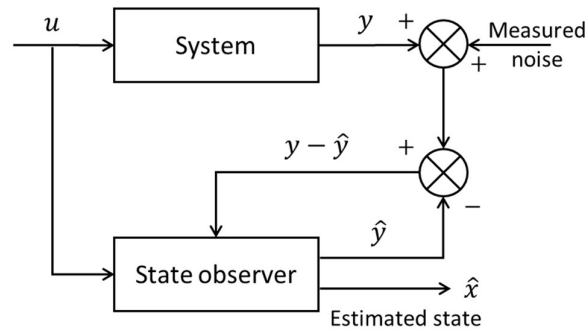


Figure 3. Schematic representation of a state observer integrated within a control system.

The following will look at the commonly employed Luenberger observer as an illustrative example. Consider a discrete-time linear state space model:

$$x(k+1) = Ax(k) + Bu(k) \quad (1)$$

where $y(k) = Cx(k)$, A , B , and C are constant matrices. The values of state variables at time $k+1$, $x(k+1)$ are predicted from current value $x(k)$ and inputs $u(k)$ using the state equation. The second equation then describes how the outputs $y(k)$ are generated. In the Luenberger observer, the difference between the predicted output $\hat{y}(k)$ will be subtracted from the measured outputs $y(k)$ and then amplified by a constant gain matrix L to correct and refine the estimates of state variables, as described by the equations below.

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L[y(k) - \hat{y}(k)], \quad (2)$$

where $\hat{y}(k) = C\hat{x}(k)$.

Provided a system is observable, namely, the so-called observability matrix

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (3)$$

is full rank, the matrix parameter L can be designed such that the observer error $e(k) := \hat{x}(k) - x(k)$ converges to zero as k goes to infinity. More precisely, the Luenberger observer error follows the dynamics $e(k+1) = (A - LC)e(k)$, and therefore, when $A - LC$ is stable (i.e., all its eigenvalues are

inside the unit circle), then the error dynamics is asymptotically stable, meaning that $e(k)$ converges to zero as k increases, for any initial error $e(0)$.

The above Luenberger observer is a type of full-order observer which aims to estimate all the state variables. On the other hand, reduced-order observers are an alternative approach for software sensor that uses system measurements to estimate only the 'hidden' states (i.e., target variables) as a subset of the state variables of a system. While reduced-order observers may be more challenging to design compared to their full-order counterparts, they offer advantages in terms of computational efficiency, simplicity, and potential for improved performance in certain applications [34–36].

The Kalman filter is named after Rudolf E. Kálmán (May 19, 1930 – July 2, 2016), who published his famous paper [37] which describes a recursive method for estimating the state of a linear dynamic system from a series of measurements with Gaussian noises. Consider the linear discrete-time system:

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (4)$$

$$y_k = Cx_{k-1} + v_k \quad (5)$$

where x_{k-1} denotes the state at time $k - 1$, and x_k , u_{k-1} , and y_{k-1} are the state, control input, and measured output at time k , respectively. Moreover, the constant matrices A , B , and C are referred to as transition matrix, input matrix, and output matrix, respectively, w_k is Gaussian process noise with covariance matrix Q , and v_k is the measurement noise with covariance matrix R . The objective of the Kalman filter is to discover the probability distribution of the state x at time k .

The standard Kalman filter is a two-step process, as shown in Figure 4. The first step is *prediction*, i.e., to compute the state prediction and the error covariance P based on the linear model without sensor measurements. This error covariance matrix P can be considered a measure of uncertainty in the estimated state. This variance comes from the process noise and propagation of the uncertain $\hat{x}_k^-$. At the start of the algorithm, the values for $\hat{x}$ and P come from their initial estimates. The second step is called *correction*, where we weigh in the sensor measurement. More precisely, the a priori estimates calculated in the prediction step are updated to find the a posteriori estimates of the state $\hat{x}_k$ and error covariance P_k . To do so, the Kalman gain K is calculated such that it minimizes this new error covariance P_k , and it determines how heavily the measurement and the a priori estimate contribute to the calculation of x_k , or simply speaking, how much the sensor measurements can be trusted. If we have sensors providing accurate measurements, meaning that the measurement noise is small, then the measurement can be trusted more and have more weight such that it contributes to the calculation of x_k , more than the a priori state estimate does. In the opposite case where we have bad sensors, the measurement noise will be large, namely a large R , then the sensors are trusted less, so in the equation, the computation of P_k , mostly comes from the a priori estimate.

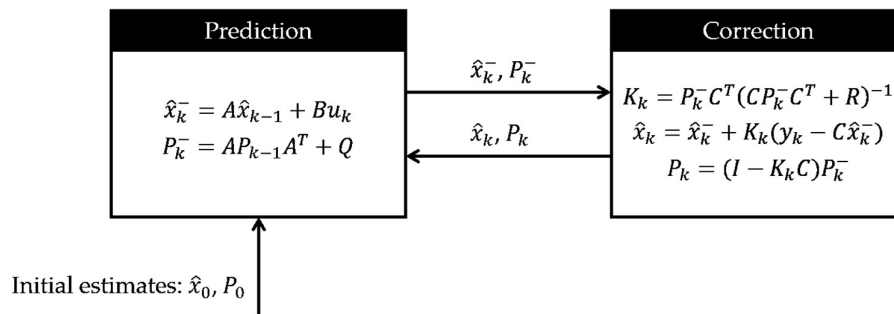


Figure 4. Flowchart illustrating the two-step process of the Kalman filter algorithm for state estimation in a linear discrete-time system.

Aside from the standard Kalman filter, which is widely used for state estimation in linear dynamic systems with Gaussian noise, there have been various developments and variations to

accommodate different types of systems and noise characteristics. Below, we list some of the notable ones:

- (1) *Extended Kalman Filter* (EKF) [38,39]: This is an extension of the Kalman filter to nonlinear systems. It linearizes the nonlinear system at each time step around the current mean and covariance estimate and then applies the standard Kalman filter equations. While widely used, the EKF has limitations, especially when dealing with highly nonlinear systems or when linearization errors are significant.
- (2) *Unscented Kalman Filter* (UKF) [40]: The UKF addresses some of the limitations of the EKF by approximating the mean and covariance through a set of carefully chosen sample points (called sigma points) rather than linearization. It captures the mean and covariance of the state distribution more accurately in nonlinear systems and is more robust to nonlinearity than the EKF.
- (3) *Ensemble Kalman Filter* (EnKF) [41,42]: The EnKF is an extension to handle large-scale systems, which uses a Monte Carlo approach with a finite number of ensemble members. Essentially, it makes use of sample covariance in place of the traditional covariance matrix.
- (4) *Particle Filter* [43] is a non-parametric filter that represents the state estimate as a set of weighted particles (similar to the ensemble members in EnKF). Unlike other Kalman filter variants, the particle filter does not rely on Gaussian assumptions, and therefore, it can deal with highly nonlinear and non-Gaussian systems. However, this flexibility comes at the cost of efficiency compared to EnKF.

3.3. Software Sensor Design

This section provides a brief overview of the common methodologies for designing software sensors, aiming to help non-expert readers understand the main components of the design process without delving into exhaustive theoretical and mathematical details. For more in-depth theoretical insights, readers are encouraged to refer to the cited bibliography.

The model, or estimation algorithm, is the cornerstone of the software sensor (refer to Figure 1), making its development and identification the most critical step of the software sensor design process. Hence, based on the modelling approach employed in the estimation algorithms, software sensors can be classified into two subtypes: the hypothetico-deductive (or white-box) approach and the inductive (or black-box) approach. In the following subsection the main differences of both modelling approaches are explained.

3.2.1. Modelling Approaches

The hypothetico-deductive (white-box) modelling approach involves defining a priori conceptual model structures based on first principles (*ab initio*) assumptions derived from established scientific paradigms [44]. This method, also known as mechanistic modelling, relies on fundamental physical laws such as thermodynamics, fluid dynamics, and heat transfer. For example, the calculation of food thermal properties for designing storage and refrigeration equipment strictly using heat transfer and thermodynamics laws, without incorporating any empirical models or fitting parameters, exemplifies a deductive modelling approach. The primary advantage of this method is its ability to provide a deep mechanistic/physical understanding of the process, leading to high accuracy if the system is well understood and the necessary parameters are available. However, this approach often requires extensive domain knowledge and detailed data, which can be a limitation in complex or poorly characterized systems.

The inductive or data-driven (black-box) modelling, on the other hand, abstains from making theoretical preconceptions at the early stages of analysis [44]. Instead, it aims to discover patterns directly from the observational data. The model structure in this data-driven approach is not pre-specified but is inferred from the data using statistical and machine learning techniques. Inductive modelling encompasses a variety of methods, including artificial neural networks (ANN), support vector machines (SVM), and more recently, advanced deep learning techniques. These methods excel in capturing complex patterns and relationships within large datasets without requiring detailed

knowledge of the underlying physical/physiological processes. The rapid growth in computational power and data availability has significantly expanded the use of inductive modelling in soft-sensing applications. However, the accuracy of these models is highly dependent on the quality and quantity of data, and they may struggle to generalize beyond the conditions represented in the training data.

The choice between deductive and inductive approaches depends on several factors, including the complexity of the system, the availability of process data, the level of understanding of the underlying process mechanisms, and the specific objective of the software sensor design.

Hypothetico-deductive reasoning has long been integral to biological research practice, underpinning foundational theories such as evolution and germ theories. However, the inherent complexity and nonlinearity of biological systems make extreme deductive (pure white-box) modelling approaches challenging [1,16]. This is often resulting in highly intricate mechanistic models characterized by numerous differential equations and parameters. For instance, metabolic processes and pathways are typically modelled using a series of differential equations that describe the time evolution of metabolites and enzyme concentrations in cellular biochemical reactions [45]. Such computational complexity can place a significant burden on resources, leading to slower convergence times. This, in turn, hampers the practical application of these models in real-time or online software sensing applications.

Conversely, the inductive approach prioritizes practical problem-solving irrespective of the method used. While an extreme inductive (data-driven) model might accurately fit the available data, it can lead to false predictions without an understanding of the underlying system and mechanisms.

Nevertheless, the inductive approach offers a more pragmatic solution for designing software sensors, especially when the primary goal is to achieve practical and timely results without needing intricate mechanistic details. Additionally, this approach leverages the power of empirical data and advanced computational techniques to develop models that can adapt and respond to real-time data with greater flexibility and efficiency. Therefore, for the sake of convenience, this article focuses exclusively on data-driven software sensors.

3.2.2. Data-Driven Software Sensors

Designing a software sensor using a data-driven approach involves several critical steps to infer the unmeasured (target) variable from measured (auxiliary) data. These steps encompass data acquisition, selection of auxiliary variables and features, modelling, model validation and testing, and implementation. In this article, we will mainly focus on the modelling step of the design process including the selection of the appropriate data-driven modelling technique. For more comprehensive information on the other essential steps, readers can consult specific literature sources such as [17,18,46–49], which provide in-depth discussions and methodologies for effective data acquisition and preparation, feature engineering, and system integration needed for software sensor design process.

The choice of data-driven modelling techniques is fundamentally driven by the nature of the target variable that needs to be estimated. When the target variable is continuous, meaning it can take any numerical value y within a range $[a, b] \in \mathbb{R}$, a regression model (Figure 5) is typically employed. Regression models are supervised learning algorithms designed to predict continuous value by identifying the relationship between measured input (auxiliary) variables and the continuous target output. For instance, predicting cell growth rate [50], animal weight [51] or biomass concentration [45], in a biosystem monitoring or control applications, would necessitate the use of regression techniques such as linear and polynomial regression [52], decision trees for regression [53], or more advanced methods such as support vector regression (SVR) [54] or neural networks (NNs) [55].

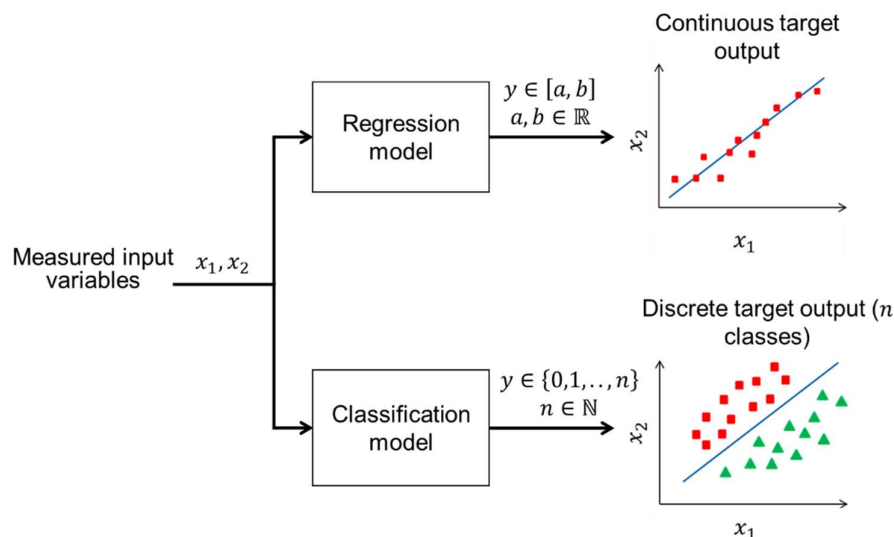


Figure 5. Regression model is used to predict continuous target variable, while classification model is required to predict discrete target variable.

On the other hand, when the target variable is discrete, meaning it can only take on a specific set of discrete values or labels $y \in \{0, 1, \dots, n\}$, $n \in \mathbb{N}$, a classification model is the appropriate choice. Classification models are used to predict categorical outcomes by learning from the patterns and relationships in the input data. For example, determining the presence or absence of a particular condition in biosystems such as infection status [56,57] or predicting different levels of biological responses such as thermal sensation [58] would require the use of classification techniques such as logistic regression [59], k-nearest neighbours [60], decision trees and random forests (RF) [61], support vector machines (SVM) [60], or NNs [62].

4. Applications and Biosystem Case Studies

Software sensors are utilized in many applications across diverse fields, spanning from manufacturing, industrial processes, environmental engineering, and energy management to medical applications, agriculture, biology, and beyond. Table 1 presents an overview of exemplary applications of the software sensing approach in different fields.

The utilization of software sensors in the fields of biology and biosystems is highly prevalent. In digital agriculture, for instance, where software sensors emerge as game-changers. Traditionally, estimating soil moisture content, for example, relied on costly methods such as wireless sensor networks and satellite imagery, which often lacked the required precision [63]. Researchers have introduced an innovative solution to these challenges — a soft sensing algorithm leveraging deep learning techniques [7]. This approach creates a virtual soil moisture sensor, bypassing the limitations associated with physical sensors and offering a more accurate and cost-effective solution. The integration of software sensors has similarly revolutionized practices in animal sciences and precision livestock farming (PLF), offering non-invasive means (Table 1) of monitoring animal behaviour, health, and welfare in real-time. For example, Youssef et al. [64,65] employed a software sensing approach to non-invasively monitor the cardiogenic signal and heart rate from moving pigs and incubated avian embryos. Moreover, advancements in computer vision algorithms coupled with camera technology enable real-time monitoring and recognition of animal activities [66–68]. Additionally, software sensors are employed in early warning systems, such as in [56,69–71], for infection and animal health problems using contactless sensors (e.g., microphones). With the advancement in software sensor technology, digital healthcare is being promoted to change people's lives [72]. Data reliability and robustness can be enhanced by constructing sensor arrays that simultaneously gather comprehensive biological parameter data from multiple body locations [72].

Table 1. Overview of software sensor applications in different fields.

	Field	Example applications
Various applications	Manufacturing and industrial processes	- <i>Process Industry</i> : [48,73]
	Environmental engineering	- <i>Wastewater management</i> : [74–76] - <i>Environmental monitoring</i> : [77] - <i>Ecology</i> : [78]
	Transportation and smart cities	- <i>Smart cities</i> : [79–82]
	Cybersecurity	- [83–88]
Biosystems applications	Biology	- <i>Neuroscience</i> : [72] - <i>Molecular and cell biology</i> : [50,89,90] - <i>Fermentation</i> : [50,83,89–91] - <i>Bioprocess monitoring and control</i> : [47,50,92–94]
	Medical and human health	- <i>Digital Healthcare</i> : [52,95–98] - <i>Pharmaceutical and drug discovery</i> : [99,100]
	Agriculture and animal health	- <i>Digital agriculture</i> : [101,102] - <i>Soil moisture estimation</i> : [7,103–105] - <i>Greenhouses monitoring and control</i> : [106,107] - <i>Precision livestock farming (PLF)</i> : [51,64–66,68,108–111]

In the following subsections, we present three case studies to illustrate the applications of data-driven software sensors in monitoring and controlling various biosystems. The first case study demonstrates the use of software sensor for real-time monitoring of animal health, specifically as an early warning tool for respiratory infections. The second case study showcases the application of software sensors in the fusion of different sensor data to estimate difficult-to-measure biosystem variables, such as soil organic matter. Lastly, the third case study explores the use of software sensors for automatic control of biosystems, exemplified by a greenhouse environment.

4.1. Early Warning System: Software Sensor for Real-Time Monitoring off Animal Respiratory Infection

In general, respiratory infections pose a significant challenge to animal health and welfare, especially in the case of intensive animal farming, particularly among calves and pigs. The increased severity and mortality rates associated with such infections show the pressing need for a robust early warning system. In current practice, reliance on manual illness detection and subsequent veterinarian intervention seems time-consuming, not early enough, and expensive [69]. Hence, there is a compelling call for an efficient automatic monitoring system to enhance the early detection of animal respiratory infections and their related outcomes, such as mortality. Such automatic monitoring systems should be robust, providing continuous and unobtrusive surveillance without disrupting the animal’s environment. Additionally, detecting the infection at an early stage requires swift observation of the clinical signs associated with the disease. Coughs are one of the primal clinical signs associated with respiratory infections such as bovine respiratory diseases (BRD) [57] The distinctive sound of coughing can serve as a key feature indicator for predicting respiratory infections in animals. The advantage of coughing sound, and bioacoustics in general, lies in its ability to be measured noninvasively and remotely using microphones, ensuring minimal disruption to the

animal’s normal behaviour [56]. Over the past two decades, different research efforts (e.g., [56,69,112–119]) have utilized sound measurements in conjunction with various algorithms to extract acoustic features indicative of respiratory infections. This approach can essentially be perceived as a software sensor, operating to indirectly predict an animal’s infection status. In this section, we will clarify the primary structure and main components (Figure 6) of the software sensor employed to predict whether an animal is infected or not based on its coughing sound. As illustrated in **Error! Reference source not found.**, the target variable in this context is a binary label, $y \in \{0,1\}$, indicating infected = 1 or not-infected = 0 animal status. The measured variable is sound signals recorded using microphones, representing the hardware sensor component, see Figure 6. Functioning as the software component, two algorithms, namely, Model I and Model II (see Figure 6), are utilized as follows:

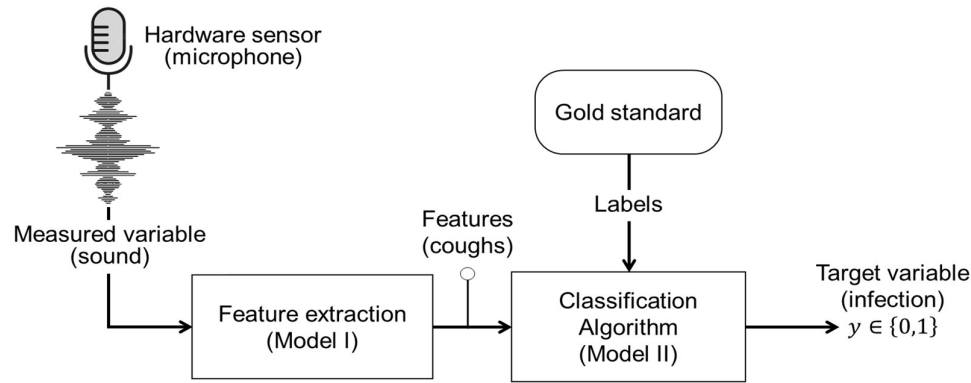


Figure 6. Schematic representation of a software sensor designed for early warning of animal respiratory infection.

4.1.1. Model I: Feature Extraction and Coughing Sound Recognition

This algorithm is responsible for extracting the key feature indicators, the cough sound, and other features from the measured variable—specifically, the sound signals—associated with the animal respiratory infection.

For coughing sound recognition, acoustic features can be manually extracted from the audio waveforms. Specifically, the Mel-frequency cepstral coefficient (MFCC) has been used for pig cough recognition and classification [70]. Moreover, other time and frequency domain features, such as power spectral density (PSD), spectral entropy, and root mean squares (RMS), are frequently used for cough sound recognition in both pigs and calves, as evidenced in studies [56,69,120]. An exemplar waveform and spectrogram visualization of extracted coughing sounds [121] from sick pigs is depicted in Figure 7.

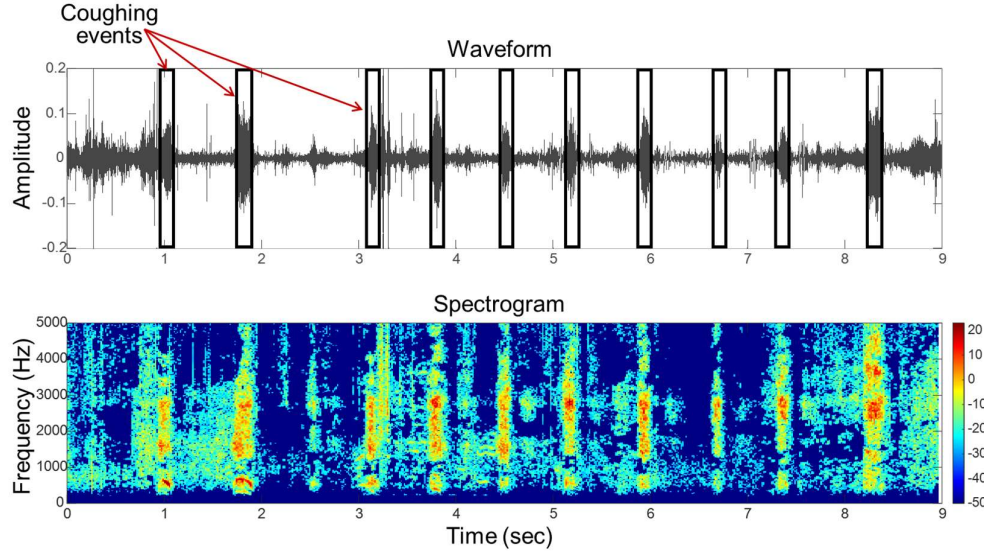


Figure 7. Exemplar waveform (upper graph) and spectrogram (lower graph) of the coughing sound acquired from a sick pig with pneumonia.

Alternative approaches utilize machine learning techniques, such as CNN, for coughing sound classification and recognition. For instance, Yin et al. [71] employed a deep CNN model (AlexNet) model in an end-to-end manner to recognize sick pig's coughs. However, this approach is computationally demanding and impractical for real-time applications. To overcome these drawbacks, researchers like Shen et al. [114] adopted a pragmatic strategy by employing shallow CNN to extract deep acoustic features. These features were then fed to a lighter classifier, such as SVM, to recognize pig's coughing sound. An extensive review of cough sound recognition and classification approaches is represented by Legua et al. [122].

4.1.2. Model II: Detection of Infected Animals

This algorithm, which is a supervised classification model in this context, learns from the labelled dataset, using the gold standard, to assign the extracted input features (e.g., number of coughing events), using Model I, to one of the target variable labels. In the context of this example, the target variable poses only two possible labels, namely infected or not-infected. Thus, the goal of the Model II is to solve the classification problem $y = \mathcal{F}(\mathbf{x})$, where $\mathcal{F}(\cdot)$ is a decision function that maps the input features $\mathbf{x}$ to the output (target) variable y .

To find the optimal parameters for $\mathcal{F}(\cdot)$, the model needs to be trained on a given labelled dataset. Consider the labelled training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with input data $\mathbf{x} \in \mathbb{R}^n$, where $n \in \mathbb{N}$ is the number training samples, and output data $y_i \in \mathbb{R}$ with class labels $y_i \in \{0, 1\}$ indicating infected (1) or not-infected (0). The classification problem presented here can be solved by a binary classifier such as SVM, NNs, random forest, logistic regression etc. Binary classifiers involve finding the parameters that minimize a certain cost or loss function. The specific form of the function $\mathcal{F}$ and the associated optimization process depends on the chosen algorithm. For example, in logistic regression, a sigmoid function is defined as follows:

$$\mathcal{F}(\mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k)}} \quad (6)$$

where $\beta_0, \beta_1, \dots, \beta_k$ are the parameters to be optimized during the training process to find a decision boundary that separates the two classes in the defined feature space $\langle x_1, \dots, x_k \rangle \in \mathbb{R}^k$.

4.2. Sensor Fusion: Software Sensor Application for Indirect Estimation of Soil Organic Matter

Assessing soil organic carbon (SOC), which is a vital indication of soil health due to its importance in carbon sequestration, fertility, and soil moisture content [123,124]. Precisely mapping SOC levels in various landscapes helps evaluate ecosystem benefits, direct sustainable land use methods, and reduce climate change impacts [125,126]. Also, advanced sensing technologies and data analytics now have a vital function in sustainable biosystem management, whether it involves optimizing crop growth conditions in high-tech greenhouses or evaluating the carbon storage potential of soils at various scales [125,127,128]. Both arenas highlight the significance of accurate, data-driven methods for protecting the health and productivity of key biological systems on the planet, regardless of their various scales and properties.

Using hyperspectral remote sensing technology is a significant advancement in digital agriculture due to its ability to capture soil surfaces with exceptional spectral precision that can reach hundreds of spectral bands (i.e., groups of a particular range of wavelengths)[129–131]. Hyperspectral imaging using the visible near-infrared (VIS-NIR) is particularly effective in detecting near-real-time fine spectral differences, as each material on the soil’s surface has a unique spectral signature akin to a fingerprint, which is often called “spectral signature”; This allows sensing capability to discriminate between different soil surface materials and improve SOC predictions [132–136]. In this case study, a software sensor is employed to estimate SOC. Such application requires following specific data management and processing frameworks such as those depicted in Figure 8.

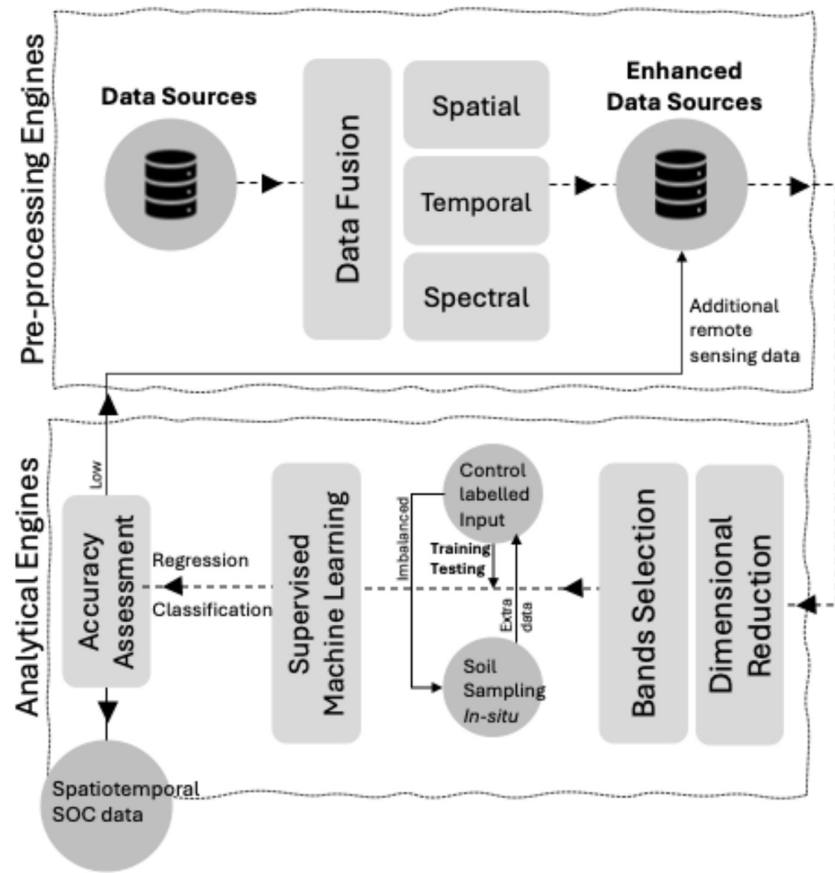


Figure 8. Software sensors sequential diagram process for SOC prediction using an integration approach of remote sensing data acquisition and management with AI modelling.

Software sensors begin their role at the data fusion stage, where hyperspectral imaging data is combined with other relevant data sources such as climate data, soil type, and historical crop yield data. Spectral features that correlate with SOC levels are extracted using methods like principal component analysis (PCA) or selecting spectral indices sensitive to SOC. This approach reduces dimensionality while preserving variance in spectral data [137,138]. PCA is a technique that converts

the initial spectral data into a collection of linearly independent variables known as principal components and given as follows:

$$PCA(X) = W^T \cdot X, \quad (8)$$

where X is the matrix of spectral data and W is the matrix of weights or coefficients. Once relevant spectral features are extracted, two main modelling approaches might be considered; linear regression can be used when there is low spatial heterogeneity between the observation [139].

$$Y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \dots + \beta_n \cdot x_n + \varepsilon, \quad (9)$$

where Y is SOC content; $X_1, X_2, \dots, X_n$ are the spectral features, $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ are the coefficients that describe the impact of each spectral feature on the SOC, and ε is the error.

For more complex relationships, nonlinear models, or machine learning techniques such as the SVM, RF, or NNs might be employed. These techniques can more effectively capture nonlinearities and interactions between spectral features [140,141].

SVM models are utilized for predicting soil organic carbon (SOC) levels, whether through spatial classification or regression [142]. The SVM algorithm works by identifying an optimal hyperplane that best separates different groups of SOC data points in a high-dimensional space, thus enabling accurate categorization and prediction of SOC levels. [143].

The SVM model for SOC classification can be represented as:

$$f(x) = w \cdot \phi(x) + b, \quad (10)$$

where x is the input feature vector, $\phi(w)$ is the feature vector transformed into a high-dimensional space, w is the weight vector, and b is the bias term. The decision function is based on the sign of $f(x)$.

RF is a machine-learning technique used for classification, regression, and other tasks. It works by creating many decision trees during the training process. A simple representation for an RF regression model is the average of all the decision trees:

$$f(x) = \frac{1}{N} \sum_{i=1}^N f_i(x), \quad (11)$$

where $f_i(x)$ is SOC prediction of the i -th decision tree, N is the number of trees in the RF.

Neural Networks, especially deep learning models, are capable of capturing complex non-linear relationships. A simple feedforward neural network for regression can be mathematically represented as a series of transformations:

$$\begin{aligned} h_1 &= \sigma(W_1 x + b_1) \\ h_2 &= \sigma(W_2 h_1 + b_2) \\ &\vdots \\ f(x) &= W_k h_{k-1} + b_k \end{aligned}, \quad (12)$$

where x is the input, h_i are the hidden layers, W_i and b_i are the weights and biases of the i -th layer, and σ is a non-linear activation function such as the sigmoid, tanh, or ReLU. For capturing non-linearities and interactions between spectral features, the convolutional neural network (CNN) can be used when (e.g., raster data) derived from hyperspectral space-borne sensors such as the Environmental Mapping and Analysis Program (EnMAP) and the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA), or a Recurrent Neural Network (RNN) if the data is sequential, which depends on multi-temporal such as the Open Soil Spectral Library (OSSL) [127,146–149].

4.3. Biosystem Control: Software Sensor Application for Automated Greenhouse Control

The modern greenhouse is a complex dynamic system that offers a controlled or partially enclosed space for plants, shielding them from external weather fluctuations. To foster an environment suitable for growing crops while minimizing energy usage, efficient control and optimization strategies have been applied to regulate greenhouse climate conditions, e.g., [13,150,151]. However, implementing these advanced control strategies is hardly achievable if we do

not have accurate information about climate conditions and crop growth status. It is reported by [152] that sensor errors can lead to a significant negative effect on crop yield and energy use in the greenhouse.

In practice, the sensors that measure the climate states, such as air temperature, humidity, and carbon dioxide level, can be affected by noises from various sources. Moreover, the real-time and non-invasive measurement of the crop growth states, such as leaf area, dry weight, and fruit content, presents an even greater challenge due to the complex nature of plant physiology. In this context, soft sensing techniques offer promising avenues for addressing these challenges. One such solution is to apply data assimilation through Kalman filtering, and some relevant results can be found in [106,153,154].

A typical implementation of data assimilation in the lettuce greenhouse is outlined in [154], where the greenhouse system has a schematic representation, as shown in Figure. 9. The microclimate within the greenhouse is characterized by three key variables: CO₂ concentration x_2 , indoor temperature x_3 and humidity x_4 . These environmental factors are subject to the influence of external weather conditions (denoted as d), such as solar radiation, external CO₂ levels, air temperature, and humidity, along with control operations (denoted as u), including CO₂ supply, ventilation, and heating. The environment states x_2 , x_3 , and x_4 , weather conditions d , and control signals dedicated sensors with inherent noises measure u . On the other hand, the dry weight of crops x_4 is hard to measure directly in practice. Therefore, the problem of interest here is to filter out the measurement noises, estimate the three environmental states of greenhouses, and infer unmeasurable dry weight using data assimilation.

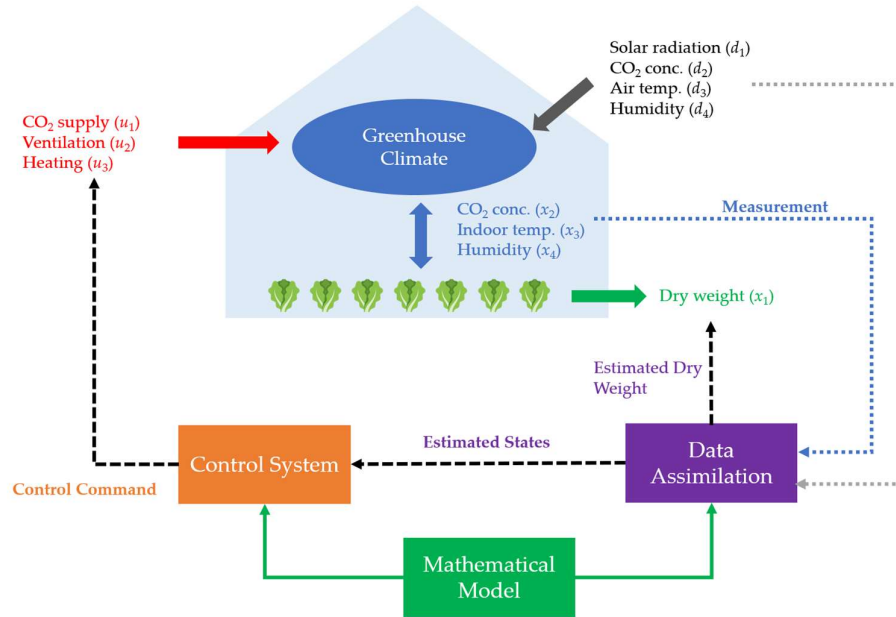


Figure 9. Schematic representation of the lettuce greenhouse model.

The data assimilation requires a dynamic model of the greenhouse system in:

$$x(k+1) = f(x(k), u(k), d(k), w(k)) \quad (13)$$

$$y(k) = g(x(k), v(k))$$

With all the variables at time step k , namely, state $x(k)$, measured output $y(k)$, controllable input $u(k)$, weather disturbance $d(k)$, process noise $w(k)$, measurement noise $v(k)$. The variable $x(k+1)$ is the prediction environmental variable at time step $k+1$. Thereby, the nonlinear function $f(\cdot)$ is a complex function describing how the environmental variables (CO₂ concentration x_2 , temperature x_3 , and humidity x_4) and the crop variable (dry weight x_1) change over time, given the

external weather condition d and control inputs u . In [154], the ensemble Kalman filter is employed for data assimilation, which roughly contains two main steps:

Step 1: Predict the ensemble forward in time using the mathematical model, which simulates the effects of the external weather conditions and the control operations on the greenhouse system.

Step 2: Update the ensemble with new sensor measurements using a formula incorporating measurement errors and the ensemble covariance. This step adjusts the ensemble to be consistent with the data and reduces the uncertainty of the state estimate. In this step, the algorithm also provides an estimate of the unmeasurable dry weight by including it in the state vector and updating it with the observations of the other variables, using the correlations between them. In [154], the ensemble Kalman filter estimates the parameters in the mathematical model by treating them as part of the ensemble.

The output of the data assimilation is the estimated state variables, which include an estimation of crop dry weight and the filtered measurement of CO₂ concentration, indoor temperature, and humidity. These estimations are then passed to control design/decision-making, generating commands to regulate the greenhouse's CO₂ supply, heating, and ventilation systems, see Figure. 9.

5. Considerations for Designing Biosystem Software Sensors

While the applications and case studies highlighted earlier have demonstrated the valuable advantages that software sensors can offer in the field of real-time monitoring and control of biosystems, it is important to acknowledge and address various considerations and constraints when designing such sensors. In this section, we will explore number of common considerations that software sensor designers need to carefully account for, covering challenges related to the biological system itself and the data-driven modelling approach.

5.1. Challenges Related to the Biological System

Biological systems present a unique set of challenges due to their inherent complexity, non-linear behaviour, time-varying dynamics, and individual variability [1]. These characteristics make it challenging to accurately model and predict the behaviour of biological process.

5.1.1. Complexity

The complexity inherent in biological systems stems from their multidimensional nature, characterized by intricate interactions between various processes and agents operating at different scales. As a result, biological systems exist on the “edge of chaos,” where they exhibit a delicate balance between order and disorder [155]. While they can show regular and predictable behaviour, these systems are also prone to sudden, massive, and stochastic changes in response to seemingly minor perturbations. This characteristic makes them exceptionally challenging to model, monitor, and control, especially when compared to more predictable man-made systems, such as car engines.

Therefore, when designing a software sensor for monitoring or controlling such systems, it is crucial to select and train data-driven models capable of incorporating stochastic elements to accurately capture and predict the likelihood of sudden changes. For instance, Bayesian networks can be used to predict the probability of sudden changes in disease progression [156] or ecological shifts [157] by modelling the dependencies between various biological factors. Statistical models such as hidden Markov models can be employed in software sensors to detect sudden changes in physiological signals, such as heart rate variability or neural activity patterns [158]. For more complex applications, deep learning models with dropout regularization can be used for image classifications in medical and diagnostic applications [159].

5.1.2. Time Variability

Biological processes are inherently dynamic, characterized by continuous change over time due to numerous internal factors, such as time-dependent biochemical pathways, circadian rhythms, hormonal cycles, and aging, as well as external factors, such as fluctuations in environmental

conditions and stressors like pathogens or injuries. This temporal variability poses significant challenges in monitoring, modelling, and controlling biological systems as well. To account for such temporal variations during software sensor design process, continuous and automated data acquisition systems, such as wearable sensors [52], contactless sensors [160,161], and remote sensing technologies [162], can be employed to ensure that critical information and dynamics are not missed. Also, machine learning techniques such as recurrent neural networks, including long short-term memory networks (LSTM), are specifically designed to handle sequential data and can effectively model temporal dependencies in biological systems [163]. For controlling biological systems, adaptive control strategies, such as model predictive controllers (MPC), can anticipate future states and make proactive adjustments to control strategies to accommodate for the time-varying nature of the controlled biological system [50].

5.1.3. Individual Variability

Biological systems are intrinsically characterized by significant individual variability, which adds an extra layer of complexity to their monitoring, modelling, and control. This variability arises from genetic and environmental factors, resulting in diverse responses among different individual of the same species, and even within the same individual at different times [1]. During modelling process, capturing the full range of responses within a population requires models that can handle high level of heterogeneity. In such cases, a single, general model often is not enough. Additionally, accurately estimating model parameters becomes challenging when dealing with individual variability, as it requires extensive data from multiple individuals or time points. Therefore, personalized modelling approaches, which can be tailored to specific individuals, can be employed to account the unique individual variability and dynamic responses. For example, a personalized SVM model in combination with k-nearest neighbour (KNN) is utilized to predict individual human thermal comfort [60].

5.2. Challenges Related to the Modelling Step

Despite its several advantages, data-driven modelling comes with several challenges that must be addressed to ensure its effectiveness and reliability for software sensor applications. Here are some key challenges:

5.2.1. Model Complexity

One crucial aspect of the identification process in modelling, regardless of the specific technique used, is defining a reduced-order model following the Occam's razor principle [164]. Model complexity refers to the capacity of a model to capture the underlying patterns in the data. A model with high complexity can fit almost any data perfectly, including the noise, which can lead to *overfitting*. On the contrary, a model with low complexity may be too simplistic to capture the essential features of the data, leading to *underfitting*. The challenge lies in finding the right (optimal) level of complexity that results the best predictive performance. This trade-off between model simplicity and performance highlights the need for a good understanding of the bias-variance dilemma [165] in the modelling step of the software sensor design [16]. It refers to the trade-off encountered when making statistical predictions, such as fitting a function, between the accuracy of the prediction and its precision. This trade-off is essentially between bias (the inverse of accuracy) and variance (the inverse of precision). The prediction error, which measures the difference between actual and predicted values, consists of three components (Figure 10):

- **Bias Error:** This reflects the model's ability to fit the training data. A model with high bias (indicative of low complexity and low accuracy) tends to miss relevant relationships between input features and the target output, leading to underfitting (Figure 10).
- **Variance Error:** This indicates the model's sensitivity to small fluctuations in the training data. A model with high variance (indicative of high complexity and low precision) captures the noise in the training data, making it less stable and resulting in overfitting. Such models perform well on training data but inadequately on unseen data due to their high complexity (Figure 10).

- **Noise:** This represents the irreducible error inherent in any biological data, which cannot be eliminated during the design process.

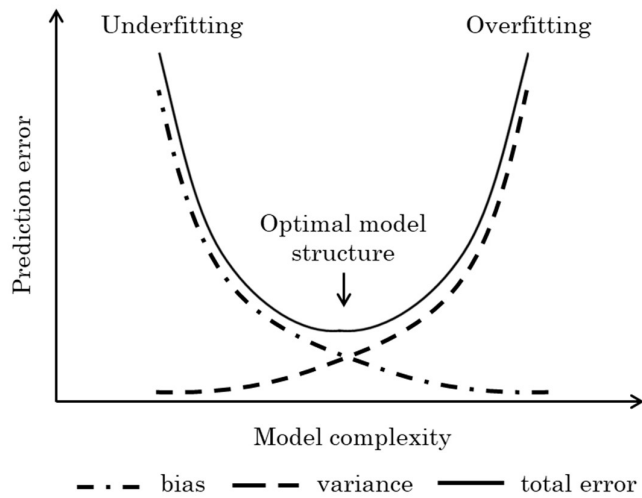


Figure 10. Relationship between model complexity, prediction error, bias, and variance. The optimal model structure balances bias and variance to minimize total error to avoid underfitting and overfitting.

5.2.2. Model Interpretability

During the design process of software sensors, it is also important to carefully consider the *interpretability* of the selected data-based model. Here comes another trade-off between model performance and these two concepts, which is largely depends on the objectives of the software sensor design. Essentially, this trade-off involves deciding whether it is more important to achieve the best prediction performance or to extend the understanding of the system, potentially at the expense of some accuracy [16].

Model interpretability refers to how each of the input (auxiliary) variables contributes to the prediction of the target output. An interpretable model provides clear insights into how input features are mathematically mapped to target outputs. Figure 11 provides a generalized view of the relationship between model interpretability and accuracy for some commonly used data-driven models. It is important to recognize that this comparison is a simplification, and actual performance can vary significantly depending on specific datasets and the problem domain. Take for example the following linear regression model [52], which is a common example of an interpretable model:

$$Y = a_0 + a_1X_1 + a_2X_2 + \dots + a_nX_n + e,$$

where Y the target output (e.g., thermal comfort) is directly related to the input feature variables $\{X_1, X_2, \dots, X_n\}$, such as heart rate, skin temperature, and air temperature [52], and the coefficients $\{a_1, a_2, \dots, a_n\}$ quantifies the contribution of each input variable. Each coefficient in the linear regression model directly indicates the impact of a corresponding input on the target output, making the model highly interpretable and easy to understand for stakeholders.

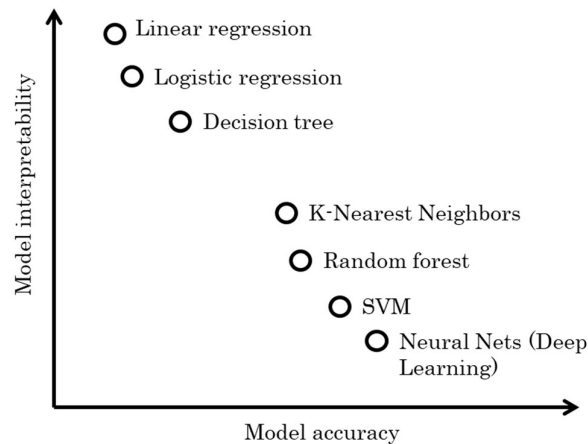


Figure 11. The interpretability-accuracy trade-off in in data-driven modelling: as model complexity increases, accuracy often improves but interpretability suffers.

Generally, in scenarios where transparency and comprehensibility are crucial, such as in clinical diagnostics and exploratory research, highly interpretable models are preferred. However, in cases where maximizing prediction accuracy is crucial, more complex models like deep neural networks (DNNs) are often employed. DNNs involve data passing through multiple layers of multiplications with learned weights and non-linear transformations. A single prediction may result from millions of mathematical operations, rendering such models almost impossible to be fully interpreted.

6. Summary

This review article explores the concept of software sensor as a vital instrument for the real-time monitoring and control of biosystems. Biosystems, which encompass a wide range of complexities from cellular processes to ecological interactions, are characterized by their time-varying, inherent nonlinearity, and individual variability. Hence, the ultimate goal of biosystem monitoring is to uncover underlying mechanisms, identify potential issues, and optimize conditions for enhanced efficiency and productivity. This is particularly relevant in modern industrial applications such as agriculture and biotechnology, where real-time decision-making can significantly impact productivity and sustainability.

Despite the advancements in modern hardware sensors, the article highlights a critical challenge: what is typically measurable may not align with what needs to be monitored. Traditional sensing approaches often fall short in capturing all necessary data due to difficulties related to physical accessibility or the high costs associated with advanced measuring techniques. Meanwhile, there's a specific need for a deeper understanding of nuanced behaviours and emergent properties of these systems, such as stress levels, health status, and overall wellbeing.

The article emphasizes that software sensors can effectively address these gaps by combining data from hardware sensors as auxiliary variables with estimations model and computational algorithms, enabling the accurate inference of target variables that are challenging to measure directly. The integration of enhanced sensing technologies and advanced computational methods has empowered significant advancements in the capabilities of software sensors, allowing for the continuous and unobtrusive monitoring of biosystems. The model-based monitoring and control framework allows for greater predictive power and responsiveness, essential for effective decision-making.

The article provides a brief overview of the common methodologies for designing software sensors with the main focus on the estimation algorithm and modelling step. Providing an overview of various modelling approaches, distinguishing the main differences between hypothetico-

deductive (mechanistic) models and inductive (data-driven) models such as linear regression, support vector machines, and neural networks.

Moreover, the article outlines practical applications of software sensors in diverse fields, from environmental monitoring to precision agriculture and healthcare. The article presents three case studies, showcasing in detail three real-world application of software sensors in biosystem monitoring and control. In the first case study, the focus is on non-invasive disease monitoring in animals, which demonstrates how software sensors can provide early warnings for respiratory infections by analysing sound signals and cough patterns. Additionally, these sensors show remarkable versatility in multi-sensor fusion applications, allowing them to operate with various hardware sensors to enhance fault detection and minimize uncertainty. The second case study further illustrates this potential, illustrating their ability to integrate data from different hardware tools, such as satellite imagery and hyperspectral signals, to accurately predict target variables, namely, soil organic carbon (SOC) levels. Finally, the third case study presents the use of data assimilation techniques, such as Kalman filter, in automated greenhouse control. This example demonstrates how software sensors can optimize environmental conditions for crop growth, ultimately leading to improved yields and greater resource efficiency.

While the article thoroughly discusses the advantages of software sensors, it also addresses various considerations essential to their design process. Challenges related to the biological systems themselves, such as complexity, time-variability, and individual differences, are discussed, along with commonly used computational techniques to mitigate these issues. Additionally, the article outlines the challenges inherent in data-driven modelling approach, particularly the trade-off between model complexity and performance. The bias-variance dilemma is discussed as a key concern for software sensor designers.

Another considerable trade-off highlighted is between model performance and interpretability. Simpler models like linear regression offer high interpretability but may fail to capture complex patterns as effectively as more advanced models like deep neural networks, which, despite their higher accuracy, suffer from poor interpretability. Ultimately, selecting the most suitable modelling technique should align with the specific design objectives of the software sensor, balancing accuracy, interpretability, and the ability to adapt to the dynamic nature of biological systems.

In conclusion, software sensors represent a significant advancement in the monitoring and control of biosystems, enabling better decision-making and optimization through the integration of advanced hardware technologies and computational techniques. As we continue to develop these technologies, the collaboration between interdisciplinary teams, including data scientists, biologists, and engineers, will be crucial to addressing the complexities and challenges of modern biosystems.

Author Contributions: Conceptualization, A.Y. and N.B.; methodology, A.Y., N.B. and X.C.; investigation, A.Y., N.B. and X.C.; resources, A.Y., N.B. and X.C.; data curation, A.Y., N.B. and X.C.; writing—original draft preparation, A.Y., N.B. and X.C.; writing—review and editing, A.Y., N.B. and X.C.; visualization, A.Y., N.B. and X.C.; supervision, A.Y. and N.B. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Youssef, A. Model-Based Control of Micro-Environment with Real-Time Feedback of Bioresponses, KU Leuven, 2014.
2. Ruth, M.; Hannon, B. *Modeling Dynamic Biological Systems*; Springer New York: New York, NY, 1997; ISBN 978-1-4612-6856-7.
3. Aerts, J.M. Modelling of Bio-Responses to Micro-Environmental Variables Using Dynamic Data-Based Models. Ph.D. Thesis, Katholieke Universiteit Leuven, 2001.
4. Young, P.C. Data-Based Mechanistic and Topdown Modelling. In Proceedings of the Proceedings International Environmental Modelling and Software Society Conference; 2002; pp. 363–374.
5. Mainka, T.; Mahler, N.; Herwig, C.; Pflügl, S. Soft Sensor-Based Monitoring and Efficient Control Strategies of Biomass Concentration for Continuous Cultures of *Haloferax Mediterranei* and Their Application to an Industrial Production Chain. *Microorganisms* **2019**, *7*, doi:10.3390/MICROORGANISMS7120648.
6. Paul, H.L. Energy Expenditure during Extravehicular Activity through Apollo. *42nd International Conference on Environmental Systems 2012, ICES 2012* **2012**, doi:10.2514/6.2012-3504.

7. Cai, Y.; Zheng, W.; Zhang, X.; Zhangzhong, L.; Xue, X. Research on Soil Moisture Prediction Model Based on Deep Learning. *PLoS One* **2019**, *14*, e0214508, doi:10.1371/JOURNAL.PONE.0214508.
8. Moghadas, D.; Badorreck, A. Machine Learning to Estimate Soil Moisture from Geophysical Measurements of Electrical Conductivity. *Near Surface Geophysics* **2019**, *17*, 181–195, doi:10.1002/NSG.12036.
9. Al-Abbas, A.H.; Swain, P.H.; Baumgardner, M.F. Relating Organic Matter and Clay Content to the Multispectral Radiance of Soils. *Soil Sci* **1972**, *114*, 477–485, doi:10.1097/00010694-197212000-00011.
10. Mozaffari, H.; Moosavi, A.A.; Cornelis, W. Vis-NIR-Spectroscopy- and Loss-on-Ignition-Based Functions to Estimate Organic Matter Content of Calcareous Soils. <https://doi.org/10.1080/03650340.2022.2047944> **2022**, doi:10.1080/03650340.2022.2047944.
11. Liu, C.; Zhai, Z.; Zhang, R.; Bai, J.; Zhang, M. Field Pest Monitoring and Forecasting System for Pest Control. *Front Plant Sci* **2022**, *13*, 990965, doi:10.3389/FPLS.2022.990965/BIBTEX.
12. Baulch, H.M.; Elliott, J.A.; Cordeiro, M.R.C.; Flaten, D.N.; Lobb, D.A.; Wilson, H.F. Soil and Water Management: Opportunities to Mitigate Nutrient Losses to Surface Waters in the Northern Great Plains. *Environmental Reviews* **2019**, *27*, 447–477, doi:10.1139/ER-2018-0101/ASSET/IMAGES/ER-2018-0101TAB5.GIF.
13. Blasco, X.; Martínez, M.; Herrero, J.M.; Ramos, C.; Sanchis, J. Model-Based Predictive Control of Greenhouse Climate for Reducing Energy and Water Consumption. *Comput Electron Agric* **2007**, *55*, 49–70, doi:10.1016/J.COMPAG.2006.12.001.
14. Liu, C.; Chen, F.; Li, Z.; Cocq, K. Le; Liu, Y.; Wu, L. Impacts of Nitrogen Practices on Yield, Grain Quality, and Nitrogen-Use Efficiency of Crops and Soil Fertility in Three Paddy-Upland Cropping Systems. *J Sci Food Agric* **2021**, *101*, 2218–2226, doi:10.1002/JFSA.10841.
15. Westermann, D.T.; Tindall, T.A.; James, D.W.; Hurst, R.L. Nitrogen and Potassium Fertilization of Potatoes: Yield and Specific Gravity. *Am Potato J* **1994**, *71*, 417–431, doi:10.1007/BF02849097.
16. Youssef, A. Soft Sensor and Biosensing. *Encyclopedia of Smart Agriculture Technologies* **2023**, 1–10, doi:10.1007/978-3-030-89123-7_171-1.
17. Tham, M.T.; Montague, G.A.; Julian Morris, A.; Lant, P.A. Soft-Sensors for Process Estimation and Inferential Control. *J Process Control* **1991**, *1*, 3–14, doi:10.1016/0959-1524(91)87002-F.
18. Carstensen, J.; Harremoës, P.; Strube, R. Software Sensors Based on the Grey-Box Modelling Approach. *Water Science and Technology* **1996**, *33*, 117–126, doi:10.1016/0273-1223(96)00164-3.
19. De Assis, A.J.; Maciel Filho, R. Soft Sensors Development for On-Line Bioreactor State Estimation. *Comput Chem Eng* **2000**, *24*, 1099–1103, doi:10.1016/S0098-1354(00)00489-0.
20. Javaid, M.; Haleem, A.; Rab, S.; Pratap Singh, R.; Suman, R. Sensors for Daily Life: A Review. *Sensors International* **2021**, *2*, 100121, doi:10.1016/J.SINTL.2021.100121.
21. Aparna, K.; Dayajanaki, D.H.; Devika Rani, P.; Devu, D.; Rajeev, S.P.; Baby Sreeja, S.D. Wearable Sensors in Daily Life: A Review. *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023* **2023**, 863–868, doi:10.1109/ICACCS57279.2023.10112956.
22. Parajuli, P.B.; Jayakody, P.; Sassenrath, G.F.; Ouyang, Y.; Pote, J.W. Assessing the Impacts of Crop-Rotation and Tillage on Crop Yields and Sediment Yield Using a Modeling Approach. *Agric Water Manag* **2013**, *119*, 32–42, doi:10.1016/J.AGWAT.2012.12.010.
23. Ferencz, Cs.; Bognár, P.; Lichtenberger, J.; Hamar, D.; Tarcsait, Gy.; Timár, G.; Molnár, G.; Pásztor, SZ.; Steinbach, P.; Székely, B.; et al. Crop Yield Estimation by Satellite Remote Sensing. *Int J Remote Sens* **2004**, *25*, 4113–4149, doi:10.1080/01431160410001698870.
24. Al-Gaadi, K.A.; Hassaballa, A.A.; Tola, E.; Kayad, A.G.; Madugundu, R.; Alblewi, B.; Assiri, F. Prediction of Potato Crop Yield Using Precision Agriculture Techniques. *PLoS One* **2016**, *11*, e0162219, doi:10.1371/JOURNAL.PONE.0162219.
25. Beylen, K. Van; Youssef, A.; Aerts, J.-M.; Lambrechts, T.; Papantoniou, I. Metabolite-Based Model Predictive Control of Cell Growth. *Advancing Manufacture of Cell and Gene Therapies VI* **2019**.
26. Aerts, J.-M.; Albert, B.D.; Matheus, V.E.J. Method and System for Controlling Bioresponse of Living Organisms 2002.
27. Berckmans, D. Basic Principles of PLF: Gold Standard, Labelling and Field Data. In Proceedings of the Precision Livestock Farming 2013; Leuven, September 10 2013; pp. 21–55.
28. Tullo, E.; Fontana, I.; Diana, A.; Norton, T.; Berckmans, D.; Guarino, M. Application Note: Labelling, a Methodology to Develop Reliable Algorithm in PLF. *Comput Electron Agric* **2017**, *142*, 424–428, doi:10.1016/J.COMPAG.2017.09.030.
29. Jia, C.; Zhou, T.; Zhang, K.; Yang, L.; Zhang, D.; Cui, T.; He, X.; Sang, X. Design and Experimentation of Soil Organic Matter Content Detection System Based on High-Temperature Excitation Principle. *Comput Electron Agric* **2023**, *214*, 108325, doi:10.1016/J.COMPAG.2023.108325.
30. Mora, M.D.; Germani, A.; Manes, C. A State Observer for Nonlinear Dynamical Systems. *Nonlinear Anal Theory Methods Appl* **1997**, *30*, 4485–4496, doi:10.1016/S0362-546X(97)00184-3.
31. Tham, M.T.; Montague, G.A.; Julian Morris, A.; Lant, P.A. Soft-Sensors for Process Estimation and Inferential Control. *J Process Control* **1991**, *1*, 3–14, doi:10.1016/0959-1524(91)87002-F.

32. McAfee, M.; Kariminejad, M.; Weinert, A.; Huq, S.; Stigter, J.D.; Tormey, D. State Estimators in Soft Sensing and Sensor Fusion for Sustainable Manufacturing. *Sustainability* **2022**, *Vol. 14*, Page 3635 **2022**, *14*, 3635, doi:10.3390/SU14063635.
33. Joseph, B.; Brosilow, C.B. Inferential Control of Processes: Part I. Steady State Analysis and Design. *AIChE Journal* **1978**, *24*, 485–492.
34. Luenberger, D.G. Observers for Multivariable Systems. *IEEE Trans Automat Contr* **1966**, *11*, 190–197, doi:10.1109/TAC.1966.1098323.
35. Van Dooren, P. Reduced Order Observers: A New Algorithm and Proof. *Syst Control Lett* **1984**, *4*, 243–251, doi:10.1016/S0167-6911(84)80033-X.
36. Solsona, J.; Valla, M.I.; Muravchik, C. Nonlinear Reduced Order Observer for Permanent Magnet Synchronous Motors. *IECON Proceedings (Industrial Electronics Conference)* **1994**, *1*, 38–43, doi:10.1109/IECON.1994.397746.
37. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* **1960**, *82*, 35–45, doi:10.1115/1.3662552.
38. Palatella, L.; Trevisan, A. Interaction of Lyapunov Vectors in the Formulation of the Nonlinear Extension of the Kalman Filter. *Phys Rev E Stat Nonlin Soft Matter Phys* **2015**, *91*, 042905, doi:10.1103/PHYSREVE.91.042905/FIGURES/4/MEDIUM.
39. Ljung, L. Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems. *IEEE Trans Automat Contr* **1979**, *24*, 36–50, doi:10.1109/TAC.1979.1101943.
40. Julier, S.J.; Uhlmann, J.K. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE* **2004**, *92*, 401–422, doi:10.1109/JPROC.2003.823141.
41. Aanonsen, S.I.; N  vdal, G.; Oliver, D.S.; Reynolds, A.C.; Vall  s, B. The Ensemble Kalman Filter in Reservoir Engineering—a Review. *SPE Journal* **2009**, *14*, 393–412, doi:10.2118/117274-PA.
42. Evensen, G. The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation. *Ocean Dyn* **2003**, *53*, 343–367, doi:10.1007/S10236-003-0036-9/METRICS.
43. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing* **2002**, *50*, 174–188, doi:10.1109/78.978374.
44. Young, P.C. *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*; Springer: Berlin, Heidelberg, 2011; ISBN 3642219810.
45. Voit, E.O. The Best Models of Metabolism. *Wiley Interdiscip Rev Syst Biol Med* **2017**, *9*, doi:10.1002/WSBM.1391.
46. Fortuna, L.; Graziani, S.; Rizzo, A.; Xibilia Maria G. *Soft Sensors for Monitoring and Control of Industrial Processes*; Springer, 2007; ISBN 978-1-84628-479-3.
47. Paulsson, D.; Gustavsson, R.; Mandenius, C.F. A Soft Sensor for Bioprocess Control Based on Sequential Filtering of Metabolic Heat Signals. *Sensors (Switzerland)* **2014**, *14*, 17864–17882, doi:10.3390/S141017864.
48. Kadlec, P.; Gabrys, B.; Strandt, S. Data-Driven Soft Sensors in the Process Industry. *Comput Chem Eng* **2009**, *33*, 795–814, doi:10.1016/J.COMPCHEMENG.2008.12.012.
49. Kadlec, P.; Grbi  , R.; Gabrys, B. Review of Adaptation Mechanisms for Data-Driven Soft Sensors. *Comput Chem Eng* **2011**, *35*, 1–24.
50. Van Beylen, K.; Youssef, A.; Fern  ndez, A.P.; Lambrechts, T.; Papantoniou, I.; Aerts, J.M. Lactate-Based Model Predictive Control Strategy of Cell Growth for Cell Therapy Applications. *Bioengineering* **2020**, *Vol. 7*, Page 78 **2020**, *7*, 78, doi:10.3390/BIOENGINEERING7030078.
51. Pe  a Fern  ndez, A.; Norton, T.; Youssef, A.; Exadaktylos, V.; Bahr, C.; Bruininx, E.; Vranken, E.; Berckmans, D. Real-Time Modelling of Individual Weight Response to Feed Supply for Fattening Pigs. *Comput Electron Agric* **2019**, *162*, 895–906.
52. Youssef, A.; Colon, J.; Mantzios, K.; Gkiata, P.; Mayor, T.S.; Flouris, A.D.; De Bruyne, G.; Aerts, J.-M. Towards Model-Based Online Monitoring of Cyclist’s Head Thermal Comfort: Smart Helmet Concept and Prototype. *Applied Sciences (Switzerland)* **2019**, *9*, doi:10.3390/app9153170.
53. Debeljak, M.; D  zeroski, S. Decision Trees in Ecological Modelling. *Modelling Complex Ecological Dynamics: An Introduction into Ecological Modelling for Students, Teachers & Scientists* **2011**, 197–209, doi:10.1007/978-3-642-05029-9_14/FIGURES/5_14.
54. Roozbeh, M.; Rouhi, A.; Mohamed, N.A.; Jahadi, F.; Arashi, M.; Roozbeh, M.; Rouhi, A.; Anisah Mohamed, N.; Jahadi, F. Generalized Support Vector Regression and Symmetry Functional Regression Approaches to Model the High-Dimensional Data. *Symmetry* **2023**, *Vol. 15*, Page 1262 **2023**, *15*, 1262, doi:10.3390/SYM15061262.
55. Rocca, M. La; Perna, C.; Rocca, M. La; Perna, C. Designing Neural Networks for Modeling Biological Data: A Statistical Perspective. *Mathematical Biosciences and Engineering* **2014**, *2:331* **2014**, *11*, 331–342, doi:10.3934/MBE.2014.11.331.

56. Carpentier, L.; Berckmans, D.; Youssef, A.; Berckmans, D.; van Waterschoot, T.; Johnston, D.; Ferguson, N.; Earley, B.; Fontana, I.; Tullo, E.; et al. Automatic Cough Detection for Bovine Respiratory Disease in a Calf House. *Biosyst Eng* **2018**, *173*, 45–56, doi:10.1016/J.BIOSYSTEMSENG.2018.06.018.
57. Youssef, A.; Jansen, C.; Neethirajan, S.R. Soft-Sensing Approach for Predicting Bovine Respiratory Disease Severity. In Proceedings of the Precision Livestock Farming '22; University of Veterinary Medicine Vienna: Vienna, August 29 2022; pp. 932–939.
58. Youssef, A.; Youssef Ali Amer, A.; Caballero, N.; Aerts, J.-M. Towards Online Personalized-Monitoring of Human Thermal Sensation Using Machine Learning Approach. *Applied Sciences* **2019**, *9*, 3303.
59. Piccini, C.; Marchetti, A.; Riviaccio, R.; Napoli, R. Multinomial Logistic Regression with Soil Diagnostic Features and Land Surface Parameters for Soil Mapping of Latium (Central Italy). *Geoderma* **2019**, *352*, 385–394, doi:10.1016/J.GEODERMA.2018.09.037.
60. Youssef, A.; Amer, A.Y.A.; Caballero, N.; Aerts, J.M. Towards Online Personalized-Monitoring of Human Thermal Sensation Using Machine Learning Approach. *Applied Sciences* **2019**, *Vol. 9*, Page 3303 **2019**, *9*, 3303.
61. Kong, Y.; Yu, T. A Deep Neural Network Model Using Random Forest to Extract Feature Representation for Gene Expression Data Classification. *Scientific Reports* **2018**, *8*, 1–9, doi:10.1038/s41598-018-34833-6.
62. Tadeusiewicz, R. Neural Networks as a Tool for Modeling of Biological Systems. *Bio-Algorithms and Med-Systems* **2015**, *11*, 135–144, doi:10.1515/BAMS-2015-0021.
63. Fang, K.; Kifer, D.; Lawson, K.; Shen, C. Evaluating the Potential and Challenges of an Uncertainty Quantification Method for Long Short-Term Memory Models for Soil Moisture Predictions. *Water Resour Res* **2020**, *56*, e2020WR028095, doi:10.1029/2020WR028095.
64. Youssef, A.; Berckmans, D.; Norton, T. Non-Invasive PPG-Based System for Continuous Heart Rate Monitoring of Incubated Avian Embryo. *Sensors* **2020**, *Vol. 20*, Page 4560 **2020**, *20*, 4560.
65. Youssef, A.; Peña Fernández, A.; Wassermann, L.; Biernot, S.; Wittauer, E.-M.; Bleich, A.; Hartung, J.; Berckmans, D.; Norton, T. An Approach towards Motion-Tolerant PPG-Based Algorithm for Real-Time Heart Rate Monitoring of Moving Pigs. *Sensors* **2020**, *20*, 4251.
66. Youssef, A.; Exadaktylos, V.; Berckmans, D.A.D.A. Towards Real-Time Control of Chicken Activity in a Ventilated Chamber. *Biosyst Eng* **2015**, *135*, 31–43.
67. Peña Fernández, A.; Demmers, T.G.M.; Tong, Q.; Youssef, A.; Norton, T.; Vranken, E.; Berckmans, D. Real-Time Modelling of Indoor Particulate Matter Concentration in Poultry Houses Using Broiler Activity and Ventilation Rate. *Biosyst Eng* **2019**, *187*, doi:10.1016/j.biosystemseng.2019.09.004.
68. Peña Fernández, A.; Tullo, E.; van Hertem, T.; Youssef, A.; Exadaktylos, V.; Vranken, E.; Guarino, M. Real-Time Monitoring of Broiler Flock's Welfare Status Using Camera-Based Technology. *Biosyst Eng* **2018**, *173*, 103–114.
69. Ferrari, S.; Silva, M.; Guarino, M.; Aerts, J.M.; Berckmans, D. Cough Sound Analysis to Identify Respiratory Infection in Pigs. *Comput Electron Agric* **2008**, *64*, 318–325, doi:10.1016/J.COMPAG.2008.07.003.
70. Chung, Y.; Oh, S.; Lee, J.; Park, D.; Chang, H.H.; Kim, S. Automatic Detection and Recognition of Pig Wasting Diseases Using Sound Data in Audio Surveillance Systems. *Sensors* **2013**, *Vol. 13*, Pages 12929–12942 **2013**, *13*, 12929–12942, doi:10.3390/S131012929.
71. Yin, Y.; Tu, D.; Shen, W.; Bao, J. Recognition of Sick Pig Cough Sounds Based on Convolutional Neural Network in Field Situations. *Information Processing in Agriculture* **2021**, *8*, 369–379, doi:10.1016/J.INPA.2020.11.001.
72. Hassani, F.A. Bioreceptor-Inspired Soft Sensor Arrays: Recent Progress towards Advancing Digital Healthcare. *Soft Science* **2023**, *3*, null-null, doi:10.20517/SS.2023.23.
73. Gao, S.; Qiu, S.; Ma, Z.; Tian, R.; Liu, Y. SVAE-WGAN-Based Soft Sensor Data Supplement Method for Process Industry. *IEEE Sens J* **2022**, *22*, 601–610, doi:10.1109/JSEN.2021.3128562.
74. Choi, D.J.; Park, H. A Hybrid Artificial Neural Network as a Software Sensor for Optimal Control of a Wastewater Treatment Process. *Water Res* **2001**, *35*, 3959–3967, doi:10.1016/S0043-1354(01)00134-8.
75. Paepae, T.; Bokoro, P.N.; Kyamakya, K. Data Augmentation for a Virtual-Sensor-Based Nitrogen and Phosphorus Monitoring. *Sensors (Basel)* **2023**, *23*, doi:10.3390/S23031061.
76. Schneider, M.Y.; Furrer, V.; Sprenger, E.; Carbajal, J.P.; Villez, K.; Maurer, M. Benchmarking Soft Sensors for Remote Monitoring of On-Site Wastewater Treatment Plants. *Environ Sci Technol* **2020**, *54*, 10840–10849, doi:10.1021/ACS.EST.9B07760/SUPPL_FILE/ES9B07760_SI_001.PDF.
77. Maniscalco, U.; Pilato, G.; Vella, F. Soft Sensor Network for Environmental Monitoring. *Smart Innovation, Systems and Technologies* **2016**, *55*, 705–714, doi:10.1007/978-3-319-39345-2_63.
78. Murugan, C.; Natarajan, P. Estimation of Fungal Biomass Using Multiphase Artificial Neural Network Based Dynamic Soft Sensor. *J Microbiol Methods* **2019**, *159*, 5–11, doi:10.1016/J.MIMET.2019.02.002.
79. Kaptan, C.; Kantarci, B.; Soyata, T.; Boukerche, A. Emulating Smart City Sensors Using Soft Sensing and Machine Intelligence: A Case Study in Public Transportation. *IEEE International Conference on Communications* **2018**, *2018-May*, doi:10.1109/ICC.2018.8422969.

80. Hu, Z.; Yang, R.; Fang, L.; Wang, Z.; Zhao, Y. Research on Vehicle Speed Prediction Model Based on Traffic Flow Information Fusion. *Energy* **2024**, *292*, doi:10.1016/j.energy.2024.130416.
81. Juma, M.; Shaalan, K. Cyberphysical Systems in the Smart City: Challenges and Future Trends for Strategic Research. *Swarm Intelligence for Resource Management in Internet of Things* **2020**, *65–85*, doi:10.1016/B978-0-12-818287-1.00008-5.
82. Barodi, A.; Zemmouri, A.; Bajit, A.; Benbrahim, M.; Tamtaoui, A. Intelligent Transportation System Based on Smart Soft-Sensors to Analyze Road Traffic and Assist Driver Behavior Applicable to Smart Cities. *Microprocess Microsyst* **2023**, *100*, 104830, doi:10.1016/J.MICPRO.2023.104830.
83. Pech, M.; Vrchota, J.; Bednář, J. Predictive Maintenance and Intelligent Sensors in Smart Factory: Review. *Sensors* **2021**, *21*, 1–39, doi:10.3390/S21041470.
84. Ruiz-Arenas, S.; Horváth, I.; Mejía-Gutiérrez, R.; Opiyo, E.Z. Towards the Maintenance Principles of Cyber-Physical Systems. *Strojnicki Vestnik/Journal of Mechanical Engineering* **2014**, *60*, 815–831, doi:10.5545/SV-JME.2013.1556.
85. Papa, G.; Zurutuza, U.; Uribeetxeberria, R. Cyber Physical System Based Proactive Collaborative Maintenance. *Proceedings of 2016 International Conference on Smart Systems and Technologies, SST 2016* **2016**, 173–178, doi:10.1109/SST.2016.7765654.
86. Alassery, F. Predictive Maintenance for Cyber Physical Systems Using Neural Network Based on Deep Soft Sensor and Industrial Internet of Things. *Computers and Electrical Engineering* **2022**, *101*, 108062, doi:10.1016/J.COMPELECENG.2022.108062.
87. Shcherbakov, M. V.; Glotov, A. V.; Cheremisinov, S. V. Proactive and Predictive Maintenance of Cyber-Physical Systems. *Studies in Systems, Decision and Control* **2020**, *259*, 263–278, doi:10.1007/978-3-030-32579-4_21.
88. Zhang, Y.; Huang, S.; Guo, S.; Zhu, J. Multi-Sensor Data Fusion for Cyber Security Situation Awareness. *Procedia Environ Sci* **2011**, *10*, 1029–1034, doi:10.1016/J.PROENV.2011.09.165.
89. Martínez-Monge, I.; Martínez, C.; Decker, M.; Udugama, I.A.; Marín de Mas, I.; Gernaey, K. V.; Nielsen, L.K. Soft-Sensors Application for Automated Feeding Control in High-Throughput Mammalian Cell Cultures. *Biotechnol Bioeng* **2022**, *119*, 1077–1090, doi:10.1002/BIT.28032.
90. Martínez-Monge, I.; Martínez, C.; Decker, M.; Udugama, I.A.; Marín de Mas, I.; Gernaey, K. V.; Nielsen, L.K. Soft-Sensors Application for Automated Feeding Control in High-Throughput Mammalian Cell Cultures. *Biotechnol Bioeng* **2022**, *119*, 1077–1090, doi:10.1002/BIT.28032.
91. Kroll, P.; Stelzer, I. V.; Herwig, C. Soft Sensor for Monitoring Biomass Subpopulations in Mammalian Cell Culture Processes. *Biotechnol Lett* **2017**, *39*, 1667–1673, doi:10.1007/S10529-017-2408-0.
92. Yan, A.; Shao, H.; Wang, P. A Soft-Sensing Method of Dissolved Oxygen Concentration by Group Genetic Case-Based Reasoning with Integrating Group Decision Making. *Neurocomputing* **2015**, *169*, 422–429, doi:10.1016/j.neucom.2014.07.081.
93. Sagmeister, P.; Kment, M.; Wechselberger, P.; Meitz, A.; Langemann, T.; Herwig, C. Soft-Sensor Assisted Dynamic Investigation of Mixed Feed Bioprocesses. *Process Biochemistry* **2013**, *48*, 1839–1847, doi:10.1016/J.PROCBIO.2013.09.018.
94. Sang, H.; Wang, F.; He, D.; Chang, Y.; Zhang, D. On-Line Estimation of Biomass Concentration and Specific Growth Rate in the Fermentation Process. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* **2006**, *1*, 4644–4648, doi:10.1109/WCICA.2006.1713262.
95. Bhattacharya, P.; Tanwar, S.; Bodkhe, U.; Tyagi, S.; Kumar, N. BinDaaS: Blockchain-Based Deep-Learning as-a-Service in Healthcare 4.0 Applications. *IEEE Trans Netw Sci Eng* **2021**, *8*, 1242–1255, doi:10.1109/TNSE.2019.2961932.
96. Mbunge, E.; Muchemwa, B.; Jiyane, S.; Batani, J. Sensors and Healthcare 5.0: Transformative Shift in Virtual Care through Emerging Digital Health Technologies. *Global Health Journal* **2021**, *5*, 169–177, doi:10.1016/J.GLOHJ.2021.11.008.
97. Hatamie, A.; Angizi, S.; Kumar, S.; Pandey, C.M.; Simchi, A.; Willander, M.; Malhotra, B.D. Review—Textile Based Chemical and Physical Sensors for Healthcare Monitoring. *J Electrochem Soc* **2020**, *167*, 037546, doi:10.1149/1945-7111/AB6827.
98. Youssef, A.; Amer, A.Y.A.; Caballero, N.; Aerts, J.-M. Towards Online Personalized-Monitoring of Human Thermal Sensation Using Machine Learning Approach. *Applied Sciences (Switzerland)* **2019**, *9*, doi:10.3390/app9163303.
99. Aydin, E.B.; Aydin, M.; Sezgin, M.K. Biosensors in Drug Discovery and Drug Analysis. *Curr Anal Chem* **2018**, *15*, 467–484, doi:10.2174/1573411014666180912131811.
100. Beke, Á.K.; Gyürkés, M.; Nagy, Z.K.; Marosi, G.; Farkas, A. Digital Twin of Low Dosage Continuous Powder Blending – Artificial Neural Networks and Residence Time Distribution Models. *European Journal of Pharmaceutics and Biopharmaceutics* **2021**, *169*, 64–77, doi:10.1016/j.ejpb.2021.09.006.
101. Jayaraman, P.P.; Yavari, A.; Georgakopoulos, D.; Morshed, A.; Zaslavsky, A. Internet of Things Platform for Smart Farming: Experiences and Lessons Learnt. *Sensors (Switzerland)* **2016**, *16*, doi:10.3390/S16111884.

102. García-Mañas, F.; Rodríguez, F.; Berenguel, M. Leaf Area Index Soft Sensor for Tomato Crops in Greenhouses. *IFAC-PapersOnLine* **2020**, *53*, 15796–15803, doi:10.1016/j.ifacol.2020.12.230.
103. Vaz, C.M.P.; Jones, S.; Meding, M.; Tuller, M. Evaluation of Standard Calibration Functions for Eight Electromagnetic Soil Moisture Sensors. *Vadose Zone Journal* **2013**, *12*, 1–16, doi:10.2136/vzj2012.0160.
104. Moghadas, D.; Badorreck, A. Machine Learning to Estimate Soil Moisture from Geophysical Measurements of Electrical Conductivity. *Near Surface Geophysics* **2019**, *17*, 181–195, doi:10.1002/NSG.12036.
105. Cui, H.; Jiang, L.; Paloscia, S.; Santi, E.; Pettinato, S.; Wang, J.; Fang, X.; Liao, W. The Potential of ALOS-2 and Sentinel-1 Radar Data for Soil Moisture Retrieval With High Spatial Resolution Over Agroforestry Areas, China. *IEEE Transactions on Geoscience and Remote Sensing* **2021**, doi:10.1109/TGRS.2021.3082805.
106. Shi, P.; Luan, X.; Liu, F.; Karimi, H.R. Kalman Filtering on Greenhouse Climate Control. In Proceedings of the Proceedings of the 31st Chinese Control Conference; 2012; pp. 779–784.
107. Van Henten, E.J. *Greenhouse Climate Management: An Optimal Control Approach*; Wageningen University and Research, 1994; ISBN 9798728200710.
108. Youssef, A.; Viazzi, S.; Exadaktylos, V.; Berckmans, D. Non-Contact, Motion-Tolerant Measurements of Chicken (*Gallus Gallus*) Embryo Heart Rate (HR) Using Video Imaging and Signal Processing. *Biosyst Eng* **2014**, *125*, 9–16.
109. Lu, M.; Norton, T.; Youssef, A.; Radojkovic, N.; Fernández, A.P.; Berckmans, D. Extracting Body Surface Dimensions from Top-View Images of Pigs. *International Journal of Agricultural and Biological Engineering* **2018**, *11*, 182–191, doi:10.25165/IJABE.V11I5.4054.
110. Peña Fernández, A.; Demmers, T.G.M.; Tong, Q.; Youssef, A.; Norton, T.; Vranken, E.; Berckmans, D. Real-Time Modelling of Indoor Particulate Matter Concentration in Poultry Houses Using Broiler Activity and Ventilation Rate. *Biosyst Eng* **2019**, *187*, 214–225.
111. Wang, M.; Youssef, A.; Larsen, M.; Rault, J.-L.; Berckmans, D.; Marchant-Forde, J.N.; Hartung, J.; Bleich, A.; Lu, M.; Norton, T. Contactless Video-Based Heart Rate Monitoring of a Resting and an Anesthetized Pig. *Animals* **2021**, *11*, doi:10.3390/ani11020442.
112. Yin, Y.; Ji, N.; Wang, X.; Shen, W.; Dai, B.; Kou, S.; Liang, C. An Investigation of Fusion Strategies for Boosting Pig Cough Sound Recognition. *Comput Electron Agric* **2023**, *205*, doi:10.1016/j.compag.2023.107645.
113. Exadaktylos, V.; Silva, M.; Aerts, J.M.; Taylor, C.J.; Berckmans, D. Real-Time Recognition of Sick Pig Cough Sounds. *Comput Electron Agric* **2008**, *63*, 207–214, doi:10.1016/j.compag.2008.02.010.
114. Shen, W.; Ji, N.; Yin, Y.; Dai, B.; Tu, D.; Sun, B.; Hou, H.; Kou, S.; Zhao, Y. Fusion of Acoustic and Deep Features for Pig Cough Sound Recognition. *Comput Electron Agric* **2022**, *197*, doi:10.1016/j.compag.2022.106994.
115. Guarino, M.; Jans, P.; Costa, A.; Aerts, J.M.; Berckmans, D. Field Test of Algorithm for Automatic Cough Detection in Pig Houses. *Comput Electron Agric* **2008**, *62*, 22–28, doi:10.1016/j.compag.2007.08.016.
116. Wang, X.; Yin, Y.; Dai, X.; Shen, W.; Kou, S.; Dai, B. Automatic Detection of Continuous Pig Cough in a Complex Piggery Environment. *Biosyst Eng* **2024**, *238*, 78–88, doi:10.1016/j.biosystemseng.2024.01.002.
117. Cuan, K.; Zhang, T.; Li, Z.; Huang, J.; Ding, Y.; Fang, C. Automatic Newcastle Disease Detection Using Sound Technology and Deep Learning Method. *Comput Electron Agric* **2022**, *194*, 106740, doi:10.1016/j.compag.2022.106740.
118. Vandermeulen, J.; Bahr, C.; Johnston, D.; Earley, B.; Tullo, E.; Fontana, I.; Guarino, M.; Exadaktylos, V.; Berckmans, D. Early Recognition of Bovine Respiratory Disease in Calves Using Automated Continuous Monitoring of Cough Sounds. *Comput Electron Agric* **2016**, *129*, 15–26, doi:10.1016/j.compag.2016.07.014.
119. Aerts, J.M.; Jans, P.; Halloy, D.; Gustin, P.; Berckmans, D. Labeling of Cough Data from Pigs for On-Line Disease Monitoring by Sound Analysis. *Transactions of the ASAE* **2005**, *48*, 351–354, doi:10.13031/2013.17948.
120. Exadaktylos, V.; Silva, M.; Berckmans, D. Automatic Identification and Interpretation of Animal Sounds, Application to Livestock Production Optimisation. In; 2014.
121. Exadaktylos, V.; Silva, M.; Aerts, J.-M.; Taylor, C.J.; Berckmans, D. Real-Time Recognition of Sick Pig Cough Sounds. *Comput Electron Agric* **2008**, *63*, 207–214.
122. Lagua, E.B.; Mun, H.S.; Ampode, K.M.B.; Chem, V.; Kim, Y.H.; Yang, C.J. Artificial Intelligence for Automatic Monitoring of Respiratory Health Conditions in Smart Swine Farming. *Animals (Basel)* **2023**, *13*, doi:10.3390/ANI13111860.
123. Biney, J.K.M.; Saberioon, M.; Borůvka, L.; Houška, J.; Vašát, R.; Agyeman, P.C.; Coblinski, J.A.; Klement, A. Exploring the Suitability of UAS-Based Multispectral Images for Estimating Soil Organic Carbon: Comparison with Proximal Soil Sensing and Spaceborne Imagery. *Remote Sensing* **2021**, *13*, 308, doi:10.3390/RS13020308.
124. Sanchez, P.A.; Ahamed, S.; Carré, F.; Hartemink, A.E.; Hempel, J.; Huising, J.; Lagacherie, P.; McBratney, A.B.; McKenzie, N.J.; De Lourdes Mendonça-Santos, M.; et al. Digital Soil Map of the World. *Science* (1979) **2009**, *325*, 680–681, doi:10.1126/SCIENCE.1175084.
125. Taghizadeh-Mehrjardi, R.; Nabiollahi, K.; Kerry, R. Digital Mapping of Soil Organic Carbon at Multiple Depths Using Different Data Mining Techniques in Baneh Region, Iran. *Geoderma* **2016**, *266*, 98–110, doi:10.1016/j.geoderma.2015.12.003.

126. Sodango, T.H.; Sha, J.; Li, X.; Noszczyk, T.; Shang, J.; Aneseyee, A.B.; Bao, Z. Modeling the Spatial Dynamics of Soil Organic Carbon Using Remotely-Sensed Predictors in Fuzhou City, China. *Remote Sensing* **2021**, *Vol. 13*, Page 1682 **2021**, *13*, 1682, doi:10.3390/RS13091682.
127. Goetz, S.; Dubayah, R. Advances in Remote Sensing Technology and Implications for Measuring and Monitoring Forest Carbon Stocks and Change. *Carbon Manag* **2011**, *2*, 231–244, doi:10.4155/CMT.11.18.
128. Paustian, K.; Collier, S.; Baldock, J.; Burgess, R.; Creque, J.; DeLonge, M.; Dungait, J.; Ellert, B.; Frank, S.; Goddard, T.; et al. Quantifying Carbon for Agricultural Soil Management: From the Current Status toward a Global Soil Information System. *Carbon Manag* **2019**, *10*, 567–587, doi:10.1080/17583004.2019.1633231.
129. Teke, M.; Deveci, H.S.; Haliloglu, O.; Gurbuz, S.Z.; Sakarya, U. A Short Survey of Hyperspectral Remote Sensing Applications in Agriculture. *RAST 2013 - Proceedings of 6th International Conference on Recent Advances in Space Technologies* **2013**, 171–176, doi:10.1109/RAST.2013.6581194.
130. Ge, Y.; Thomasson, J.A.; Sui, R. Remote Sensing of Soil Properties in Precision Agriculture: A Review. *Front Earth Sci* **2011**, *5*, 229–238, doi:10.1007/S11707-011-0175-0/METRICS.
131. Adamchuk, V.I.; Hummel, J.W.; Morgan, M.T.; Upadhyaya, S.K. On-the-Go Soil Sensors for Precision Agriculture. *Comput Electron Agric* **2004**, *44*, 71–91, doi:10.1016/j.compag.2004.03.002.
132. Kühnel, A.; Bogner, C. In-Situ Prediction of Soil Organic Carbon by Vis-NIR Spectroscopy: An Efficient Use of Limited Field Data. *Eur J Soil Sci* **2017**, *68*, 689–702, doi:10.1111/EJSS.12448.
133. Gomez, C.; Viscarra Rossel, R.A.; McBratney, A.B. Soil Organic Carbon Prediction by Hyperspectral Remote Sensing and Field Vis-NIR Spectroscopy: An Australian Case Study. *Geoderma* **2008**, *146*, 403–411, doi:10.1016/J.GEODERMA.2008.06.011.
134. Viscarra Rossel, R.A.; Walvoort, D.J.J.; McBratney, A.B.; Janik, L.J.; Skjemstad, J.O. Visible, near Infrared, Mid Infrared or Combined Diffuse Reflectance Spectroscopy for Simultaneous Assessment of Various Soil Properties. *Geoderma* **2006**, *131*, 59–75, doi:10.1016/J.GEODERMA.2005.03.007.
135. Laamrani, A.; Berg, A.A.; Voroney, P.; Feilhauer, H.; Blackburn, L.; March, M.; Dao, P.D.; He, Y.; Martin, R.C. Ensemble Identification of Spectral Bands Related to Soil Organic Carbon Levels over an Agricultural Field in Southern Ontario, Canada. *Remote Sensing* **2019**, *Vol. 11*, Page 1298 **2019**, *11*, 1298, doi:10.3390/RS11111298.
136. Bai, Z.; Xie, M.; Hu, B.; Luo, D.; Wan, C.; Peng, J.; Shi, Z. Estimation of Soil Organic Carbon Using Vis-NIR Spectral Data and Spectral Feature Bands Selection in Southern Xinjiang, China. *Sensors* **2022**, *Vol. 22*, Page 6124 **2022**, *22*, 6124, doi:10.3390/S22166124.
137. Uddin, M.P.; Mamun, M. Al; Hossain, M.A. PCA-Based Feature Reduction for Hyperspectral Remote Sensing Image Classification. *IETE Technical Review* **2021**, *38*, 377–396, doi:10.1080/02564602.2020.1740615.
138. Ibrahim, M.F.I.; Al-Jumaily, A.A. PCA Indexing Based Feature Learning and Feature Selection. *2016 8th Cairo International Biomedical Engineering Conference, CIBEC 2016* **2017**, 68–71, doi:10.1109/CIBEC.2016.7836122.
139. Geniaux, G.; Martinetti, D. A New Method for Dealing Simultaneously with Spatial Autocorrelation and Spatial Heterogeneity in Regression Models. *Reg Sci Urban Econ* **2018**, *72*, 74–85, doi:10.1016/J.REGSCIURBECO.2017.04.001.
140. Were, K.; Bui, D.T.; Dick, Ø.B.; Singh, B.R. A Comparative Assessment of Support Vector Regression, Artificial Neural Networks, and Random Forests for Predicting and Mapping Soil Organic Carbon Stocks across an Afrotropical Landscape. *Ecol Indic* **2015**, *52*, 394–403, doi:10.1016/J.ECOLIND.2014.12.028.
141. Song, J.; Gao, J.; Zhang, Y.; Li, F.; Man, W.; Liu, M.; Wang, J.; Li, M.; Zheng, H.; Yang, X.; et al. Estimation of Soil Organic Carbon Content in Coastal Wetlands with Measured VIS-NIR Spectroscopy Using Optimized Support Vector Machines and Random Forests. *Remote Sensing* **2022**, *Vol. 14*, Page 4372 **2022**, *14*, 4372, doi:10.3390/RS14174372.
142. de Santana, F.B.; Otani, S.K.; de Souza, A.M.; Poppi, R.J. Comparison of PLS and SVM Models for Soil Organic Matter and Particle Size Using Vis-NIR Spectral Libraries. *Geoderma Regional* **2021**, *27*, e00436, doi:10.1016/J.GEODRS.2021.E00436.
143. Wilson, M.D. Support Vector Machines. *Encyclopedia of Ecology, Five-Volume Set* **2008**, 1–5, 3431–3437, doi:10.1016/B978-008045405-4.00168-3.
144. Pouladi, N.; Møller, A.B.; Tabatabai, S.; Greve, M.H. Mapping Soil Organic Matter Contents at Field Level with Cubist, Random Forest and Kriging. *Geoderma* **2019**, *342*, 85–92, doi:10.1016/J.GEODERMA.2019.02.019.
145. Grimm, R.; Behrens, T.; Märker, M.; Elsenbeer, H. Soil Organic Carbon Concentrations and Stocks on Barro Colorado Island — Digital Soil Mapping Using Random Forests Analysis. *Geoderma* **2008**, *146*, 102–113, doi:10.1016/J.GEODERMA.2008.05.008.
146. Sanderman, J.; Gholizadeh, A.; Pittaki-Chrysodonta, Z.; Huang, J.; Safanelli, J.L.; Ferguson, R. Transferability of a Large Mid-Infrared Soil Spectral Library between Two Fourier-Transform Infrared Spectrometers. *Soil Science Society of America Journal* **2023**, *87*, 586–599, doi:10.1002/SAJ2.20513.

147. Castaldi, F.; Chabrilat, S.; van Wesemael, B. Sampling Strategies for Soil Property Mapping Using Multispectral Sentinel-2 and Hyperspectral EnMAP Satellite Data. *Remote Sensing* **2019**, *Vol. 11*, Page 309 **2019**, *11*, 309, doi:10.3390/RS11030309.
148. Ward, K.J.; Brell, M.; Spengler, D.; Castaldi, F.; Neumann, C.; Segl, K.; Foerster, S.; Chabrilat, S.; Ward, K.J.; Brell, M.; et al. Mapping Soil Organic Carbon Based on Simulated EnMAP Images and the LUCAS Soil Spectral Library. *EGUGA* **2020**, 3013, doi:10.5194/EGUSPHERE-EGU2020-3013.
149. Ng, W.; Minasny, B.; Montazerolghaem, M.; Padarian, J.; Ferguson, R.; Bailey, S.; McBratney, A.B. Convolutional Neural Network for Simultaneous Prediction of Several Soil Properties Using Visible/near-Infrared, Mid-Infrared, and Their Combined Spectra. *Geoderma* **2019**, *352*, 251–267, doi:10.1016/J.GEODERMA.2019.06.016.
150. Svensen, J.L.; Cheng, X.; Boersma, S.; Sun, C. Chance-Constrained Stochastic MPC of Greenhouse Production Systems with Parametric Uncertainty. *Comput Electron Agric* **2024**, *217*, 108578, doi:10.1016/J.COMPAG.2023.108578.
151. Montoya, A.P.; Guzmán, J.L.; Rodríguez, F.; Sánchez-Molina, J.A. A Hybrid-Controlled Approach for Maintaining Nocturnal Greenhouse Temperature: Simulation Study. *Comput Electron Agric* **2016**, *123*, 116–124, doi:10.1016/J.COMPAG.2016.02.014.
152. Bontsema, J.; Van Henten, E.J.; Gieling, T.H.; Swinkels, G.L.A.M. The Effect of Sensor Errors on Production and Energy Consumption in Greenhouse Horticulture. *Comput Electron Agric* **2011**, *79*, 63–66, doi:10.1016/J.COMPAG.2011.08.008.
153. van Mourik, S.; van Beveren, P.J.M.; López-Cruz, I.L.; van Henten, E.J. Improving Climate Monitoring in Greenhouse Cultivation via Model Based Filtering. *Biosyst Eng* **2019**, *181*, 40–51, doi:10.1016/J.BIOSYSTEMSENG.2019.03.001.
154. Boersma, S.; Van Mourik, S.; Xin, B.; Kootstra, G.; Bustos-Korts, D. Nonlinear Observability Analysis and Joint State and Parameter Estimation in a Lettuce Greenhouse Using Ensemble Kalman Filtering. *IFAC-PapersOnLine* **2022**, *55*, 141–146, doi:10.1016/J.IFACOL.2022.11.129.
155. Mazzocchi, F. Complexity in Biology. Exceeding the Limits of Reductionism and Determinism Using Complexity Theory. *EMBO Rep* **2008**, *9*, 10, doi:10.1038/SJ.EMBOR.7401147.
156. Polotskaya, K.; Muñoz-Valencia, C.S.; Rabasa, A.; Quesada-Rico, J.A.; Orozco-Beltrán, D.; Barber, X. Bayesian Networks for the Diagnosis and Prognosis of Diseases: A Scoping Review. *Mach Learn Knowl Extr* **2024**, *6*, 1243–1262, doi:10.3390/MAKE6020058/S1.
157. Strith, A.; Rabe, S.E.; Robaina, O.; Grêt-Regamey, A.; Celio, E. An Online Platform for Spatial and Iterative Modelling with Bayesian Networks. *Environmental Modelling & Software* **2020**, *127*, 104658, doi:10.1016/J.ENVSOFT.2020.104658.
158. Masaracchia, L.; Fredes, F.; Woolrich, M.W.; Vidaurre, D. Computational Neuroscience: Dissecting Unsupervised Learning through Hidden Markov Modeling in Electrophysiological Data. *J Neurophysiol* **2023**, *130*, 364, doi:10.1152/JN.00054.2023.
159. Mall, P.K.; Singh, P.K.; Srivastav, S.; Narayan, V.; Paprzycki, M.; Jaworska, T.; Ganzha, M. A Comprehensive Review of Deep Neural Networks for Medical Image Processing: Recent Developments and Future Opportunities. *Healthcare Analytics* **2023**, *4*, 100216, doi:10.1016/J.HEALTH.2023.100216.
160. Khanam, F.-T.-Z.; Al-Naji, A.; Chahl, J. Remote Monitoring of Vital Signs in Diverse Non-Clinical and Clinical Scenarios Using Computer Vision Systems: A Review. *Applied Sciences* **2019**, *9*, 4474.
161. Kashiha, M.; Pluk, A.; Bahr, C.; Vranken, E.; Berckmans, D. Development of an Early Warning System for a Broiler House Using Computer Vision. *Biosyst Eng* **2013**, *116*, 36–45.
162. Karmakar, P.; Teng, S.W.; Murshed, M.; Pang, S.; Li, Y.; Lin, H. Crop Monitoring by Multimodal Remote Sensing: A Review. *Remote Sens Appl* **2024**, *33*, 101093, doi:10.1016/J.RSASE.2023.101093.
163. Ghislieri, M.; Cerone, G.L.; Knaflitz, M.; Agostini, V. Long Short-Term Memory (LSTM) Recurrent Neural Network for Muscle Activity Detection. *J Neuroeng Rehabil* **2021**, *18*, 153, doi:10.1186/S12984-021-00945-W.
164. Bargagli Stoffi, F.J.; Cevolani, G.; Gnecco, G. Simple Models in Complex Worlds: Occam's Razor and Statistical Learning Theory. *Minds Mach (Dordr)* **2022**, *32*, 13–42, doi:10.1007/S11023-022-09592-Z/METRICS.
165. Geman, S.; Bienenstock, E.; Doursat, R. Neural Networks and the Bias/Variance Dilemma. *Neural Comput* **1992**, *4*, 1–58, doi:10.1162/NECO.1992.4.1.1.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.