H2020-SFS-2018-2020

DECIDE

Data-driven control and prioritisation of non-EU-regulated contagious animal diseases

Deliverable D2.6

Report on all developed software tools

WP2 – Methods for data analysis and modelling

Authors	Anders Ringgaard Kristensen (UCPH) Sébastien Picault (INRAE) Dan Børge Jensen (UCPH)
Lead participant	UCPH
Delivery date	11 July 2025
Dissemination level	Public
Type	Report

Revision History

Author Name (Partner short name)	Description	Date
Anders Ringgaard Kristensen (UCPH) Sébastien Picault (INRAE) Dan Børge Jensen (UCPH)	Draft deliverable	10.06.2025
Jasmeet Kaler (UoN)	Revision 1	26.06.2025
Anders Ringgaard Kristensen (UCPH) Sébastien Picault (INRAE) Dan Børge Jensen (UCPH) Leonardo Victor de Knegt (UCPH)	Final version	09.07.2025
Johannes Ripperger (accelCH)	Final checks and formatting	11.07.2025

Contents

EXECUTIVE SUMMARY 6
1 INTRODUCTION 7
1.1 Brief description of WP2 7
1.2 Overview of the developed software tools 7
1.3 Objective of this report 7
2 OPEN-SOURCE SOFTWARE TOOLS TO PERFORM MULTIVARIATE MONITORING OF TIME SERIES DATA 8
2.1 Modelling technique 8
2.1.1 Model specification 8
2.1.2 Prior information 8
2.1.3 Estimation of variance components 8
2.1.4 Filtering and smoothing 9
2.2 Implementation 9
3 OPEN-SOURCE SOFTWARE TOOLS TO PERFORM MULTILEVEL MONITORING OF TIME SERIES DATA 10
3.1 Modelling technique 10
3.1.1 Model specification 10
3.1.2 Variance components 11
3.1.3 Prior information 11
3.1.4 Estimation of variance components 11
3.1.5 Filtering and smoothing 12
3.2 Implementation 12
4 OPEN-SOURCE MECHANISTIC DISEASE-SPECIFIC SIMULATION MODELS 13
4.1 Methodological approach 13
4.2 Implemented applications 13
5 OPEN-SOURCE INFERENCE ALGORITHM ADAPTABLE TO SPECIFIC CASES 15
5.1 Methodological approach 15
5.2 Demonstrated applications 15
6 WARNING SYSTEMS 16
6.1 Methodological approach 16
6.2 Implementation 17
6.3 Demonstrated performance on calf-respiratory dataset 17

7 OVERALL CONCLUSIONS 18
7.1 Strengths 18
7.2 Limitations 18
7.3 Key take-aways 19

Abbreviations

Abbreviation	Description
DLM	Dynamic Linear Model
DoA	Description of Action
EM	Expectation Maximization
EU	European Union
H2020	Horizon 2020
WP	Work Package

Partner short names

Short name	Organisation
UCPH	Københavns Universitet
INRAE	Institut National de Recherche pour l’Agriculture, l’Alimentation et l’Environement
UoN	University of Nottingham
accelCH	accelopment Schweiz AG

Executive Summary

This Deliverable is a report providing an overview of all the developed software tools in WP2:

- Monitoring models
- Mechanistic simulation models
- Combined models

Objectives of the Deliverable

With the help of this deliverable, the reader will achieve an overview of all the software tools previously submitted as deliverables of WP2.

For each tool, a short description is given and a link is provided to a repository for download of the code and a short tutorial for use of the tool.

Activities

UCPH and INRAE described the tools with more details about the methodological approaches than what was included in the short tutorials accompanying the software deliverables.

Outcome

Since the software deliverables D2.1-D2.5 have been of type “other” (i.e. software code), this report (together with the publications from the work package) is the only place where a more comprehensive description is given.

Next steps

This deliverable completes WP2.

1 Introduction

1.1 Brief description of WP2

The overall objective of WP2 was to develop methods for data analysis and modelling. The methods are developed for the use cases (pilot implementations) of the project and as basis for some of the decision support tools of WP3.

In essence, two kinds of models have been developed:

- Monitoring models where two aspects were studied: Multi-variate models integrating several heterogeneous data streams and multi-level models simultaneously monitoring several levels. The models are also used for early warning.
- Mechanistic simulation models for building biologically meaningful simulation models for use cases and for developing an inference algorithm to enhance data integration into simulation models.

In addition, there has been an ambition of attempting to combine the two kinds of modelling for development of early warning systems.

The monitoring methods have been developed by UCPH whereas INRAE has been responsible for developing the mechanistic simulation methods.

The developed methods have been shared with the other partners through regular workshops and software deliverables.

This deliverable is the last one of WP2 and it provides an overview of all the developed software tools.

1.2 Overview of the developed software tools

The developed software tools consist of the following previous deliverables:

- D2.1. Open-source software tools to perform multivariate monitoring of time series data
- D2.2. Open-source software tools to perform multilevel monitoring of time series data
- D2.3. Open-source mechanistic disease-specific simulation models
- D2.4. Open-source inference algorithm adaptable to specific cases
- D2.5. Warning systems

1.3 Objective of this report

The objective of this report is to provide an overall overview of the software tools developed in WP2. For each tool, a short description is given and a link is provided to a repository for download of the code and a short tutorial for use of the tool.

2 Open-source software tools to perform multivariate monitoring of time series data

2.1 Modelling technique

As described in the DoA, the monitoring models developed in WP2 are state space models. In most cases, they are Dynamic Linear Models (for Gaussian data), even though also Dynamic Generalized Linear Models (for binomial data) have been developed.

2.1.1 Model specification

The tools provided as a deliverable (D2.1) cover DLMs, which are models consisting of an observation equation

$$Y_t = F_t'\theta_t + v_t, \qquad v_t \sim N(0, V_t)$$

linking an observation, Y_t, at time t to an underlying (unobservable) vector of parameters, θ_t, in a linear expression described by the design matrix, F_t'. The observation is, furthermore, assumed to be influenced by a random observation error, v_t, drawn from a (multivariate) normal distribution. If the observation Y_t and the observation error v_t are vectors (i.e. not just scalars), the model is multivariate.

In the DLM, the parameter vector, θ_t, is assumed to evolve over time according to a system equation

$$\theta_t = G_t\theta_{t-1} + w_t, \qquad w_t \sim N(0, W_t)$$

linking the value at time t to the value at time $t - 1$ in a linear expression described by the system matrix G_t. The transition is, furthermore, influenced by an evolution error w_t drawn from a (multivariate) normal distribution.

For a full specification of a model,

- The design matrix F_t',
- The system matrix G_t,
- The observation variance-covariance matrix V_t,
- The evolution variance-covariance matrix W_t

must be known. Typically, the design and system matrices are defined based on domain specific knowledge about the system being monitored, whereas the two variance-covariance matrices are estimated from data.

2.1.2 Prior information

In addition to the matrices, also the prior mean m_0 and the prior variance covariance matrix C_0 must be specified. those values are typically determined from a subset of the available data. Thus,

$$(\theta_0 | D_0) \sim N(m_0, C_0),$$

where $D_0 = \{m_0, C_0\}$ is the initial information set.

2.1.3 Estimation of variance components

As mentioned, the variance components, V_t and W_t, are estimated from data. This is typically done on a subset of the data known as the *learning* set. The standard algorithm for estimation of these matrices is the Expectation Maximization algorithm assuming constant values (i.e. $V_t = V$ and $W_t = W$).

In some cases, it is more convenient to use a discount factor approach for estimation of W_t where it is seen as a fixed fraction (defined by the discount factor) of the total uncertainty in the system.

2.1.4 Filtering and smoothing

Having estimated the variance components, the model is ready for running on new data (referred to as the test data). Denote as $D_n = \{m_0, C_0, Y_1, \dots, Y_t\}$ the set of prior information and all observations until time t. The filtered estimates for the underlying parameter vector given all observations are expressed by a conditional mean m_t and a conditional variance-covariance matrix C_t:

$$(\theta_t | D_t) \sim N(m_t, C_t)$$

The updated values of m_t and C_t are computed by use of the Kalman filter algorithm applied at each time step. They represent the best possible estimates given observations until now.

Assume, that observations of Y_t are available until time T. For retrospective analyses, the Kalman smoother is used to compute the conditional mean vector $\widetilde{m}_t$ and variance-covariance matrix $\tilde{C}_t$ at time $t < T$ given *all* data :

$$(\theta_t | D_T) \sim N(\widetilde{m}_t, \tilde{C}_t)$$

The values $\widetilde{m}_t$ and $\tilde{C}_t$ are also denoted as the smoothed mean and variance-covariance, respectively. The smoothed values are the best possible estimates that can be obtained given the model.

2.2 Implementation

The software tool has been implemented in R and it is shared with the DECIDE consortium through a GitHub repository at

https://github.com/decide-project-eu/WP2-Deliverables

and also on the project internal document storage system (accelCLOUD) at:

https://cloud.accelopment.com/index.php/apps/files/?dir=/DECIDE/Deliverables.

The software implementation consists of a script called *DLM functions - DECIDE deliverable_Resubmission.R*. The script (which is Deliverable 2.1 of DECIDE) implements functions for:

- Estimating (based on the first few observations) initial (prior) information in terms of m_0 and C_0.
- Providing an initial estimate of the observation variance-covariance matrix V_t.
- Estimating the variance components V and W by use of the EM algoritm.
- Alternatively, estimating a discount factor to account for W_t.
- Running the Kalman filter
- Running the Kalman smoother
- Validating the model by analysis of forecast errors

As a help to the project partners, a short report (tutorial) is also available together with an example data set and two example scripts illustrating how to use the functions provided in the deliverable. It is also illustrated how to produce plots.

3 Open-source software tools to perform multilevel monitoring of time series data

3.1 Modelling technique

Multi-level models (also referred to as hierarchical models) are used when several units (animals, pens, herds) organized in a hierarchical structure are being monitored simultaneously. The motivation for using a multi-level approach is to utilize the hierarchical structure directly in order to account for correlations between units and/or to enable monitoring at multiple levels. Examples of such hierarchical structures are country/region/herd or herd/section/pen.

The modelling technique applied to this kind of monitoring is, basically, the same as in the multivariate case. Thus, the framework is dynamic linear models where the hierarchy (i.e. the multiple levels) is expressed in the matrix structure as detailed below.

3.1.1 Model specification

An example of a hierarchical model is when a property (e.g. mortality or performance) is observed simultaneously in several herds clustered in several regions of a country is observed at regular time steps. Consider a simplified framework, where a country consists of only two regions (A and B) each having two production sites (1 and 2). Let y_{rit} denote the observed value at time t for herd i in region r. A direct way of modeling the hierarchy (country/region/herd) is through a model like the following:

$$y_{rit} = \mu_t + \alpha_{rt} + \beta_{rit} + v_{rit}, \qquad v_{rit} \sim N(0, \sigma^2),$$

where μ_t is a dynamic country level, α_{rt} is a dynamic region level, β_{rit} is a dynamic herd level, and v_{rit} is a random observation error. Thus, changes over time can happen at any level (country, region or herd).

Let $y_t = (y_{A1t}, y_{A2t}, y_{B1t}, y_{B2t})'$ and $v_t = (v_{A1t}, v_{A2t}, v_{B1t}, v_{B2t})'$ be vectors of all site level observations and observation errors, respectively, at month t. Then the observation equation can be written in matrix notation as

$$y_t = F_t^{'}\theta_t + v_t, \qquad v_t \sim N(\underline{0}, V),$$

where

$$F_t^{'} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$\theta_t = (\mu_t, \alpha_{At}, \alpha_{Bt}, \beta_{A1t}, \beta_{A2t}, \beta_{B1t}, \beta_{B2t}).$$

The system equation of the hierarchical DLM is

$$\theta_t = G_t\theta_{t-1} + w_t, \qquad w_t \sim N(\underline{0}, W_t),$$

where, in this example (without any particular patterns), the system matrix G_t defaults to the identity matrix

$$G_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

3.1.2 Variance components

The variance-covariance matrices W_t and V, it can in principle be estimated by the EM algorithm as described for the multivariate case. Often, however, there will be hundreds of herds in multiple regions. Thus, the dimension of the matrices will reach prohibitive sizes, and a more structured approach is recommended.

For the observation equation of this model, the hierarchy is already taken into account by seeing an observation as a result of effects at all three levels. It may, however, also be natural to assume that the error terms $v_{A1t}, v_{A2t}, v_{B1t}, v_{B2t}$ are mutually correlated. This is achieved if the individual error term is seen as the sum of three underlying independent errors at country, region and herd level, respectively, so that (where indexes refer to country (c), region (r) and herd (i))

$$v_{rit} = v_{ct} + v_{rt} + v_{it}, \;\; v_{ct} \sim N(0, \sigma_C^2), \;\; v_{rt} \sim N(0, \sigma_R^2), \;\; v_{it} \sim N(0, \sigma_I^2).$$

Assuming independence, the variance-covariance matrix V of v_t has the following form:

$$V = \begin{bmatrix} \sigma_C^2 + \sigma_R^2 + \sigma_I^2 & \sigma_C^2 + \sigma_R^2 & \sigma_C^2 & \sigma_C^2 \\ \sigma_C^2 + \sigma_R^2 & \sigma_C^2 + \sigma_R^2 + \sigma_I^2 & \sigma_C^2 & \sigma_C^2 \\ \sigma_C^2 & \sigma_C^2 & \sigma_C^2 + \sigma_R^2 + \sigma_I^2 & \sigma_C^2 + \sigma_R^2 \\ \sigma_C^2 & \sigma_C^2 & \sigma_C^2 + \sigma_R^2 & \sigma_C^2 + \sigma_R^2 + \sigma_I^2 \end{bmatrix}.$$

For the system variance-covariance matrix, W_t, a component discounting approach is used to calculate it for time t using a technique described by West & Harrison (1997)[1].

A discount factor is a number between 0 and 1. Values close to 0 reflect large system variance and values close to 1 reflect small variance. It is called component discounting because it allows each level (component) to have its own discount factor. Thus, the model specification allows for instance the country and regional levels to fluctuate less than the herd level.

Thus, for a full specification of the variance components of this hierarchical model, six parameters are needed: the three observational variances $\sigma_C^2, \sigma_R^2, \sigma_I^2$ and three discount factors ρ_C, ρ_R, ρ_I.

The six parameters are estimated from a learning data set as the values minimizing the sum of squares of forecast errors.

3.1.3 Prior information

Just as in the usual multivariate case, the prior mean m_0 and the prior variance covariance matrix C_0 must be specified. those values are typically determined from a subset of the available data. Thus,

$$(\theta_0 | D_0) \sim N(m_0, C_0),$$

where $D_0 = \{m_0, C_0\}$ is the initial information set.

3.1.4 Estimation of variance components

The developed tools provide a function for estimating all variances and discount factors as the set of values minimizing the root mean square of the forecast errors. The function makes use of the built-in function `optim` in R.

---

[1] West, M. & J. Harrison. 1997. *Bayesian Forecasting and Dynamic Models.* Second Edition. Springer, New York.

3.1.5 Filtering and smoothing

Filtering and smoothing is performed in the same way as in the multivariate case.

3.2 Implementation

The software tool has been implemented in R and it is shared with the DECIDE consortium through a GitHub repository at

https://github.com/decide-project-eu/WP2-Deliverables

and also on the project internal document storage system (accelCLOUD) at:

https://cloud.accelopment.com/index.php/apps/files/?dir=/DECIDE/Deliverables/Submitted/D2.2_SoftwareHierarchicalMonitoring

The software implementation consists of a script called *Functions - Multi-level DLM.R*. The script (which is Deliverable 2.2 of DECIDE) implements functions for:

- Estimating (based on the first few observations) initial (prior) information in terms of m_0 and C_0.
- Estimating the variances and discount factors defining V and W_t by numerical optimization (the `optim` function)
- Running the Kalman filter
- Running the Kalman smoother
- Validating the model by analysis of forecast errors

As a help to the project partners, a short report (tutorial) is also available together with an example data set and two example scripts illustrating how to use the functions provided in the deliverable. The case-example is the monitoring of mortality in salmon production in Scotland. It is also illustrated how to produce plots.

4 Open-source mechanistic disease-specific simulation models

4.1 Methodological approach

All mechanistic models in DECIDE are written using the EMULSION[2] domain-specific language (YAML) and executed by its open-source stochastic simulation engine. This choice guarantees fully transparent model structure, parameter sets and assumptions, while version control through public Git repositories and Software-Heritage snapshots secures long-term accessibility and citability. Three hands-on workshops and accompanying tutorials trained partners to design, check and refine their own models within this framework, ensuring methodological consistency across diseases and species.

The declarative language of EMULSION keeps the model human- and machine-readable, so veterinarians, economists or data-scientists can audit or tweak assumptions without touching code. Besides, this choice also accelerates the development of potential Decision-Support Tools built upon the mechanistic models, which can be processed by another software tool, PASTE[3], to be transformed into a web application designed for stakeholders (with farm-specific inputs, scenarios, and meaningful outputs). This shortens the prototype-to-policy loop: the same model can be more easily adapted across diseases, species and production systems, coupled to early-warning analytics, or wrapped into decision-support web apps with PASTE in a reduced development time.

4.2 Implemented applications

Eight EMULSION models have been delivered (five finalised, three in development), covering major respiratory pathogens of cattle and pigs and paving the way for poultry models.

Highlights include:

- a family of pathogen-specific BRD models for young beef cattle in multi-batch fattening farms, successively enriched with collective-treatment rules, DGLM-based early warning, and vaccination scenarios. The different BRD models are already exploited to (i) compare batch-organisation rules, (ii) rank collective-treatment triggers — including the new “slope” criterion that cuts both incidence and antimicrobial use more efficiently than conventional thresholds — and (iii) test early-warning coupling with Dynamic GLMs for real-time decisions on-farm. These simulations underpin two decision-support prototypes, Eval’BRD and Connect’BRD, generated with the PASTE toolkit.
- an individual-based model for infections caused by the respiratory pathogen *Mycoplasma hyopneumoniae* in Dutch pig farms, that quantifies health, economic, labour and carbon impacts of alternative interventions. This model will also be adapted to a Spanish context to support farm-level decision tools and environmental impacts of interventions.
- ongoing adaptations of the BRD model to veal calves in Belgium and France
- the forthcoming development of a model of swine influenza and infectious bronchitis in poultry to explore stocking-density, vaccination and biosecurity options

The models can be accessed at the following repositories:

- Multi-batch BRD: https://forge.inrae.fr/dynamo/brd-models/brd-public/-/tree/pathogen-specific-multi-batch
- Treatment criteria: https://forge.inrae.fr/dynamo/brd-models/brd-public/-/tree/PCI_animal_science2024

---

[2] https://doi.org/10.1371/journal.pcbi.1007342

[3] https://doi.org/10.1016/j.prevetmed.2024.106233

- Coupling with DGLMs: https://forge.inrae.fr/dynamo/brd-models/brd-public/-/tree/sorin-merca-2024
- Mitigating co-circulation of BRD pathogens through vaccination: https://forge.inrae.fr/dynamo/brd-models/brd-public/-/tree/coinfection-metaphylaxis
- Veal calves: https://github.com/decide-project-eu/BRD_model_veal_calves
- *M. hyopneumoniae* in pigs: https://github.com/decide-project-eu/Mycoplasma_hyopneumoniae

Calibration status

Parameters related to the infectious processes and pathogens involved, to disease duration, severity and clinical signs, to case detection sensitivity, and to treatment success and vaccine efficacy, were drawn from previous studies published in the literature. Other parameters related to batch rearing and farming practices, to treatment protocols, and to the individual risk levels of developing respiratory diseases were assumed based on expert elicitation. The dynamics of cases was compared to observation data when available, otherwise validated by expert veterinary opinion. To ensure the robustness of key model outcomes (e.g., case occurrence and severity, disease duration, antimicrobial use or vaccination efficacy at individual, batch and farm scale) with regard to the uncertainty of the parameter values, exploratory sensitivity analyses were carried out. As new observation data become available for each pathosystem, they will enable in 2025-2026 a more accurate estimation of the most influential parameters using the ABC-regression and ABC-SMC workflows released in D2.4; these algorithms have already proven their ability to recover plausible posteriors on synthetic BRD data. The veal-calf and pig models explicitly list calibration with available datasets as the next milestone before deployment.

Expected impact

Once fully calibrated, the mechanistic models can deliver:

- evidence-based ranking of control strategies (treatments, vaccination, management) that can be tailored to local herd data, supporting WP3 decision-support tools and antimicrobial-stewardship goals
- transferable modelling assets — shared EMULSION YAML files and open Git/SWH archives — that other WPs or external partners can repurpose for new pathogens or countries, accelerating comparative assessments across Europe
- reduced research-to-practice latency: the PASTE pipeline turns a validated model into a web app, allowing rapid feedback to/from farmers and veterinarians.

In short, while some models are not yet fully calibrated, they already guide experimental design and tool prototyping; the forthcoming inference phase will sustain their decision-support potential for stakeholders across the beef, veal, pig and poultry sectors.

5 Open-source inference algorithm adaptable to specific cases

5.1 Methodological approach

Deliverable D2.4 releases an inference toolbox based on Approximate Bayesian Computation (ABC). Two complementary R packages are documented: BRREWABC, a parallel ABC-Sequential Monte Carlo engine optimised for high-dimension stochastic models, and the well-established abc package for ABC-regression when rapid re-analysis is required. Step-by-step tutorials cover installation, data preparation, tolerance choice, distance metrics and posterior diagnostics, making the workflow accessible to non-statisticians.

Added value

Deploying the BRREWABC (ABC-SMC) and abc (ABC-regression) R packages gives DECIDE partners an off-the-shelf way to turn any stochastic simulator into a statistically calibrated tool without having to derive an explicit likelihood (which is not always feasible). BRREWABC offers parallelised ABC-SMC with on-the-fly tuning and "pause-and-resume" features, so it can tackle high-dimension parameter spaces while keeping runs reproducible and adjustable to hardware availability. Package abc re-uses a reference table of pre-simulated data; once that table exists, new posterior estimates arrive in seconds, which is ideal for rapid re-analysis or sensitivity checks. Together they cover the needs for parameter inference and thus shorten the calibration loop for every model in WP2.

Ease of use for non-modellers

The deliverable comes with fully commented R scripts, helpers and four walk-through case studies, thus requiring only basic skills in the R language to copy-paste a specific simulator function and rely upon the package for parameter estimation.

Model agnosticism

Both packages only expect two ingredients: (i) a function that, given a vector of parameters, returns simulated summary statistics, and (ii) the observed statistics to match. This interface means they can be coupled to any EMULSION model — or indeed to external simulators — simply by wrapping the model run in an R function. The same inference workflow therefore scales from cattle BRD to pig *M. hyopneumoniae* or future poultry models with no change to package internals.

5.2 Demonstrated applications

Four case studies illustrate how to calibrate the BRD models under contrasting data scenarios: complete vs degraded observations, varying biological stochasticity, and alternative pathogen parameterisations. In each setting the toolbox recovers sensible posterior distributions and reproduces observed epidemic trajectories, with BRREWABC proving more robust for complex parameter spaces while abc offers faster turnaround on limited hardware. These examples show how posterior estimates can be plugged straight back into the EMULSION models and, by extension, into decision-support tools, thus closing the modelling-data loop for the whole project.

The developed software can be accessed at the following repository:

https://forge.inrae.fr/dynamo/decide-d2.4-inference-procedure

6 Warning systems

This section summarises Deliverable D2.5 **"Warning systems based on monitoring and simulation models"** and explains how the early-warning layer complements the multivariate and multilevel monitoring tools described in the previous sections of the present report.

6.1 Methodological approach

Early-warning is performed on the **standardised one-step-ahead forecast errors** produced by the Dynamic (Generalised) Linear Models developed in D2.1 and D2.2. Five detectors are supplied, ordered by increasing modelling complexity:

Detector	Principle (applied to DLM errors)	Typical use-case & main strengths	Main limitations
Shewhart 3-sigma chart with Montgomery rules	Flags points or runs that breach pre-defined control limits	• Instantaneous alarms • Very easy to interpret by farmers/technicians	• Sensitive only to large shifts • Assumes independence/Normality
V-mask on cumulative sums (CUSUM)	Uses a moving "V" overlay to capture small, persistent drifts in the cumulative forecast error	• Best balanced accuracy in the case study • Visual explanation of alarms	• Requires data-specific tuning of angle and dist • Slight detection delay
Tabular CUSUM	Computes positive and negative CUSUMs and signals when they cross threshold *h*	• Parameter (k, h) can be chosen for a target ARL • Minimises average delay for a given shift	• Choosing k/h is non-trivial on short series • Same distributional assumptions as above
Mahalanobis distance (χ^2 test)	Whitened multivariate error vector → χ^2 statistic compared with theoretical upper control limit	• Handles correlation between several sensor streams • Very high sensitivity after calf-level aggregation	• Low specificity (many false alarms) • Needs stable covariance estimates
Random Forest classifier	Non-linear ensemble trained on a feature set that combines raw errors and detector outputs	• Captures interactions & seasonality • Best AUC in the test herds (up to 0.84)	• Requires labelled outbreaks for training • Lower interpretability

6.2 Implementation

- **Software assets** Two fully commented R scripts (one with generic monitoring functions, one walkthrough) plus an anonymised calf dataset are provided in the DECIDE GitHub organisation under WP2-Deliverables/D2.5_Early-WarningSystems and mirrored on accelCLOUD .
- **Dependencies** All code uses base-R plus the packages *tidyverse*, *randomForest* and *qcc*; no proprietary software is required.
- **Workflow** The scripts:
    1. train a DLM on “healthy” herds,
    2. predict a held-out herd (leave-one-herd-out CV),
    3. compute the five detectors above,
    4. output an alarm timeline and a performance summary table ready for WP3 dashboards.

6.3 Demonstrated performance on calf-respiratory dataset

Detector	Best metric (cross-validated)	Comment
Shewhart + Montgomery	MMA 0.59 on *ut_DrinkingSpeed*	Highest specificity (81 %)
V-mask CUSUM	Balanced accuracy 0.66 (angle 0.2)	Fastest detection of subtle drifts
Tabular CUSUM	MMA 0.65 (k 0.75, h 3.34)	Slightly lower sensitivity than V-mask
Mahalanobis χ^2	Sensitivity > 90 % after calf aggregation	Specificity drops to ≈ 24 %
Random Forest	AUC 0.50–0.84 across herds	Major mean accuracies up to 79.5 % were observed

Overall, the highest performances by far was achieved with the Random Forest method, which suggests that (at least for these data) the ability to include additional relevant variables in addition to the DLM outputs improves the detectability of diseases in calves.

7 Overall conclusions

Work Package 2 set out to supply DECIDE with an open-source **tool-chain that spans the full modelling cycle** – from data ingestion and monitoring to mechanistic understanding, statistical calibration and early warning. Deliverables D2.1-D2.5 each contributed a building block, and D2.6 now brings them together and shows how the parts interlock.

- **Monitoring engines (D2.1 & D2.2).** Multivariate and multilevel Dynamic (Generalised) Linear Models allow partners to turn heterogeneous sensor streams into **standardised, autocorrelation-free** residuals that are comparable across animals, pens or farms. The R scripts released with these deliverables perform estimation, filtering, smoothing and basic diagnostics in just a few function calls .
- **Mechanistic simulation models (D2.3).** Eight EMULSION2 models now cover bovine respiratory disease, veal-calf management and swine **Mycoplasma** infections. Their declarative YAML structure and the PASTE web-app pipeline make them transparent and easily re-purposed by other WPs .
- **Inference toolbox (D2.4).** The twin R packages **BRREWABC** (parallel ABC-SMC) and **abc** (ABC-regression) give users a likelihood-free route to parameter calibration, with tutorials that walk non-statisticians through tolerance choice and posterior checking .
- **Warning systems (D2.5).** Classical control charts (Shewhart, V-mask CUSUM, Tabular CUSUM), a multivariate χ^2 test and a Random-Forest classifier were benchmarked on calf-respiratory data. While Shewhart rules remain the most interpretable, their MMA plateaued around 0.53–0.59 and V-mask sensitivity improved only after calf-level aggregation . **The Random Forest, by contrast, reached AUCs up to 0.84 and MMA near 0.80 in the best herd** . This clearly shows that, at least for these data, **enriching the DLM outputs with additional contextual variables** (age, season, management) yields the largest gain in early-warning power.

7.1 Strengths

- **Open and modular.** Every component is released under OSI-approved licences, with Git/SWH snapshots, tutorials and example datasets, ensuring long-term reproducibility.
- **Vertical integration.** Residuals from the monitoring layer can be streamed directly into the inference or machine-learning layers; simulation models can be retro-fitted with data-driven triggers; and all code is written in R or YAML, maximising accessibility.
- **Performance head-room.** The Random-Forest results confirm that the framework is ready for modern ensemble learners, which can exploit new sensor types as they become available.

7.2 Limitations

- **Data-hungry methods.** Machine-learning detectors need labelled outbreaks; control-chart thresholds still require herd-specific tuning; and ABC-SMC remains computationally intensive for high-dimensional models.
- **Calibration pending.** Several mechanistic models still mix literature priors with expert opinion; large-scale posterior estimation using D2.4 is planned for 2025–2026.
- **Interpretability vs. accuracy trade-off.** The Random Forest outperforms statistical charts but is harder to explain to farmers and regulators, suggesting that hybrid dashboards (e.g. RF probability gated by a Shewhart sanity-check) will be preferable.

7.3 Key take-aways

1. **Early-warning works best when residual-based features are** augmented **with biologically or contextually relevant covariates.** The Random-Forest’s superior performance illustrates this principle on calf data.
2. **Layering beats single methods.** Control charts remain valuable as fast, transparent screens, while multivariate tests and machine-learning layers add depth where data allow.
3. **Simulation and monitoring are no longer silos.** EMULSION models can now call real-time DLM predictions for adaptive interventions, and ABC-calibrated posteriors can feed back into decision thresholds.