An innovating Statistical Learning Tool based on Partial Differential Equations, intending livestock Data Assimilation

Hélène Flourent, Emmanuel Frénod, Vincent Sincholle

An innovative Statistical Learning Tool based on Partial Differential Equations for livestock Data Assimilation

Hélène Flourent[1,2,4], Emmanuel Frénod[2,3,5], Vincent Sincholle[1]

[1] *NutriX*[6], *France*
[2] *Université Bretagne Sud, Laboratoire de Mathématiques de Bretagne Atlantique, UMR CNRS* 6205*, Campus de Tohannic, Vannes, France*
[3] *See-d,* 6*, rue Henri Becquerel - CP* 101*,* 56038 *Vannes Cedex, France*

Abstract

Realistic modeling for quantifying certain biological mechanisms in a precise manner is a task that requires a large volume of prior knowledge and generally leads to heavy mathematical models. On the other hand, the structure of classical Machine Learning algorithms, such as Neural Networks, have limited ability to take into account the existence of complex underlying phenomena and need a large quantity of data to be fitted.

This paper demonstrates that applied mathematics - and specifically Partial Differential Equations (PDEs) - involved in a mathematical modeling approach, coupled with statistical analyses, can be powerful to build Artificial Intelligence (AI) tools, especially in the Smart Farming sector. Indeed, as we will detail later on, most of the issues addressed in this sector need to strike a compromise between accuracy, flexibility and information extraction. To be more precise, we built a particularly parsimonious parameterized PDE system that nonetheless has the capability to deal with all the dynamics that seem to us necessary to link biological variables. We also design its integration within a statistical edifice. We did it at a high level of genericity to ensure the repeatability of our method for the building of Mathematical Modeling Based AI Tools for other purposes and domains.

We postulated that all the physico-chemical phenomena occurring in animal's body can be summarized by describing an overall information flow. Therefore, the PDE system we built, mathematically translates our hypothesis: It embeds the mathematical expressions of biological determinants and describes the circulation, the evolution and the action of an overall information flow.

The application of our approach on real data in a «Small Data» context, showed that the integration of biological knowledge in a parsimonious model increases forecasting accuracy and reduces the training-data-dependency. Moreover, the learning of the dynamics linking inputs and outputs, confers to the tool the capability to be trained on a given range of data and then to be accurately applied to a wider range of data. This extrapolation capability is a real improvement over existing statistical tools.

keywords : Statistical Learning, PDE, Forecasting, Data Assimilation, Model-Data Coupling, Biological Mathematical Modeling, Artificial Intelligence.

[4]helene.flourent@univ-ubs.fr
[5]emmanuel.frenod@univ-ubs.fr
[6]The compagny wishes to remain anonymous

1 Introduction

According to Vázquez-Cruz et al. [1], there are currently two general approaches for analyzing biological data. The first approach involves realistic models that aim to describe and quantify in a precise manner all the biological processes observed from the injection or the ingestion of a set of molecules to its action somewhere in the living organism. Following studies carried out by Bastianelli et al. [2] and by Martin et al. [3], the construction of realistic models is a task that requires time and a large quantity of biological knowledge and generally leads to unwieldy models containing a large number of equations and parameters. The in-silico experiments allowed by this type of model are valuable for describing and explaining specific biological processes with the use of particular *Inputs*. However, the complex implementation of these models limits their adaptability and flexibility, in particular when it comes to processing or assimilating field data presenting high variability, missing and aberrant values.
The second approach corresponds to «Black Box» models, such as Neural Networks. Over the past decade, the use of Machine Learning (ML) algorithms and especially Neural Networks (NN) has been on the rise [4]. According to some studies ([5], [6], [7] and [8]), the popularity of these tools can be explained by the ease of their implementation and the diversity of issues that these algorithms can handle. Nevertheless, these algorithms are based on relatively simple mathematical models that are cannot easily take into account complex phenomena, such as delay and saturation. Hence, the tools based on these types of ML algorithms contain little prior biological knowledge. It is therefore necessary to learn the parameters of these models from a large amount of data to compensate the absence of prior biological expertise ([9], [10], [11] and [12]).

The goal of this paper is to introduce a new paradigm combining these two kinds of approaches, and demonstrate that applied mathematics can be very pertinent to build Artificial Intelligence (AI) tools to deal with Smart Farming issues. Our first aim was to build a tool that can link *Inputs* and *Outputs* concerning an animal or a group of farm animals to perform simulation and forecasting. The second aim was to build an AI tool that can interpret and synthesize a more or less continuous stream of data collected on farms for Data Assimilation (See [13], [14], [15] et [16]).

To do so, the authors of [17], [18], [19], [1] and [20], explain that it is necessary to develop a mathematical model that can incorporate some aspects of the dynamics of the system under study, corresponding here to the animal body. In the light of the limits of the current methods for predicting biological responses, we explored an approach aiming to construct a tool combining accuracy, parsimony, and flexibility. We designed a biomimetic predictive tool able to deal with the existence of complex underlying phenomena. To be more precised, we built an advanced *Mathematical Model* which is a system of Partial Differential Equations (PDE) that embed the mathematical expression of various biological phenomena (diffusion, convection, accumulation, saturation, *etc.*) and depend on the parameters carrying the learning ability of the tool. We also built the statistical method to implement this learning capability. Then, a sensibility analysis and the identification of qualitative properties of the built model were performed. After those analyses, the optimization routine leading to the fitting of the model to data, brought the effective learning capability.

After this Introduction, putting this research work in its proper context, we will detail in Section 2 the problems encountered in the field of biological modeling. In this particular context, the existing tools are limited and not entirely appropriate for achieving the above-presented objectives. Therefore, in Section 3 we will present the *Mathematical Model* and the built *Statistical Learning Tool*. In Section 4, we will study the functioning of the *Mathematical Model* and the ranges of values of the various parameters. To verify the tool's capacities, several tests were performed. In Section 5 we will present the simulation tests performed to verify the ability of the model to learn parameters from noisy data. Then, in Section 6, an application of our approach to field data concerning the growth of animals during a given period will be presented. This application demonstrates the prediction capability of the tool in real conditions. To obtain an idea of the real potential of this new *Statistical Learning Tool*, we compared our biomimetic model with some Logistic Models, Mechanistic Models and Machine Learning algorithms such as Neural Networks. These comparisons will be detailed in Section 7.

2 Problem description

2.1 Difficulties and challenges related to the construction of AI tools to deal with Smart Farming issues

In their review, Dumas et al.[21] explain that the construction of mathematical models to tackle livestock production issues, began between 1910 and 1925 to predict and simulate processes by integrating field knowledge. Today, mathematical modeling is still a valuable tool to simplify, describe and simulate the mechanisms and the links existing between various factors, especially in biological field studies ([21], [20] and [1]). As it can be identified in [20], [22], [23], [24], [25] and [26], in the agri-food sector, simulating and predicting the effects of nutrition on animal performances are two decisive and strategic goals for breeders and companies to understand how optimize animal efficiency.

However, as illustrated in [27] and [28], databases collected on living organisms generally contain a large amount of variability. Although some of this variability is related to individual differences, there is also noise, generated by the measurements tools, and heterogeneity resulting from the lack of continuity over the various experiments. In addition to this variability, some values may be missing or aberrant. Moreover, data collection on farms comes within an evolving framework. Indeed [29], [30], [31], [32] and [33], present some new technologies which make it possible to directly monitor animals (connected collar, troughs recording feeding behavior, connected scales, boluses, etc.). However, these technologies are still expensive and their spread takes time.

All these elements constrained and guided our modeling approach: we sought to carry-out simulations and Data Assimilation via a light and parsimonious AI tool. This parsimony also makes it possible to quickly adapt our tool to different farm species studied by the agri-food companies. Our choices also led to an AI tool with a high potential for information extraction. This extraction potential aims to make our tool compatible with the complexity of the studied phenomena coupled with the current lack of exploitable data, and in the future, with the large volumes of data that will be

generated from the evolution of the means of farm data collection.

2.2 Exploration of an intermediate approach: Model-Data Coupling

The intermediate approach we explored to build our AI tool, combines the use of knowledge involving the studied system and the use of data to extract complex information from available data. Therefore, our work falls within the Model-Data Coupling theoretical framework. Nowadays, Model-Data Coupling is essentially used in the fields of meteorology (see [34]), hydrology (see [35],[36] and [37]), biogeochemistry (see [38], [39], [40] and [41]) and oceanography (see [42]). Like biology, these fields are domains in which it is necessary to take into account the dynamics of the studied system to make forecasts. However, the system under study is often complex and its exact modeling would take time and result in a complex mathematical model. Therefore, as seen in [43], [44], [45] and [46], this Model-Data Coupling approach consists in building a parsimonious mathematical model, corresponding to a mathematical synthesis of the studied system. The parameters contained in the model are optimized and fitted from data. As in the above-cited studies, the construction of our AI tool is based on an optimal combination between knowledge - to design a *Mathematical Model* presenting the «optimal» degree of complexity - and data - to optimize the model parameters - in order to obtain a predictive tool that is accurate, parsimonious and flexible.

In our approach, particular attention was paid to the construction of the *Mathematical Model*. Indeed, the construction of the model - that is a PDE system - constitutes a central feature of our approach and its design was the key element to achieve our objectives.

The first objective was to obtain a tool with a strong information extraction capability to make it accurate even for small biological datasets, without overfitting the training dataset. To minimize the training-data-dependency of our tool, we postulated that the integration of prior knowledge in the *Mathematical Model* may make the learning process more efficient. Therefore, we integrated knowledge on the dynamics of animal body in the *Mathematical Model*.

The second objective was to have a PDE system sufficiently accurate to perform forecasting, but sufficiently light and efficient to perform Data Assimilation. Therefore, we sought to build a PDE system able to take into account the overall effects of dynamic biological processes occurring in the animal body, without describing all the processes. Our exploration is based on the hypothesis that the synthetic consideration of the biological processes may help gain precision, in comparison with a classical ML tool which integrates no prior knowledge, while keeping a parsimonious and light tool, in comparison with a tool based on a realistic model.

To summarize, our exploration relies on the relationship between several diverse elements (Figure 1).

The *Real Animal* is a complex living organism in which a high number of physical flows and chemical reactions act and interact. Therefore, the used support of reflection is not directly the *Real Animal*, but an *Avatar* of the *Real Animal*. This imagined *Avatar* outlines, in the framework of a specific problem, the dynamics of the biological

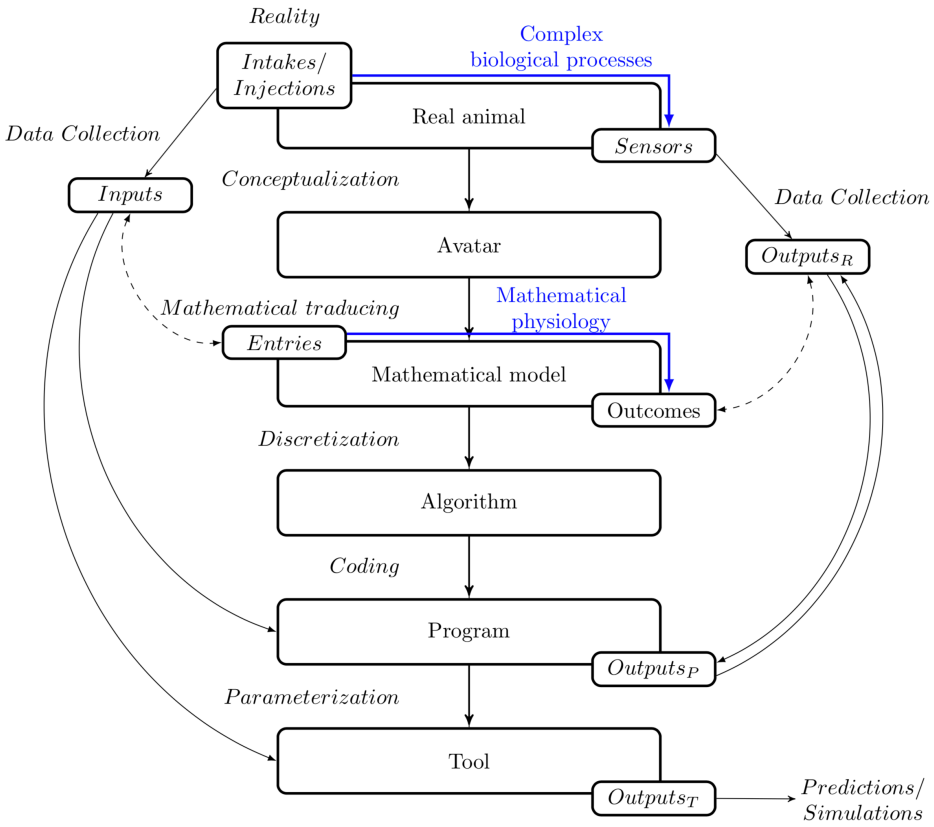

Figure 1: Articulation of the elements of the exploration

reactions occurring in animal body. Here, we outlined all the physico-chemical dynamics, such as the circulation, the evolution and the action of an overall information flow. This information summarizes all the phenomena of convection, diffusion, accumulation, saturation, and delay that a set of molecules may undergo in the body of an animal. The built *Mathematical Model* mathematically translates the evolution and the action of the overall information circulating in the *Avatar*.

Therefore, in our approach we distinguished different dimensions. There is the *Reality* in which there are $Intakes$ and $Injections$ inducing complex biological processes in the animal body. Some *Sensors* extract information from this *Reality*, producing databases made of *Inputs* and *Outputs*. The *Inputs* are translated by a mathematical function into *Entries*, that are pieces of information integrated into the *Mathematical Model* and that induce the generation of *Outcomes*, also linked to the *Outputs* extracted from the *Reality* by a mathematical function.
The *Mathematical Model* has no biological state. The *Real Animal* has a biological condition induced by the introduction of molecules in its body, whereas the model has no biological condition but a physiological-like condition induced by the integration of *Entries* in an involved geometrical space. This mathematical physiological-like status links the *Entries* and the *Outcomes*.

The *Algorithm* comes out of the discretization of the *Mathematical Model*, e.g. the PDE system mathematically translates what takes place in the *Avatar*. This system of PDEs contains parameters corresponding to biological-like factors: convection and

diffusion speeds, some saturation levels, the fixation speed, *etc.* These parameters can be learned from a database and by using optimization algorithms. Therefore, the presence of parameters that can be learned from data in the *Mathematical Model* confers learning ability to the tool based on this model. Hence, the constructed tool is a *Statistical Learning Tool*.
The *Program* corresponds to the code that manages the learning of the parameter values via an iterative training process during which an optimization algorithm finds the values of the parameters that minimize the difference between the measured and the predicted *Outputs*.
The *Tool* finally corresponds to the *Mathematical Model* parameterized with the values of the parameters obtained at the end of the learning step.

3 Structure and discretization of the *Mathematical Model*

3.1 Description of the *Mathematical Model*

Through the conceptualization of the *Avatar*, we set up a parsimonious summary of a given biological process. We hypothesized that, when a molecule or a group of molecules enter the body of a living organism, it circulates in the body through a network of vessels containing a fluid. It integrates this fluid and uses it as a vector to evolve via convection and diffusion mechanisms. In the network of vessels, the molecules may be in competition with other mechanisms which may delay its progression. The circulating molecules may then be captured and accumulated in an organ or a specific tissue. During its storage, the molecules can be used and induce a change in some biological variables.

Then, we built the PDE system, mathematically translating the previously set up summary, as illustrated in Figure 2.

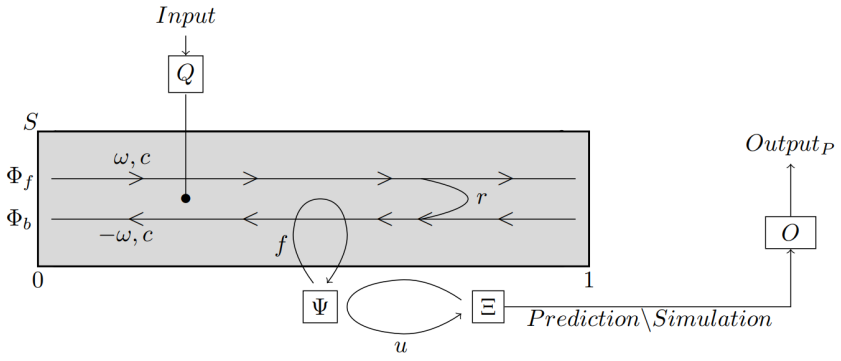

Figure 2: Schematic of the *Mathematical Model*

In concrete terms, we modeled our *Avatar* using variables, densities and fields that are all unitless and dimensionless. We also reduced the geometrical space S relative to the *Avatar* to interval $[0\,;1]$. We considered a Forward Flow Φ_f, and a Backward

Flow Φ_b streaming in this one-dimensional geometrical space. These flows can be seen as a very synthetic summary of a blood circulation, a circulation in the nervous system or a circulation in the digestive tract. The involved *Inputs* basically correspond to the data collected on feed intake, water intake and administration of drugs. These *Inputs* are integrated in the *Mathematical Model* via a function Q transforming these *Inputs* into information inflows, called *Entries*. We modeled that part of the injected information circulates in a forward direction, via Φ_f and the rest circulates backwards, via Φ_b. This information can evolves via convection and diffusion phenomena. We hypothesized that the circulating information can be delayed, captured, stored and used to ultimately induce a modification in the *Outcome* O. This *Outcome* corresponds to the model prediction or simulation of a biological variable.
Therefore, a given piece of information $\mathtt{d}$, is associated with various elements. $\big\{\Phi_f(\mathtt{d})\big\}(t,\mathbf{x})$ and $\big\{\Phi_b(\mathtt{d})\big\}(t,\mathbf{x})$ are, at each instant t, two space densities respectively associated with a forward flux with a velocity $\omega_{\mathtt{d}}$ and a backward flux with a velocity $-\omega_{\mathtt{d}}$.
The spatial density $\big\{\Phi_f(\mathtt{d})\big\}(t,\mathbf{x})$ is supposed to be solution to:

$$\frac{\partial\big\{\Phi_f(\mathtt{d})\big\}}{\partial t}(t,\mathbf{x})+\omega_{\mathtt{d}}\frac{\partial\big\{\Phi_f(\mathtt{d})\big\}}{\partial \mathbf{x}}(t,\mathbf{x})-\frac{\partial\left[c_{\mathtt{d}}\chi\dfrac{\partial\Big[\big\{\Phi_f(\mathtt{d})\big\}+\big\{\Phi_b(\mathtt{d})\big\}\Big]}{\partial\mathbf{x}}\right]}{\partial\mathbf{x}}(t,\mathbf{x})$$
$$=\frac{1}{2}\big\{Q(\mathtt{d})\big\}(t,\mathbf{x})-f_{\mathtt{d}}\big\{F(\mathtt{d})\big\}(\mathbf{x})\big\{\Phi_f(\mathtt{d})\big\}(t,\mathbf{x})-r_{\mathtt{d}}\big\{\Phi_f(\mathtt{d})\big\}(t,\mathbf{x}),\quad(1)$$

Similarly, $\big\{\Phi_b(\mathtt{d})\big\}(t,\mathbf{x})$ is supposed to be solution to:

$$\frac{\partial\big\{\Phi_b(\mathtt{d})\big\}}{\partial t}(t,\mathbf{x})-\omega_{\mathtt{d}}\frac{\partial\big\{\Phi_b(\mathtt{d})\big\}}{\partial \mathbf{x}}(t,\mathbf{x})-\frac{\partial\left[c_{\mathtt{d}}\chi\dfrac{\partial\Big[\big\{\Phi_f(\mathtt{d})\big\}+\big\{\Phi_b(\mathtt{d})\big\}\Big]}{\partial\mathbf{x}}\right]}{\partial\mathbf{x}}(t,\mathbf{x})$$
$$=\frac{1}{2}\big\{Q(\mathtt{d})\big\}(t,\mathbf{x})-f_{\mathtt{d}}\big\{F(\mathtt{d})\big\}(\mathbf{x})\big\{\Phi_b(\mathtt{d})\big\}(t,\mathbf{x})+r_{\mathtt{d}}\big\{\Phi_f(\mathtt{d})\big\}(t,\mathbf{x}),\quad(2)$$

In these equations, the parameter $c_{\mathtt{d}}$ is the diffusion velocity of the information. The space-time density $\big\{Q(\mathtt{d})\big\}$, corresponds to an external source of information. The function $\big\{F(\mathtt{d})\big\}$ is worth 0 in certain areas of the involved geometrical space and 1 in others. The area where this function is worth 1 corresponds to the location of the entity capturing the information. The parameter $f_{\mathtt{d}}$ determines the rate of fixed information. The parameter $r_{\mathtt{d}}$ determines the fraction of the circulating information transferred from the Forward Flow to the Backward Flow, which induces a delay in the progression of the information. At each time t, the spatial density $\big\{\Psi(\mathtt{d})\big\}(t,\mathbf{x})$, associated with the fixed information, is solution to:

$$\frac{\partial\big\{\Psi(\mathtt{d})\big\}}{\partial t}(t,\mathbf{x})=f_{\mathtt{d}}\big\{F(\mathtt{d})\big\}(\mathbf{x})\Big[\big\{\Phi_b(\mathtt{d})\big\}(t,\mathbf{x})+\big\{\Phi_f(\mathtt{d})\big\}(t,\mathbf{x})\Big]-u_{\mathtt{d}}\big\{\Psi(\mathtt{d})\big\}(t,\mathbf{x}).\quad(3)$$

The parameter $u_{\mathtt{d}}$ is the coefficient determining the usage rate of the fixed information. At each time t, the spatial density $\big\{\Xi(\mathtt{d})\big\}(t,\mathbf{x})$, associated with the used information, is solution to:

$$\frac{\partial\big\{\Xi(\mathtt{d})\big\}}{\partial t}(t,\mathbf{x})=u_{\mathtt{d}}\big\{\Psi(\mathtt{d})\big\}(t,\mathbf{x}).\quad(4)$$

The parameter $\Omega(\mathtt{d})$ corresponds to the area of action of the circulating information on the *Outcome*. $\{O(\mathtt{d})\}(t)$ is the *Outcome* of the model, given by:

$$\{O(\mathtt{d})\}(t) = \int_{\Omega(\mathtt{d})} \{\Xi(\mathtt{d})\}(t, \mathbf{x})\, d\mathbf{x}. \tag{5}$$

3.2 The «usage» equation

The fourth equation of the model is the «usage» equation. This equation determines the action of the injected information on the variable to predict. Therefore, this equation has to adapt the different ways in which an intake or an injection may affect a biological variable. Equation (4) models an accumulative process. Hence, it can be used to study data on the total production over a given period.

To model a logistical growth, we added a limiter in this equation. In this case, the «usage» equation becomes:

$$\frac{\partial\{\Xi(\mathtt{d})\}}{\partial t}(t, \mathbf{x}) = u_{\mathtt{d}}\{\Psi(\mathtt{d})\}(t, \mathbf{x}) \left(\frac{L_{\mathtt{d}} - \{O(\mathtt{d})\}(t)}{L_{\mathtt{d}}} \right) \tag{4b}$$

With this version of the equation, data related to the change in weight of an animal can be included. This equation is essentially the differential equation of Verhulst [47]:

$$\frac{\partial y}{\partial t}(t) = r\ y(t) \left(\frac{K - y(t)}{K} \right) \tag{6}$$

whose structure is equivalent. Indeed, in the case when nothing depends on x, the value of $u_{\mathtt{d}}$ is very high and $\Omega(\mathtt{d})$ is the whole interval $[0\ ;\ 1]$, $\{\Xi(\mathtt{d})\}$, $\{\Psi(\mathtt{d})\}$ and $\{O(\mathtt{d})\}$ are very similar. Hence, Equations (6) and (4b) are essentially the same.

It may be also necessary to model variations to use our tool to treat, for example, data on drug effects. To do so, we have to be able to model an increase or a decrease in the *Outcome* which may vary between an upper and a lower bound. Hence, we built two other equations: The equation

$$\frac{\partial\{\Xi(\mathtt{d})\}}{\partial t}(t, \mathbf{x}) = -\Big(\{\Xi(\mathtt{d})\}(t, \mathbf{x}) - Upp_{\mathtt{d}} \Big) - u_{\mathtt{d}}\{\Psi(\mathtt{d})\}(t, \mathbf{x}) \Big(\{\Xi(\mathtt{d})\}(t, \mathbf{x}) - Low_{\mathtt{d}} \Big) \tag{4c}$$

models that the fixed information $\{\Psi(\mathtt{d})\}$ orients the *Outcome* $\{O(\mathtt{d})\}$ toward a state that is lower than the steady state, and the equation

$$\frac{\partial\{\Xi(\mathtt{d})\}}{\partial t}(t, \mathbf{x}) = -u_{\mathtt{d}}\{\Psi(\mathtt{d})\}(t, \mathbf{x}) \Big(\{\Xi(\mathtt{d})\}(t, \mathbf{x}) - Upp_{\mathtt{d}} \Big) - \Big(\{\Xi(\mathtt{d})\}(t, \mathbf{x}) - Low_{\mathtt{d}} \Big) \tag{4d}$$

models that the fixed information $\{\Psi(\mathtt{d})\}$ orients the *Outcome* $\{O(\mathtt{d})\}$ toward a state which is greater than the steady state. In these two cases, the *Outcome* varies between a lower bound $Low_{\mathtt{d}}$ and an upper bound $Upp_{\mathtt{d}}$.

The «usage» equation must be defined according to the issue at hand and the knowledge available on the link existing between the *Entries* and the *Outcomes*.

3.3 Initial and boundary conditions

The function χ is compactly supported in $(0,1)$, mainly constant and worthing 1. This function integrated in the diffusion term makes diffusion vanish at the edges of the domain.

We also imposed :

$$\forall t \in (0,\infty),\ \left\{\Phi_f(\mathtt{d})\right\}(t,0) = \left\{\Phi_b(\mathtt{d})\right\}(t,0) \text{ and } \left\{\Phi_b(\mathtt{d})\right\}(t,1) = \left\{\Phi_f(\mathtt{d})\right\}(t,1) \qquad (7)$$

These conditions allow the circulating information to move back and forth between the two edges of the domain.

The initial conditions $\left\{\Phi_f(\mathtt{d})\right\}(0,\mathbf{x})$, $\left\{\Phi_b(\mathtt{d})\right\}(0,\mathbf{x})$, $\left\{\Psi(\mathtt{d})\right\}(0,\mathbf{x})$, $\left\{\Xi(\mathtt{d})\right\}(0,\mathbf{x})$ and $\left\{O(\mathtt{d})\right\}(0)$ are given for all x in $(0,1)$.

3.4 The model parameters

The system of PDEs contains several parameters that have to be learned: i.e. $\omega_{\mathtt{d}}$, $c_{\mathtt{d}}$, $r_{\mathtt{d}}$, $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$. To simplify, in the first studies we set $c_{\mathtt{d}}$ to 0.001.

All the other parameters are learned from a database by using an optimization algorithm to find the parameter values that minimize the error associated with the model on a training database. To do so, we used the function *directL* developed by Johnson [48], which is embedded in R ([49]) and applying the DIRECT algorithm developed by Finkel [50].

4 Study of the *Mathematical Model* functioning

For the discretization of the *Mathematical Model*, we first used the classical Finite Difference method, with a given space step, to obtain semi-discrete in space equations. And, because the *Mathematical Model* is coded using R software, we used the R-function $Ode.1D$ developed by Soetaert et al.[51] to manage the temporal discretization of the semi-discrete equations. This R-function calls upon the fourth order Runge Kutta method with a given time step (See [52]).

A detailed mathematical analysis of the model and its discretization will be performed in an upcoming paper. Nevertheless, we already know that because we set the discretization steps, the convection and diffusion speeds must follow the CFL conditions and be no larger than the given limits (See [53] and [54]). In this first exploration, to find a compromise between precision and calculation time, we parameterized the mesh with a time step of 0.001 and a space step of 0.025. Hence, ω must be smaller than 25 and c must be smaller than 0.625.

Before launching the model to learn the parameters, we needed to specify lower and upper values for each parameter between which the optimization algorithm will search the values that minimize the error associated with the model. Therefore, a comprehensive study of the ranges of values of the different model parameters was performed and presented in the working paper [55]. We refer to it for the details of this study.

5 Simulation tests of the learning capability of the model

The objective of this Section is to present the tests by simulation, performed to verify the ability of the tool to learn parameters from noisy biological data. To do so we started by generating a fictitious database from our parameterized *Mathematical Model*. Then, we used this database to study the compensation effects existing between the parameters. Finally, we simulated the learning of the parameters from the fictitious data and verified if the model fit the data correctly.

5.1 Generation of a Learning Database

To test the learning capability of the model, we generated a Learning Database containing 50 individuals, that is 50 *Output Curves*. The objective was to obtain a database having the same characteristics as a real field database. To do so, we integrated noise and individual variability in this fictitious database.

The construction of this Learning Database is presented in the first appendix associated with this paper. We refer to it for the details.

Figure 10 shows an example of the generated curves without and with noise. We divided the obtained database into two datasets: A *Training Database* made of 30 curves and a *Test Database* made of 20 curves.

In the rest of this Section, we supposed that we have an experimental-like database and a model containing four parameter values to determine.

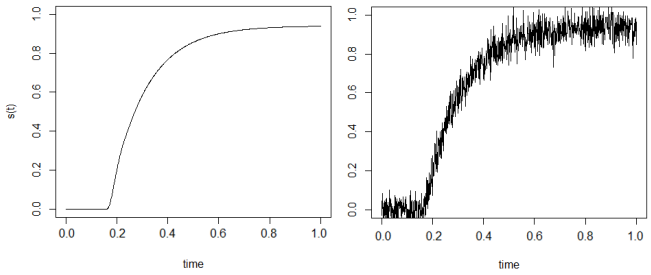

Figure 3: Example of simulated curves without and with noise

5.2 The relations existing between the parameters

A study of the compensation effects existing between $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$ and between $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$, permitted to determine the relationships existing between these two pairs of parameters. This study is presented in the second appendix associated with this paper. We refer to it for the details.

We concluded from this study that, using a Nadaraya-Watson kernel regressions (See [56] and [57]), we obtained a non-parametric relationship linking $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$ in the form of,

$$r_{\mathtt{d}} = \hat{m_\omega}(\omega_{\mathtt{d}}) + \epsilon_\omega, \tag{8}$$

and we obtained a non-parametric relationship linking $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$ in the form of,

$$u_{\mathtt{d}} = \hat{m_f}(f_{\mathtt{d}}) + \epsilon_f, \tag{9}$$

where $\hat{m_\omega}$ and $\hat{m_f}$ corresponds to the Nadaraya-Watson estimators and ϵ_ω and ϵ_f are the residuals.

Knowing the relationship existing between $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$ and the one existing between $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$, it is possible to fit $\omega_{\mathtt{d}}$ and $f_{\mathtt{d}}$ and then deduce the values of $r_{\mathtt{d}}$ and $u_{\mathtt{d}}$. Hence, these relations permits to reduce the number of parameters to learn simultaneously.

5.3 Parameter fitting and calculation of the model accuracy

We fitted the parameters from the *Training Database* and then we tested the accuracy of the obtained model by calculating the error made on the *Test Database*.

5.3.1 Fitting of $\omega_{\mathtt{d}}$ and $f_{\mathtt{d}}$:

To fit the model on different datasets, we sampled the *Training Database*: we sampled 20 curves from the 30 curves in the *Training Database* and we fitted the parameters from the 20 sampled curves. By proceeding in this manner, we performed 30 fittings of the parameters. To determine the values of $\omega_{\mathtt{d}}$, $r_{\mathtt{d}}$, $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$, we fitted $\omega_{\mathtt{d}}$ and $f_{\mathtt{d}}$ on the selected curves of the *Training Database* and then we deduced the values of $r_{\mathtt{d}}$ and $u_{\mathtt{d}}$.

To optimize the parameters, we used the algorithm DIRECT permitting to find the pair $(\omega_{\mathtt{d}}, f_{\mathtt{d}})$ that minimizes the objective function (10).

$$f_{obj}(\omega, f) = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\left(\frac{(y_{ij_{obs}} - y_{ij_{pred}}(\omega, f))}{y_{ij_{obs}}}\right)^2\right) \tag{10}$$

After 200 iterations, we obtained the values of $\omega_{\mathtt{d}}$ and $f_{\mathtt{d}}$ and deduced the values of $r_{\mathtt{d}}$ and $u_{\mathtt{d}}$.

After the 30 fittings of the parameters, we obtained 30 values of $\omega_{\mathtt{d}}$, $r_{\mathtt{d}}$, $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$. We calculated the mean and the Relative Standard Deviation (RSD) of each parameter (Table 1). We also looked at the fit of the model (Figure 4) and we calculated from the *Training Database* the value of the Determination Coefficient (R^2) of the obtained model (Table 1). The Determination Coefficient is high, indicating that the model fits the curves of the *Training Database* well.

5.3.2 The model accuracy:

To validate the ability of the tool to learn parameters from noisy data, we calculated the accuracy of the model on the *Test Database*. To do so, we calculated the RRSS and the Determination Coefficient associated with each curve contained in the *Test Database* and we obtained the distributions of these indicators (Figure 5). The $RRSS$ is low and the Determination Coefficient is high, indicating that the model fits the curves of the *Test Database* well.

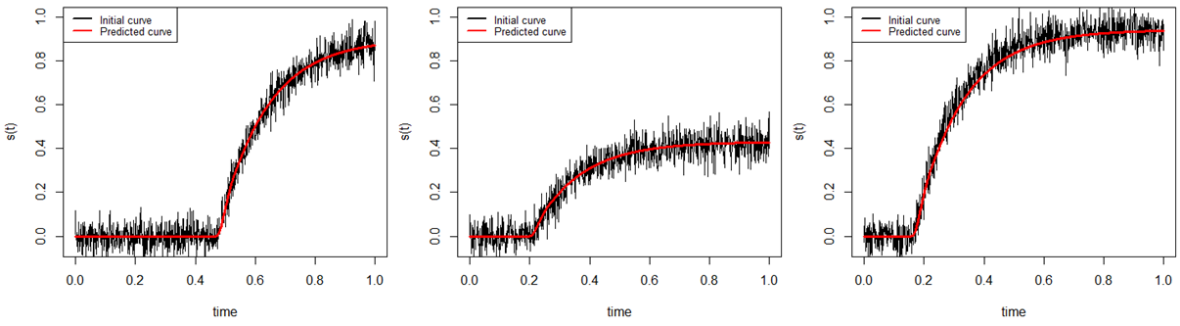

Figure 4: Examples of results

Table 1: Average and Relative Standard Deviation of the parameters and the Determination Coefficient calculated on the *Training Database*.

Parameter	Average	Relative standard deviation
ω	9.9	0.009
f	920.3	0.001
r	35.6	0.016
u	139.5	0.001
R^2	0.97	0.011

We compared the R^2 (R^2_{Gener}) and the $RRSS$ ($RRSS_{Gener}$) associated with the *Generator model* - i.e. the model used to generate the *Learning Database* - and the R^2 (R^2_{Fit}) and the $RRSS$ ($RRSS_{Fit}$) associated with the *Fitted Model* (Figure 5 and Table 2). $RRSS_{Fit}$ is low and this value is very similar to the value of $RRSS_{Gener}$. The R^2_{Fit} is high and this value is also very similar to the value of R^2_{Gener}. These indicators thus demonstrate that the model fitting method is highly satisfactory and the error associated with the adjusted model is limited to the amount of noise and individual differences initially integrated in the generated database.

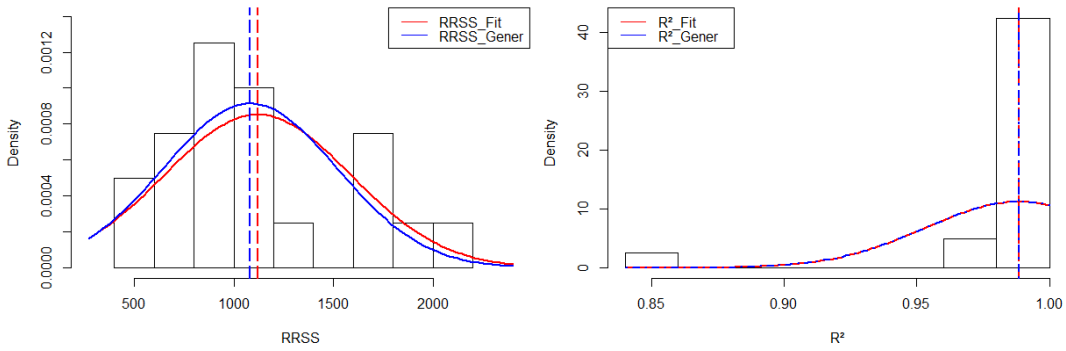

Figure 5: Distributions of the $RRSS$ and of the R^2 coefficient associated with the *Generator Model* and the *Fitted Model*.

Table 2: Comparison between the indicators associated with the *Generator Model* and the *Fitted Model*.

	$RRSS$	R^2
Generator Model	1082	0.9887
Fitted Model	1119	0.9886

6 Application of the *Statistical Learning Tool* on field data

In this section we present an application of our approach to field data. The database we used is confidential therefore only the dimensionless *Inputs* and *Outputs* are given.

6.1 Objectives of this application on field data

The objective of this application is to build a tool that can predict the logistic growth of animals according to an initial weight and intakes.

Adaptations were made to the basic model described in Section 3.1, to adapt it to the dynamics variable to predict.

6.2 Adaptation of the basic model

To mimic logistic behavior, we used Equation (4b) containing a limiter as the «usage» equation. In this equation, $L_{\mathtt{d}}$ corresponds to the maximum value of the variable to predict. Experts have an idea of the maximum value reachable by this variable. Therefore, during model fitting, the value of $L_{\mathtt{d}}$ minimizing the error of the model was screened for across a restricted range of values.

In total, there were five parameters to fit: $\omega_{\mathtt{d}}$, $r_{\mathtt{d}}$, $f_{\mathtt{d}}$, $u_{\mathtt{d}}$ and $L_{\mathtt{d}}$.

6.3 The data used

The database we used is made of two parts corresponding to two different individual groups monitored during two different periods (Table 3). The first group contained 8 individuals, monitored over a unit-period from $t = 0$ until $t = 1$. For this group, the weight of the animals was measured at $t = 0$ and at $t = 1$. The second group contained 7 individuals, monitored from $t = 0$ until $t = 2.5$. For this group the weight of the animals was measured at $t = 0$, $t = 0.6$, $t = 1.52$ and at $t = 2.5$. For both groups, intakes of each individual were recorded over each time-step of 0.16 time-unit. Therefore, for each individual, information relative to those intakes is periodically injected in the model with a time step of 0.16.

The dataset involving the first group constitutes our *Training Database* and the dataset involving the second group constitutes our *Test Database*. The objective was to fit the parameters on the *Training Database* and test the accuracy of the *Fitted Model* on the *Test Database*.

Table 3: Description of the data used.

	First group	Second group
Number of individuals	8	7
Output measured at	$t = 0$ $t = 1$	$t = 0$ $t = 0.60$ $t = 1.52$ $t = 2.50$
Time step of the *Entries* injections	$\Delta t_{In} = 0.16$	$\Delta t_{In} = 0.16$

6.4 Study of the relationships between the model parameters

As in the second appendix associated with this paper, we analyzed the relationships between the model parameters by applying the same methodology on the field *Training Database*. We refer to it for the details of this study.

Using a Nadaraya-Watson kernel regression, we obtained a non-parametric regression linking $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$ and using an other Nadaraya-Watson kernel regression, we obtained a non-parametric regression linking $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$. Knowing the relationships between these parameters, it is possible to fit $\omega_{\mathtt{d}}$ and $f_{\mathtt{d}}$ and then deduce the value of $r_{\mathtt{d}}$ and $u_{\mathtt{d}}$. Hence, these relations reduce the number of parameters to learn simultaneously.

6.5 Parameter fitting

We only fitted $\omega_{\mathtt{d}}$, $f_{\mathtt{d}}$ and $L_{\mathtt{d}}$ and deduced the values of $r_{\mathtt{d}}$ and $u_{\mathtt{d}}$.

The parameters were fitted on the *Training Database* by minimizing the difference between the simulated and the real *Outputs* at time $t = 1$. To fit the parameters, we used the algorithm DIRECT that minimized the following objective function:

$$f_{obj}(\omega_{\mathtt{d}}, r_{\mathtt{d}}, f_{\mathtt{d}}, u_{\mathtt{d}}, L_{\mathtt{d}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(s_{i_{obs}}(1) - s_{j_{pred}}(1))}{s_{i_{obs}}(1)}\right)^2, \tag{11}$$

where n is the number of individuals and $O_{i_{obs}}(1)$ and $O_{i_{pred}}(1)$ correspond respectively to the value of the observed and the predicted *Output* value for the i^{th} individual at $t = 1$.

To test the stability of the set of values of the parameters minimizing the error of the model on the *Training Database*, we performed several fitting procedures. To do so, we sampled the *Training Database*: we randomly selected 7 individuals from the 8 individuals before each fitting procedure and we fitted the parameters on the data associated with the selected individuals. Therefore, we performed 8 fittings and we obtained 8 sets of values of $(\omega_{\mathtt{d}}, r_{\mathtt{d}}, f_{\mathtt{d}}, u_{\mathtt{d}}, L_{\mathtt{d}})$.

6.6 Results

We calculated the average and the Relative Standard Deviation (RSD) of each parameter (Table 4). The RSD of each parameter is low, indicating that our fitting method

permited to identify one set containing the parameter values that minimize the error associated with the *Fitted Model*. The existence of a single optimal set of values of $(\omega_{\mathtt{d}}, r_{\mathtt{d}}, f_{\mathtt{d}}, u_{\mathtt{d}}, L_{\mathtt{d}})$ attests to the identifiability of the model.

We then parameterized the model with the average values of the parameters.

We calculated the error associated with the model on the *Training Database*. To do so, we calculated the Average Relative Error (ARE) between the measured and predicted values of the *Output* at time $t = 1$ (12).

$$ARE(t) = \frac{1}{n} \sum_{i=1}^{n} \sqrt{\left(\frac{(s_{i_{obs}}(t) - s_{j_{pred}}(1))}{s_{i_{obs}}(t)} \right)^2} \tag{12}$$

Table 4: Average values and Relative Standard Deviation (RSD) of the adjusted parameters. ARE calculated at time $t = 1$ on the *Training Database*.

Parameter	*Mean*	*RSD*
ω	9.24	0.079
r	17.91	0.14
f	707.01	0.36
u	21.49	0.17
$L_{\mathtt{d}}$	1.70	0.009
$ARE(1)$ (%)	1.83	0.013

The ARE value calculated on the *Training Database* at time $t = 1$, is worth 1.83%. This result is satisfactory, but the accuracy of the model must be calculated on a *Test Database* to ensure that the model does not overfit the training data.

To do so, we calculated the ARE on the *Test Database* at time $t = 0.6$, $t = 1.52$ and $t = 2.5$ (Table 5 and Figure 6).

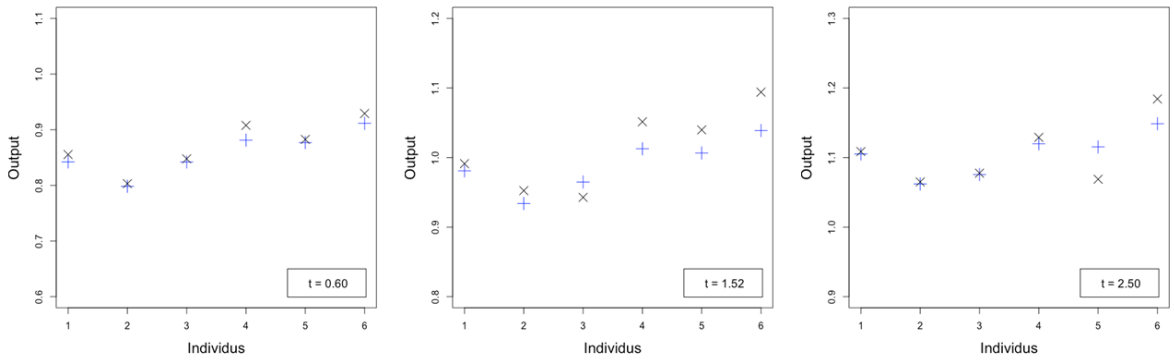

Figure 6: Difference obtained between the measured (+) and predicted (×) values of the *Output* variable at different times t for the individuals in the *Test Database*.

Table 5: Average Relative Error (*ARE*) calculated on the *Test Database* at different instants.

t	0.6	1.52	2.5
ARE(t) (%)	1.3	2.9	1.5

6.7 Discussion of the results

The error associated with the model is low on the *Test Database*. The errors made before and after time $t = 1$ remain low, indicating that the model is able to learn the dynamics linking the inputs and the outputs over a certain period and remain appropriate over a period 2.5 times longer than the training period. Therefore, being based on a parsimonious model, the tool has, in addition to the anticipated interpolation capability, an extrapolation capability.

This extrapolation capability illustrates that our tools holds high potential for information extraction. As it will be demonstrated below, this capability distinguishes our approach from other inference methods.

Moreover, in addition to the information extraction potential, the extrapolation capability helps to reduce the training-data-dependency of our tool. Indeed, this extrapolation capability permits to fit the tool using time series that are defined over shorter and distinct periods. Hence, the tool can be fitted by using a smaller *Training Database*. Therefore, this extrapolation capability permits to reduce the duration of data collection, the duration of in situ experiments, and thus the computational and the experimental costs.

7 Comparison with existing growth models

According to [1] and [58], the current methods used to simulate and predict logistic growth processes, involve two main types of models: Phenomenological Models corresponding to the «Black Box» models, and Mechanistic Models corresponding to the «White Box» models. In this section, we will compare some models belonging to these two main types of models with our Biomimetic Model presented in this paper.

7.1 The Phenomenological Models

As defined in [1], the Phenomenological Models are «Black Box» models corresponding to direct descriptions of the data. This type of model includes Linear, Multiple Linear and Nonlinear Regressions, as well as Logistic Models and Neuronal Networks. We compared our Biomimetic Model with Logistic Models and Neuronal Networks.

7.1.1 Comparison between the Biomimetic Growth Model with Classical Logistic Growth Models

The models of Gompertz [59],

$$\frac{dN(t)}{dt} = a_G.N(t).\ln\left(\frac{K_G}{N(t)}\right), \tag{13}$$

and Verhulst [47],

$$\frac{dN(t)}{dt} = a_V.N(t).\left(1 - \frac{N(t)}{K_V}\right), \tag{14}$$

are two models frequently used to model growth processes (e.g. see [60], [61], [62], [63], [64] and [65]). The models built by Gompertz and Verhulst are based on the hypothesis that growth processes are bounded respectively by K_G and K_V.

We fitted the parameters of the Gompertz's and the Verhulst's models on our *Training Database* by using the same optimization algorithm that we used to fit the Biomimetic Model. These two classical models are also fitted by minimizing $ARE(1)$, as the Biomimetic Model.

On the *Training Database*, the Biomimetic Model is associated with the highest accuracy (Table 6), but the accuracy of all three models is globally similar on this dataset.

To test and compare the accuracy of the different models, we calculated on the *Test Database* the Average Relative Accuracy, ARA (15) at different times t :

$$ARA = 1 - ARE \tag{15}$$

To do so, we used the three parameterized models to generate the growth curve of each individual contained in the *Test Database* and we compared the measured and the predicted values at times $t = 0.6$, $t = 1.52$, $t = 2.5$.
The results contained in Table 6 and the curves of Figures 7 and 8 show that the curves generated from the Gompertz's model featured a premature deceleration. However, the Verhulst's model is associated with a good accuracy over the whole studied period.

The similarity of the results from the Verhulst and the Biomimetic Growth Models was expected, because our model includes an equation assimilable to the Verhulst's equation (see Section 3.2). The real advantage of our biomimetic growth model is its ability to integrate input data. Indeed the Verhulst equation only takes into account the initial conditions of the system under study, whereas our model also integrates *Inputs* throughout the studied period. The integration of additional information appears to help refine of the results and increase the accuracy of the model.

7.1.2 Comparison between the Biomimetic Growth Model and Neural Networks

We applied different Neural Networks on our *Training Database* to compare this kind of ML tool and our Biomimetic Growth Model. We tested six Neural Networks having different numbers of nodes and hidden layers (Table 7), taken the initial state of each individual and their periodically recorded intakes as *Inputs*.

Table 6: Parameter values and $ARA(1)$ calculated on the *Training Database*.

Model	a	K	ARA(1)
Gompertz	$a_G = 0.412$	$K_G = 0.563$	0.978
Verhulst	$a_V = 0.411$	$K_V = 1.563$	0.979
Biomimetic			0.981

t	Verhulst	Gompertz	Biomimetic
0.6	0.985	0.980	0.986
1.52	0.968	0.937	0.971
2.5	0.979	0.923	0.985

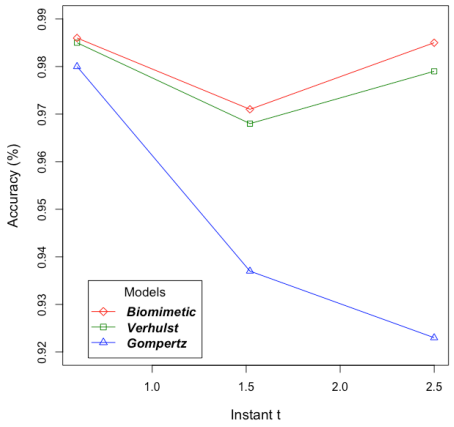

Figure 7: The *ARA* calculated on the *Test Database* at different times associated with different models.

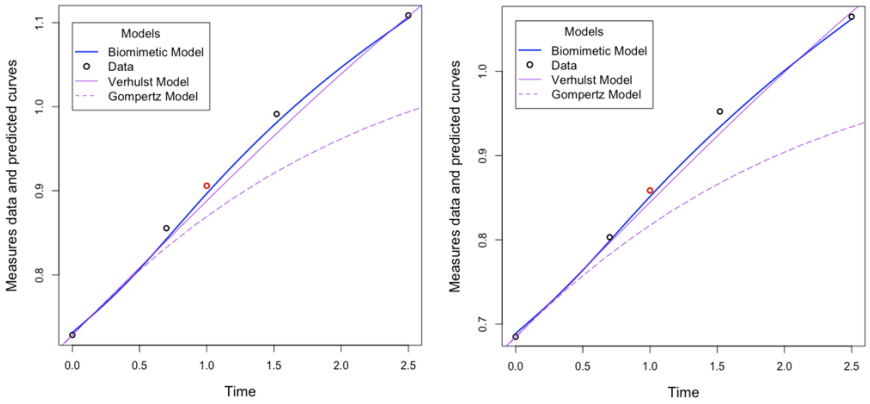

Figure 8: Plot of the predicted growth curves of two individuals contained in the *Test Database* with the different models.

We fitted each tested Neural Network on our *Training Database* by using the R-function *neuralnet* developed by Fritsch et al.[66], and we calculated the accuracy of those Neural Networks on the *Training* and on the *Test Database*.

The results given in Table 7 show that all the tested Neural Network fit the curves of the *Training Database* very well, but not the curves of the *Test Database*. These Neural Network thus overfit the training curves, particularly when the structure of the studied Neural Networks is composed of too many or too few nodes and hidden layers. The accuracy of the Neural Networks on the *Test Database* increases up to a certain

Table 7: The *ARA* calculated on the *Training Database* (ARA_{Train}), and on the *Test Database* (ARA_{Test}), at $t = 1$, with different Neural Networks. The Neural Network (k_1,...,k_i,...,k_n) corresponds to a Neural Network containing n hidden layers and the i^{th} hidden layer contains k_i nodes.

Structure	$ARA_{Train}(1)$ (%)	$ARA_{Test}(1)$ (%)
(4)	99.9	78.8
(4,3)	99.8	90.5
(6,5)	99.7	93.4
(4,6,6,3)	99.9	94.8
(5,7,7,7,4)	99.8	95.3
(5,9,9,9,5)	99.9	93

number of nodes and hidden layers and then decreases when the complexity of the structure continues to increase. The highest accuracy value is reached using a Neural Networks containing 5 hidden layers, but this value is lower than that obtained using our Biomimetic Model (Table 7).

Therefore, in the framework of this study of a generally well-known process with few available data, Neural Networks overfit the training curves and are less accurate than our Biomimetic Model.

Nevertheless, the accuracy of these ML tools is satisfactory and the real advantage of our Biomimetic Model over Neural Networks does not arise from its prediction ability. Indeed, as the Biomimetic Model, the studied Neural Networks were fitted only from the value of the *Output* at $t = 1$. In this case, the fitted classical Neural Networks can only be used to predict the *Output* at $t = 1$. Hence, the Neural Networks cannot interpolate or extrapolate, in contrast to our the Biomimetic Model.

7.2 Mechanistic Growth Models

Some Mechanistic specific and complex Growth Models have been developed in [67], [68], [69] and [19]. These models integrate numerous *Inputs*, and not all of which are available in our study. Hence, these models can not be applied to our database. Therefore, we only compared the structure, the functioning and the objectives of Mechanistic Explanatory Models with our Biomimetic Model.

Similar to our Biomimetic Model, the goal of Mechanistic Growth Models is to integrate existing knowledge in a mathematical model, but more with the purpose of building realistic and explanatory model rather than to perform Data Assimilation. Although, as it is said in [1], [70], [2] and [71], these Mechanistic Growth Models remain abstractions of reality, they are used to perform quantitative analysis in the framework of very specific process studies. Therefore, the construction of those models generally focuses on the biological meaning of the overall model. This objective explains the need to take into account the dynamics of the system under study with higher precision.

Therefore, the construction of the explanatory mechanistic models takes time, requires a large quantity of zootechnical knowledge and results in complex models. As it is explained in [72], [2] and [73], these models contain a large number of unknown

parameters and include many factors, forcing the user to enter a large number of *Input* values, which are sometimes difficult or costly to obtain. Hence, the complex structure of these models makes Mechanistic Realistic Models inappropriate for fitting data and Data Assimilation.

Therefore, the structure of those two types of models are very different, but both are appropriate in light of the respective objectives of these modeling methods.

8 Conclusion

To conclude, we built a *Biomimetic Statistical Learning Tool* based on a PDE system embedding the mathematical expression of biological determinants. The performed tests and the application on field data showed that this tool has a satisfactory accuracy.

In the particular «Biological Small Data» context, our *Statistical Learning Tool* has a higher accuracy than the other tested tools.

However, our *Biomimetic Statistical Learning Tool* really distinguishes itself from the existing tools in light of its interpolation and its extrapolation capacities, its flexibility and its capacity for Data Assimilation.

The pursuit of an optimal combination between the use of data and the use of prior knowledge led to the construction of an Artificial Intelligence tool having a strong learning ability and a weak Training-Data-Dependency. Therefore, our approach appears to hold promise for building more accurate and flexible Artificial Intelligence tools to predict biological variables and to reduce the costs related to the data collection and thus experiments.

As a matter of fact, our approach can be repeated to develop other AI tools for Smart Farming, animal biology and beyond. This is the reason why we chose to work at a high genericity level. Hence, this paper can be seen as a guide-book to build Mathematical Modeling Based AI Tools.

Nevertheless, the results coming from the Biomimetic Model was obtained from a certain number of hypotheses. Some Model Selection methods could be applied to select the structure and the abstraction level of the *Mathematical Model*, permitting to obtain a more satisfying model in terms of ARE and the number of parameters to learn. Which will be studied in a forthcoming work.

Acknowledgements

The authors are very grateful to D. Causeur, G. Durrieu and E. Fokoué for fruitful discussions related to this article.

References

[1] M. A. Vázquez-Cruz, A. Espinosa-Calderón, A. R. Jiménez-Sánchez, and R. Guzmán-Cruz. "Mathematical Modeling of Biosystems". In: *Biosystems Engineering: Biofactories for Food Production in the Century XXI*. Cham: Springer International Publishing, 2014, pp. 51–76. DOI: 10.1007/978-3-319-03880-3_2. URL: https://doi.org/10.1007/978-3-319-03880-3_2.

[2] D. Bastianelli and D. Sauvant. "Modelling the mechanisms of pig growth." In: *Livestock Production Science* (1997).

[3] O. Martin and D. Sauvant. "A teleonomic model describing performance (body, milk and intake) during growth and over repeated reproductive cycles throughout the lifespan of dairy cattle. 1. Trajectories of life function priorities and genetic scaling." In: *Animal* (2010).

[4] P. Domingos. "A Few Useful Things to Know About Machine Learning". In: *Commun. ACM* 55 (2012), 78–87. DOI: 10.1145/2347736.2347755.

[5] M. T. Gorczyca, H. F. M. Milan, A. S. C. Maia, and K. G. Gebremedhin. "Machine learning algorithms to predict core, skin, and hair-coat temperatures of piglets". In: *Computers and Electronics in Agriculture* 151 (2018), pp. 286 –294. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2018.06.028. URL: http://www.sciencedirect.com/science/article/pii/S0168169918303612.

[6] J. J. Valletta, C. Torney, M. Kings, A. Thornton, and J. Madden. "Applications of machine learning in animal behaviour studies". In: *Animal Behaviour* 124 (2017), pp. 203 –220. ISSN: 0003-3472. DOI: https://doi.org/10.1016/j.anbehav.2016.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0003347216303360.

[7] C. Ma, H. H. Zhang, and X. Wang. "Machine learning for Big Data analytics in plants". In: *Trends in Plant Science* 19.12 (2014), pp. 798 –808. ISSN: 1360-1385. DOI: https://doi.org/10.1016/j.tplants.2014.08.004. URL: http://www.sciencedirect.com/science/article/pii/S1360138514002192.

[8] R. H. L. Ip, L. M. Ang, K. P. Seng, J. C. B., and J. E. Pratley. "Big data and machine learning for crop protection". In: *Computers and Electronics in Agriculture* 151 (2018), pp. 376 –383. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2018.06.008. URL: http://www.sciencedirect.com/science/article/pii/S0168169917314588.

[9] A. C. Tan and D. Gilbert. "An empirical comparison of supervised machine learning techniques in bioinformatics". In: *Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003-Volume 19*. Australian Computer Society, Inc. 2003, pp. 219–222.

[10] J. Shavlik, L. Hunter, and D. Searls. "Introduction". In: *Machine Learning* 21.1 (1995), pp. 5–9. ISSN: 1573-0565. DOI: 10.1007/BF00993376. URL: https://doi.org/10.1007/BF00993376.

[11] T. Hubbard and A. Reinhardt. "Using neural networks for prediction of the subcellular location of proteins". In: *Nucleic Acids Research* 26.9 (1998), pp. 2230–2236. ISSN: 0305-1048. DOI: 10.1093/nar/26.9.2230. eprint: http://oup.prod.sis.lan/nar/article-pdf/26/9/2230/9471729/26-9-2230.pdf. URL: https://dx.doi.org/10.1093/nar/26.9.2230.

[12] S. H. Dumpala, R. Chakraborty, and S. K. Kopparapu. *k-FFNN: A priori knowledge infused Feed-forward Neural Networks*. 2017. arXiv: 1704.07055 [cs.LG].

[13] D. Auroux and J. Blum. "Back and forth nudging algorithm for data assimilation problems". In: *Comptes Rendus Mathematique* 340.12 (2005), pp. 873–878.

[14] W. W. Gregg, M. AM. Friedrichs, A. R. Robinson, K. A. Rose, R. Schlitzer, K. R. Thompson, and S. C. Doney. "Skill assessment in ocean biological data assimilation". In: *Journal of Marine Systems* 76.1-2 (2009), pp. 16–33.

[15] R. Lguensat, P. Tandeo, P. Ailliot, M. Pulido, and R. Fablet. "The analog data assimilation". In: *Monthly Weather Review* 145.10 (2017), pp. 4093–4107.

[16] R. Lguensat, P. H. Viet, M. Sun, G. Chen, T. Fenglin, B. Chapron, and R. Fablet. "Data-driven Interpolation of Sea Level Anomalies using Analog Data Assimilation". In: *Remote Sensing* 11.7 (2019), p. 858.

[17] L. J. Renzullo, D. J. Barrett, A. S. Marks, M. J. Hill, J. P. Guerschman, Q. Mu, and S. W. Running. "Multi-sensor model-data fusion for estimation of hydrologic and energy flux parameters". In: *Remote Sensing of Environment* 112.4 (2008). Remote Sensing Data Assimilation Special Issue, pp. 1306 –1319. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2007.06.022. URL: http://www.sciencedirect.com/science/article/pii/S0034425707003227.

[18] B. Ingalls. "Mathematical Modelling in Systems Biology: An Introduction". In: (2019).

[19] E. C. T. Zúñiga, I. L. L. Cruz, and A. R. García. "Parameter estimation for crop growth model using evolutionary and bio-inspired algorithms". In: *Applied Soft Computing* 23 (2014), pp. 474 –482. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2014.06.023. URL: http://www.sciencedirect.com/science/article/pii/S156849461400297X.

[20] M. McPhee. "Mathematical modelling in agricultural systems : A case study of modelling fat deposition in beef cattle for research and industry". In: 2009.

[21] A. Dumas, J. Dijkstra, and J. France. "Mathematical modelling in animal nutrition: A centenary review". In: *Journal of Agricultural Science 146 (2008) 2* 146 (2008). DOI: 10.1017/S0021859608007703.

[22] L. Puillet, O. Martin, D. Sauvant, and M. Tichit. "Introducing efficiency into the analysis of individual lifetime performance variability: a key to assess herd management". In: *animal* 5.1 (2011), pp. 123–133. DOI: 10.1017/S175173111000162X. URL: https://hal.archives-ouvertes.fr/hal-01137029.

[23] O. Martin and D. Sauvant. "A teleonomic model describing performance (body, milk and intake) during growth and over repeated reproductive cycles throughout the lifespan of dairy cattle. 2. Voluntary intake and energy partitioning". In: *Animal* 4.12 (2010), 2048ñ2056. DOI: 10.1017/S1751731110001369.

[24] J. D. Nkrumah, J. Basarab, Z. Wang, C. Li, M. Price, E. Okine, D. H. Crews, and S. S. Moore. "Genetic and phenotypic relationships of feed intake and measures of efficiency with growth and carcass merit of beef cattle". In: *Journal of animal science* 85 (2007), pp. 2711–20. DOI: 10.2527/jas.2006-767.

[25] H. Nesetrilova. "Multiphasic growth models for cattle". In: *Czech Journal of Animal Science* 50 (2005), pp. 347–354. DOI: 10.17221/4176-CJAS.

[26] J. Basarab, M. Price, J. L. Aalhus, E. Okine, W. M. Snelling, and K. L. Lyle. "Residual Feed intake and body composition in young growing cattle". In: *Canadian Journal of Animal Science* 83 (2003), pp. 189–204. DOI: 10.4141/A02-065.

[27] J. C. W. Locke, A. J. Millar, and M. S. Turner. "Modelling genetic networks with noisy and varied experimental data: the circadian clock in Arabidopsis thaliana." In: *Journal of theoretical biology* 234 3 (2005), pp. 383–93.

[28] Y. Qi, Z. Bar-Joseph, and J. Klein-Seetharaman. "Evaluation of different biological data and computational classification methods for use in protein interaction prediction". In: *Proteins: Structure, Function, and Bioinformatics* 63.3 (2006), pp. 490–500. DOI: 10.1002/prot.20865. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.20865. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.20865.

[29] J. S. Jemila and S. S. Priyadharsini. "A Sensor-Based Forage Monitoring of Grazing Cattle in Dairy Farming". In: *International Journal on Smart Sensing and Intelligent Systems* 11 (2018), pp. 1–9. DOI: 10.21307/ijssis-2018-014.

[30] B. Miekley, I. Traulsen, and J. Krieter. "Detection of mastitis and lameness in dairy cows using wavelet analysis". In: *Livestock Science* 148.3 (2012), pp. 227 –236. ISSN: 1871-1413. DOI: https://doi.org/10.1016/j.livsci.2012.06.010. URL: http://www.sciencedirect.com/science/article/pii/S187114131200220X.

[31] R. van der Tol and A. van der Kamp. "Time series analysis of live weight as health indicator". In: *Proceedings of the first North American conference precision dairy management*. 2010, pp. 230–231.

[32] S. Büchel and A. Sundrum. "Short communication: Decrease in rumination time as an indicator of the onset of calving". In: *Journal of Dairy Science* 97.5 (2014), pp. 3120 –3127. ISSN: 0022-0302. DOI: https://doi.org/10.3168/jds.2013-7613. URL: http://www.sciencedirect.com/science/article/pii/S0022030214001684.

[33] A. Holman, J. Thompson, J. E. Routly, J. Cameron, D. N. Jones, D. Grove-White, R. F. Smith, and H. Dobson. "Comparison of oestrus detection methods in dairy cattle". In: *Veterinary Record* 169.2 (2011), pp. 47–47. ISSN: 0042-4900. DOI: 10.1136/vr.d2344. eprint: https://veterinaryrecord.bmj.com/content/169/2/47.full.pdf. URL: https://veterinaryrecord.bmj.com/content/169/2/47.

[34] A. J. Simmons and A. Hollingsworth. "Some aspects of the improvement in skill of numerical weather prediction". In: *Quarterly Journal of the Royal Meteorological Society* 128.580 (2002), pp. 647–677. DOI: 10.1256/003590002321042135. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1256/003590002321042135.

[35] G. Kim and A. P. Barros. "Space–time characterization of soil moisture from passive microwave remotely sensed imagery and ancillary data". In: *Remote Sensing of Environment* 81.2 (2002), pp. 393 –403. ISSN: 0034-4257. DOI: https://doi.org/10.1016/S0034-4257(02)00014-7. URL: http://www.sciencedirect.com/science/article/pii/S0034425702000147.

[36] W. L. Crosson, C. A. Laymon, R. Inguva, and M. P. Schamschula. "Assimilating remote sensing data in a surface flux–soil moisture model". In: *Hydrological Processes* 16 (2002), pp. 1645–1662. DOI: 10.1002/hyp.1051.

[37] D. S. Mackay, S. Samanta, R. R. Nemani, and L. E. Band. "Multi-objective parameter estimation for simulating canopy transpiration in forested watersheds". In: *Journal of Hydrology* 277.3 (2003), pp. 230 –247. ISSN: 0022-1694. DOI: https://doi.org/10.1016/S0022-1694(03)00130-6. URL: http://www.sciencedirect.com/science/article/pii/S0022169403001306.

[38] D. J. Barrett. "Steady state turnover time of carbon in the Australian terrestrial biosphere". In: *Global Biogeochemical Cycles* 16.4 (2002), pp. 55–1–55–21. DOI: 10.1029/2002GB001860. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2002GB001860. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002GB001860.

[39] D. Barrett, M. Hill, L. Hutley, J. Beringer, J. H. Xu, G. Cook, J. Carter, and R. J. Williams. "Prospects for improving savanna biophysical models by using multiple-constraints model-data assimilation methods". In: *Australian Journal of Botany* 53(7) (2005). DOI: 10.1071/BT04139.

[40] P. J. Rayner, M. Scholze, W. Knorr, T. Kaminski, R. Giering, and H. Widmann. "Two decades of terrestrial carbon fluxes from a carbon cycle data assimilation system (CCDAS)". In: *Global Biogeochemical Cycles* 19.2 (2005). DOI: 10.1029/2004GB002254. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2004GB002254. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2004GB002254.

[41] W. J. Sacks, D. S. Schimel, R. K. Monson, and B. H. Braswell. "Model-data synthesis of diurnal and seasonal CO2 fluxes at Niwot Ridge, Colorado". In: *Global Change Biology* 12.2 (2006), pp. 240–259. DOI: 10.1111/j.1365-2486.2005.01059.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2486.2005.01059.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2486.2005.01059.x.

[42] P. Ailliot, E. Frénod, and V. Monbet. "Long term object drift in the ocean with tide and wind." In: *SIAM Journal on Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal* 5 (2) (2006), pp. 514 –531. URL: https://hal.archives-ouvertes.fr/hal-00129093.

[43] E. Frénod. "A PDE-like Toy-Model of Territory Working". In: *Understanding Interactions in Complex Systems - Toward a Science of Interaction*. Understanding Interactions in Complex Systems - Toward a Science of Interaction. Cambridge Scholar Publishing, 2017, pp. 37–47. URL: https://hal.archives-ouvertes.fr/hal-00817522.

[44] A. Rousseau and M. Nodet. "Modélisation mathématique et assimilation de données pour les sciences de l'environnement". In: *Bulletin de l'APMED* 505 (2013), pp. 467–472.

[45] W. J. Sacks, D. S. Schimel, and R. K. Monson. "Coupling between carbon cycling and climate in a high-elevation, subalpine forest: a model-data fusion analysis". en. In: *Oecologia* 151.1 (2007), pp. 54–68. ISSN: 0029-8549, 1432-1939. DOI: 10.1007/s00442-006-0565-2. URL: http://link.springer.com/10.1007/s00442-006-0565-2 (visited on 11/22/2018).

[46] L. Wang, H. Zhang, K. C. L. Wong, H. Liu, and P. Shi. "Physiological-model-constrained noninvasive reconstruction of volumetric myocardial transmembrane potentials". In: *IEEE Transactions on Biomedical Engineering* 57.2 (2010), pp. 296–315.

[47] P. F. Verhulst. "Notice sur la loi que la population suit dans son accroissement". In: *Corresp. Math. Phys.* 10 (1838), pp. 113–126. URL: https://ci.nii.ac.jp/naid/10015246307/en/.

[48] S. G. Johnson. "The NLopt nonlinear-optimization package". 2008.

[49] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2014. URL: http://www.R-project.org/.

[50] D. E. Finkel. *DIRECT Optimization Algorithm*. North Carolina State University, 2003.

[51] K. Soetaert, T. Petzoldt, and R. Woodrow Setzer. "Solving Differential Equations in R: Package deSolve". In: *Journal of Statistical Software* 33.9 (2010), pp. 1–25. ISSN: 1548-7660. DOI: 10.18637/jss.v033.i09. URL: http://www.jstatsoft.org/v33/i09.

[52] W. Enright. "The Numerical Analysis of Ordinary Differential Equations: Runge Kutta and General Linear Methods". In: *SIAM Review* 31.4 (1989), pp. 693–693. DOI: 10.1137/1031147. eprint: https://doi.org/10.1137/1031147. URL: https://doi.org/10.1137/1031147.

[53] R. Courant, K. Friedrichs, and H. Lewy. "Über die partiellen Differenzengleichungen der mathematischen Physik". In: *Mathematische annalen* 100.1 (1928), pp. 32–74.

[54] E. W. Weisstein. "Courant-friedrichs-lewy condition". In: *Wolfram MathWorld–A Wolfram Web Resource.* (2014).

[55] H. Flourent. "Study of the ranges of values of a Biomimetic Statistical Learning Tool parameters". Working paper. 2019. URL: https://hal.archives-ouvertes.fr/hal-02067374.

[56] E. A. Nadaraya. "On estimating regression". In: *Theory of Probability & Its Applications* 9.1 (1964), pp. 141–142.

[57] G. S. Watson. "Smooth regression analysis". In: *Sankhyā: The Indian Journal of Statistics, Series A* (1964), pp. 359–372.

[58] R. Guzmán-Cruz, R. Castaneda-Miranda, J. Garcia-Escalante, L. Solis-Sánchez, D. Alaniz-Lumbreras, J. Mendoza-Jasso, A. Lara-Herrera, G. Ornelas-Vargas, E. Gonzalez-Ramirez, and R. Montoya-Zamora. "Evolutionary Algorithms in Modelling of Biosystems". In: 2011. ISBN: 978-953-307-171-8. DOI: 10.5772/16231.

[59] B. Gompertz. "XXIV. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. In a letter to Francis Baily, Esq. FRS &c". In: *Philosophical transactions of the Royal Society of London* 115 (1825), pp. 513–583.

[60] C. P. Winsor. "The Gompertz Curve as a Growth Curve". In: *Proceedings of the National Academy of Sciences* 18.1 (1932), pp. 1–8. ISSN: 0027-8424. DOI: 10.1073/pnas.18.1.1. eprint: https://www.pnas.org/content/18/1/1.full.pdf. URL: https://www.pnas.org/content/18/1/1.

[61] N. K. Sakomura, F. A. Longo, E. O. Oviedo-Rondon, C. Boa-Viagem, and A. Ferraudo. "Modeling energy utilization and growth parameter description for broiler chickens". In: *Poultry Science* 84.9 (2005), pp. 1363–1369.

[62] J. Buyse, B. Geypens, R. D. Malheiros, V. M. Moraes, Q. Swennen, and E. Decuypere. "Assessment of age-related glucose oxidation rates of broiler chickens by using stable isotopes". In: *Life sciences* 75.18 (2004), pp. 2245–2255.

[63] T. B. Robertson. "Experimental studies on growth II. The normal growth of the white mouse". In: *Journal of Biological Chemistry* 24.3 (1916), pp. 363–383.

[64] T. B. Robertson. *The chemical basis of growth and senescence*. JB Lippincott Company, 1923.

[65] P. Román-Román and F. Torres-Ruiz. "Modelling logistic growth by a new diffusion process: Application to biological systems". In: *Biosystems* 110.1 (2012), pp. 9–21.

[66] S. Fritsch, F. Guenther, and Ma. Suling. *neuralnet: Training of neural networks*. R package version 1.32. 2012. URL: http://CRAN.R-project.org/package=neuralnet.

[67] D. Bastianelli, D. Sauvant, and A. Rerat. "Mathematical modeling of digestion and nutrient absorption in pigs." In: *Journal of animal science* (1996).

[68] J. Mach and Z. Kristkova. "Modelling The Cattle Breeding Production in the Czech Republic". In: *AGRIS on-line Papers in Economics and Informatics* 2 (2010).

[69] L. Brun-Lafleur, E. Cutullic, P. Faverdin, L. Delaby, and C. Disenhaus. "An individual reproduction model sensitive to milk yield and body condition in Holstein dairy cows". In: *Animal : an international journal of animal bioscience* 7 (2013), pp. 1–12. DOI: 10.1017/S1751731113000335.

[70] L. O. Tedeschi, D. G. Fox, R. D. Sainz, L. G. Barioni, S. R. de Medeiros, and C. Boin. "Mathematical models in ruminant nutrition". en. In: *Scientia Agricola* 62 (2005), pp. 76 –91. ISSN: 0103-9016. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-90162005000100015&nrm=iso.

[71] D. E. Beever, A. J. Rook, J. France, M. S. Dhanoa, and M. Gill. "A review of empirical and mechanistic models of lactational performance by the dairy cow". In: *Livestock Production Science* 29.2 (1991), pp. 115 –130. ISSN: 0301-6226. DOI: https://doi.org/10.1016/0301-6226(91)90061-T. URL: http://www.sciencedirect.com/science/article/pii/030162269190061T.

[72] D. Wallach, B. Goffinet, J. E. Bergez, P. Debaeke, D. Leenhardt, and J. N. Aubertot. "Parameter estimation for crop models". In: *Agronomy journal* 93.4 (2001), pp. 757–766.

[73] G. C. Emmans. "Problems in applying models in practice". In: *Publication-European Association For Animal Production* 78 (1995), pp. 223–223.

Appendix I: Generation of the Learning Database

In order to test the learning capability of the model we generated a Learning Database containing 50 individuals, that is 50 *Output Curves*. The objective is to obtain a database having the same characteristics as a real field database. To do that we integrated in this fictitious database noise and individual variability.

I.1 Integration of individual variability

The model parameters are constants to determine. Nevertheless, to introduce individual variability in the generated data, we considered (only in this Section) the parameters as biological-like factors following a Normal distribution. To simulate individual differences, we assigned to each parameter a Normal distribution centered on an arbitrarily chosen value and with a relative variance of 0.005 (See Table 8). From these Normal probability laws, we generated 50 values of the parameters $\omega_{\mathtt{d}}$, $r_{\mathtt{d}}$, $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$. Their respective statistical and probabilistic distributions are given in Figure 9.

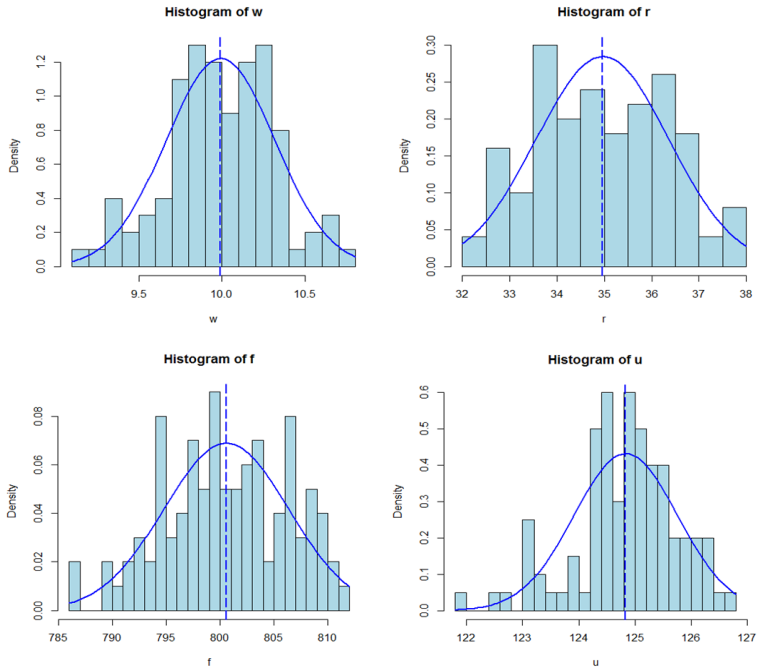

Figure 9: Distributions of the parameters ω, r, f and u

I.2 Generation of fictitious *Inputs*

The *Inputs* integrated in the model correspond to the injected volume ($VolQ$) and the moment of the injection (c_t). These parameters can take on any value between 0 and 1, therefore we applied a Uniform distribution over the interval $[0; 1]$ to these two types of *Inputs* (Table 8).

From the values of the parameters and the fictitious *Inputs*, we generated 50 *Output Curves*.

I.3 Addition of a random noise

Continuing with the objective of obtaining an experimental-like database, we added noise to the *Output Curves*. To do so, we added a random component following a Gaussian distribution centered on 0 and with a variance of 0.05 to the generated curves (Table 8).

Figure 10 shows some examples of generated curves without and with noise. We divided the obtained database into two datasets: A *Training Database* made of 30 curves and a *Test Database* made of 20 curves.

In the rest of this Section, we assumed that we have an experimental-like database and a model containing four parameter values to determine.

Table 8: The distributions followed by the parameters and the *Inputs*.

Parameter	Probability law
ω	$\mathcal{N}(10, 0.3125)$
r	$\mathcal{N}(35, 1.42)$
f	$\mathcal{N}(800, 5.175)$
u	$\mathcal{N}(125, 1)$
$VolQ$	$\mathcal{U}(0, 1)$
c_t	$\mathcal{U}(0, 1)$
$Noise$	$\mathcal{N}(0, 0.05)$

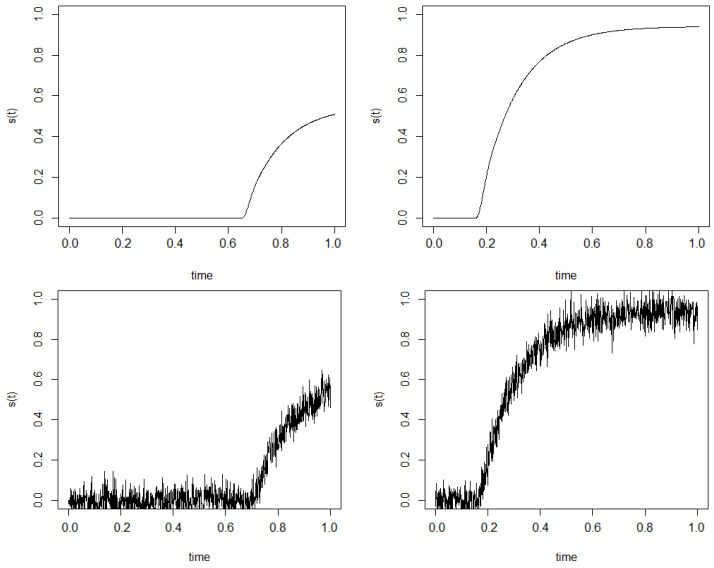

Figure 10: Example of simulated curves without and with noise

Appendix II: Study of the compensation effects existing between the model parameters

Among the parameters $\omega_{\mathtt{d}}$, $r_{\mathtt{d}}$, $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$, some parameters offset each other.

Velocity $\omega_{\mathtt{d}}$, can be offset by any delay $r_{\mathtt{d}}$, the information undergoes. For example, a low convection speed associated with a low delay may induce kinetics equivalent to that induced by a high convection speed associated with a long delay.

The fixation $f_{\mathtt{d}}$, and the use of the information $u_{\mathtt{d}}$, are also two counterbalanced processes. For instance, a high fixation rate followed by a low usage of the information can induce the same effect on the *Outcome* as a low fixation rate followed by a high use of the fixed information.

Therefore, relations exist between the parameters of those two couples. The objective of this part is to use the fictitious *Training Database* to study these relations.

II.1 Study of the relationship between $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$

First, we demonstrated the relationship existing between $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$ by calculating the error made on the *Training Database* by the model parametrized with different $(\omega_{\mathtt{d}}, r_{\mathtt{d}})$ pairs. To do so, we ranged the domain $\omega_{\mathtt{d}} \times r_{\mathtt{d}}$ and we calculated the Relative Residual Sum of Squares ($RRSS$) (16) associated with the models parametrized with different tested $(\omega_{\mathtt{d}}, r_{\mathtt{d}})$ pairs:

$$RRSS(\omega_{\mathtt{d}}, r_{\mathtt{d}}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \left(\frac{(y_{ij_{obs}} - y_{ij_{pred}}(\omega_{\mathtt{d}}, r_{\mathtt{d}}))}{y_{ij_{obs}}} \right)^2 \right), \tag{16}$$

where n corresponds to the number of individuals contained in the *Training Database* and m the number of points on the curves. $y_{ij_{obs}}$ and $y_{ij_{pred}}$ correspond respectively to the observed and the predicted value of the j^{th} point of the i^{th} individual. Therefore $RRSS$ corresponds to the sum of the squared relative differences between the predicted curves and the initially generated curves.

Figures 11 and 12 give the values of the RRSS according to the values of $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$. The existence of a series of equivalent pairs - that is a series of pairs leading to the same value of $RRSS$ - can be seen in Figure 11(a). There is an area where the $RRSS$ values are lower (Figure 12), and corresponding to the curve $EC1$ in Figure 11(b). We assumed that the optimal ($\omega_{d_{Opt}}$,$r_{d_{Opt}}$) pair, inducing the lowest RRSS, belongs to this curve. Therefore, we set out to determine the equation of the curve $EC1$.

II.2 Search for the $(\omega_{d_{Opt}}, r_{d_{Opt}})$ pairs inducing the lowest $RRSS$

To find the equation of the curve $EC1$ we sought for different values of $\omega_{\mathtt{d}}$, the value of $r_{\mathtt{d}}$ minimizing the $RRSS$ value. To do that, for each tested value of $\omega_{\mathtt{d}}$ we used

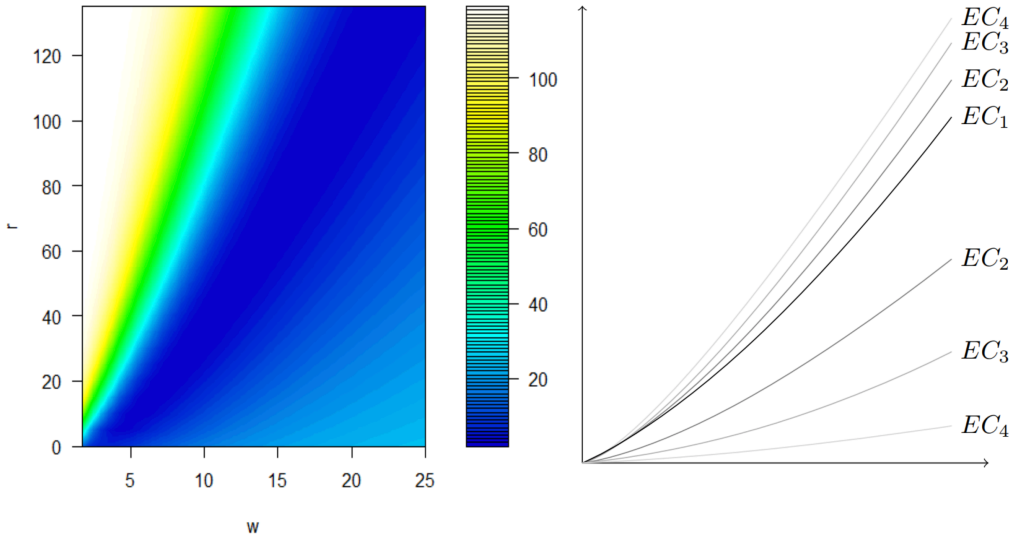

Figure 11: The value of the $RRSS$ according to $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$ (a: left), and the schema of the different Equivalent Couples (EC) (b: right)

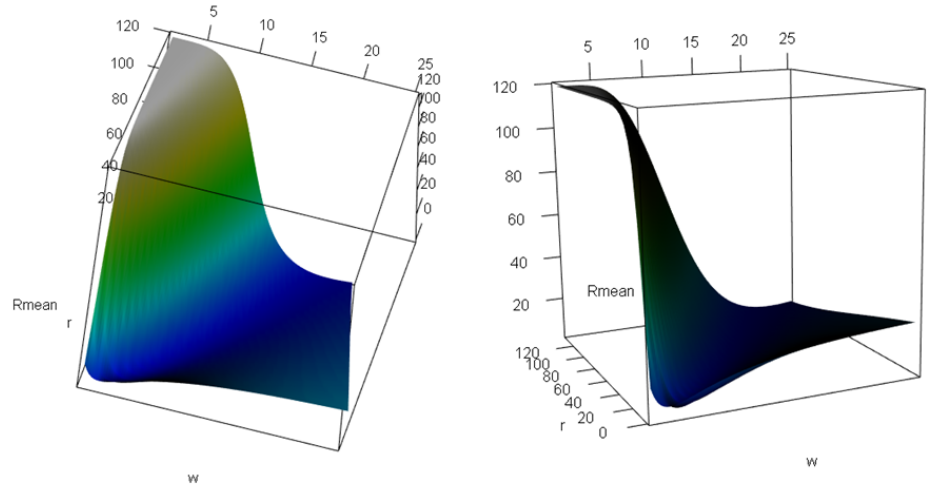

Figure 12: The $3D$ representation of the value of the $RRSS$ according to $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$

the optimization algorithm DIRECT to find the value of $r_{\mathtt{d}}$ minimizing the objective function,

$$f_{obj}(r_{\mathtt{d}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\left(\frac{(y_{ij_{obs}} - y_{ij_{pred}}(\omega_{\mathtt{d}}, r_{\mathtt{d}}))}{y_{ij_{obs}}}\right)^2\right), \tag{17}$$

corresponding to the average RRSS.

To obtain several fitted values of $r_{\mathtt{d}}$ for each tested value of $\omega_{\mathtt{d}}$, we sampled the *Training Database*: we sampled 20 curves from the 30 test curves and we fitted $r_{\mathtt{d}}$ on those 20 selected curves. We ultimately obtained three values of $r_{\mathtt{d}}$ for each tested value of $\omega_{\mathtt{d}}$ (Figure 13). Using a Nadaraya-Watson kernel regression (See [56] and [57]), we obtained a non-parametric relationship linking $\omega_{d_{Opt}}$ and $r_{d_{Opt}}$ in the form of:

$$r_{opt} = \hat{m}(\omega_{opt}) + \epsilon, \tag{18}$$

where $\hat{m}$ corresponds to the Nadaraya-Watson estimator.

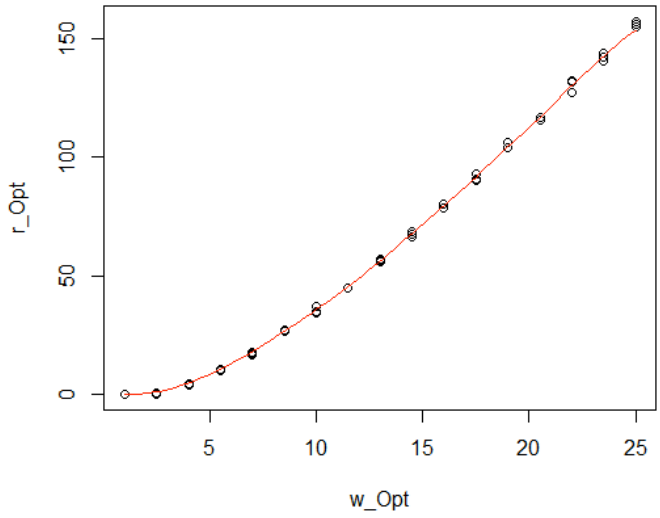

Figure 13: The Nadaraya-Watson kernel regression linking the $(\omega_{d_{Opt}}, r_{d_{Opt}})$ parameter pair.

Knowing the relationship between $\omega_{d_{Opt}}$ and $r_{d_{Opt}}$, it is possible to deduce one of these two parameters according to the value of the other parameter. Hence, this relationship reduces the number of parameters that need to be learned simultaneously.

II.3 Study of the relationship between $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$

There is also a compensation effect between $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$: a high value of $f_{\mathtt{d}}$ can be compensated by a low value of $u_{\mathtt{d}}$, and vice versa.

As above for $\omega_{\mathtt{d}}$ and $r_{\mathtt{d}}$, we sought to determine the relationship existing between $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$ to be able to deduce one of these two parameters according to the other one and further reduce the number of parameters to learn simultaneously.

As above, we ranged the domain $f_{\mathtt{d}} \times u_{\mathtt{d}}$ and calculates the $RRSS$ of the models parameterized with different pairs of values for $(f_{\mathtt{d}}, u_{\mathtt{d}})$ (Figures 14 and 15). This study demonstrates a series of equivalent pairs. There is an area where the $RRSS$ values are lower (Figure 15) and corresponding to the $EC1$ curve in Figure 14(a). We assumed that the optimal $(f_{d_{Opt}}, u_{d_{Opt}})$ pair inducing the lowest $RRSS$, belongs to this curve. Therefore, we set out to determine the equation of this curve.

II.4 Search for the $(f_{d_{Opt}}, u_{d_{Opt}})$ pairs that lead to the lowest $RRSS$

To find the equation of the curve $EC1$ associated with the lowest $RRSS$, we looked for the value of $u_{\mathtt{d}}$ minimizing the $RRSS$ value for different given values of $f_{\mathtt{d}}$. For

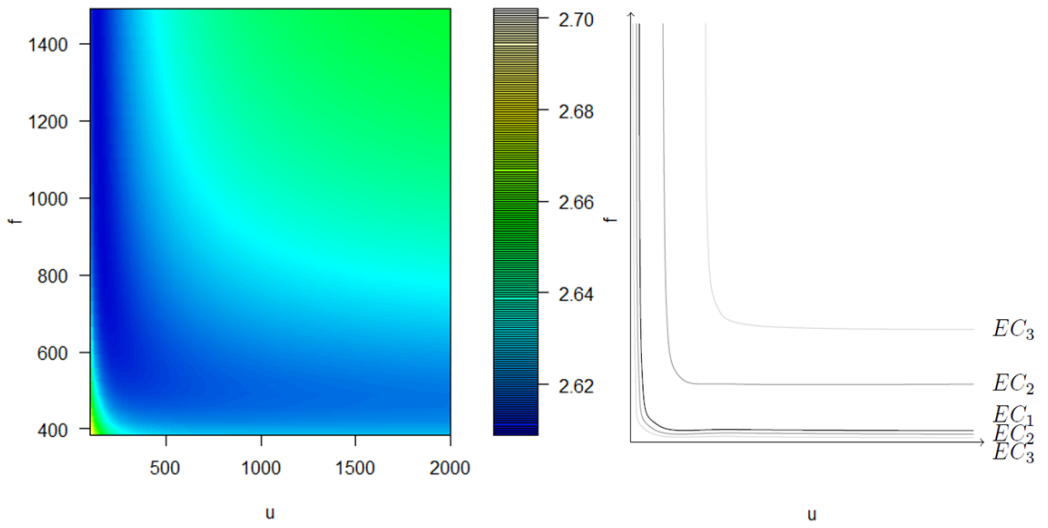

Figure 14: The value of the RRSS according to $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$ (a) and the schema of the different Equivalent Couples (EC) (b)

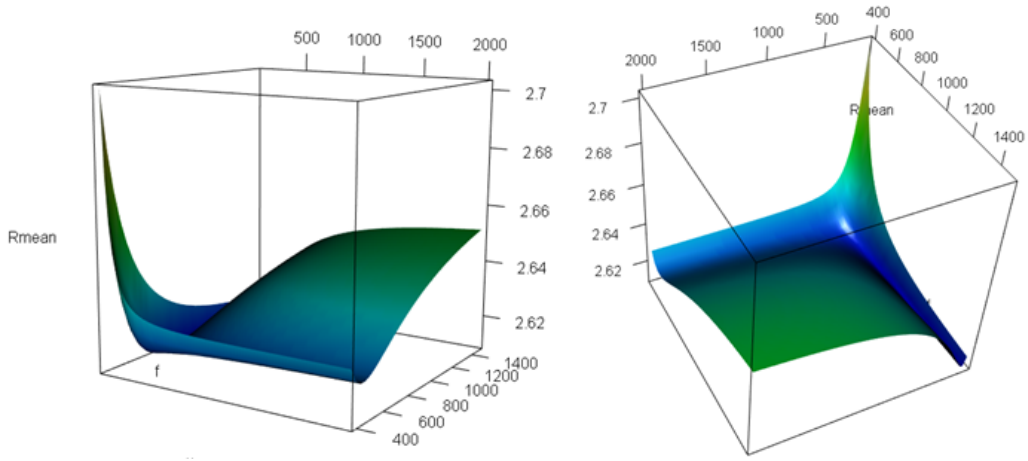

Figure 15: The $3D$ representation of the value of the RRSS according to $f_{\mathtt{d}}$ and $u_{\mathtt{d}}$

each value of $f_{\mathtt{d}}$, we used the optimization algorithm DIRECT to find the value of $u_{\mathtt{d}}$ minimizing the objective function (19) corresponding to the average $RRSS$.

$$f_{obj}(u_{\mathtt{d}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\left(\frac{(y_{ij_{obs}} - y_{ij_{pred}}(f_{\mathtt{d}}, u_{\mathtt{d}}))}{y_{ij_{obs}}}\right)^2\right). \tag{19}$$

As above, to obtain several fitted values of $u_{\mathtt{d}}$ for each tested value of $f_{\mathtt{d}}$, we sampled the *Training Database*. At the end of the fitting, we obtained three values of $u_{\mathtt{d}}$ for each tested value of $f_{\mathtt{d}}$ (Figure 16). Using a Nadaraya-Watson kernel regression, we obtained a non-parametric relationship linking $f_{d_{Opt}}$ and $u_{d_{Opt}}$ in the form of:

$$u_{d_{Opt}} = \hat{m}(f_{d_{Opt}}) + \epsilon, \tag{20}$$

where $\hat{m}$ corresponds to the Nadaraya-Watson estimator.

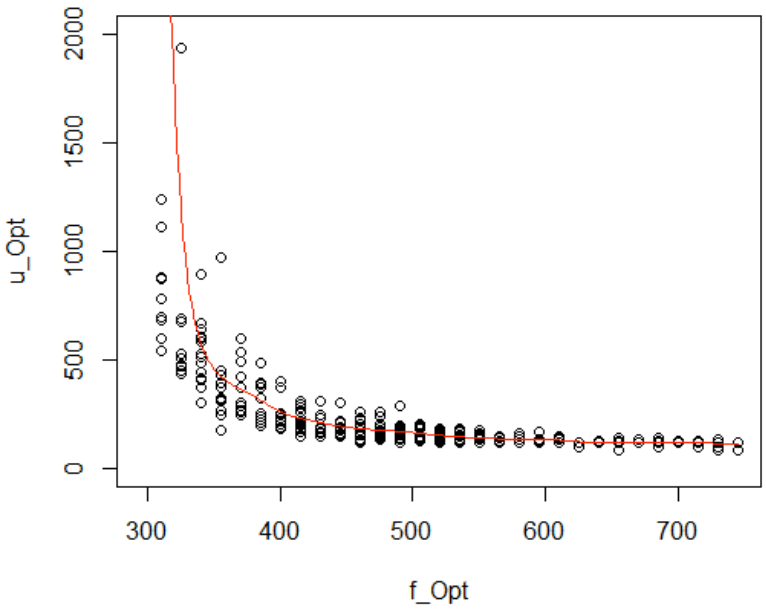

Figure 16: The Nadaraya-Watson kernel regression linking $f_{d_{Opt}}$ and $u_{d_{Opt}}$.

Knowing the relationship existing between $f_{d_{Opt}}$ and $u_{d_{Opt}}$, it is possible to deduce one of these two parameters according to the value of the other parameter. Hence, this relationship further reduces the number of parameters to learn simultaneously.