BURLEIGH DODDS SERIES IN AGRICULTURAL SCIENCE

Advances in precision dairy and beef farming technologies

Edited by Professor Daniel Berckmans and Professor Tomás Norton, Katholieke University of Leuven, Belgium

E-CHAPTER FROM THIS BOOK

burleigh dodds SCIENCE PUBLISHING

Unsupervised machine learning techniques in precision dairy farming

C. Nickmilder, H. Soyeurt and I. Alexakis, University of Liège, Belgium; T. Przybylski, Walloon Agricultural Research Centre, Belgium; M. Gertz, LKV Schleswig-Holstein, Germany; and S. Franceschini, University of Liège, Belgium

1 Introduction
2 An unsupervised learning taxonomy
3 Dimensionality reduction approaches
4 Clustering approaches
5 Other types of algorithm
6 Examples of unsupervised learning algorithms
7 Guidelines for validation and interpretation
8 Combining unsupervised learning and knowledge discovery in databases
9 Case studies in dairy farming
10 Conclusion and future trends
11 Where to look for further information
12 Acknowledgement
13 References

1 Introduction

A modern livestock production system should have a low impact on its environment, have high welfare standards for animals, be economically profitable, and should produce sufficient quantities of high-quality food to meet consumer demand. To meet these conditions, animals and their environment should ideally be monitored continuously so that farmers can intervene quickly if necessary. Such a task may have appeared impossible a decade ago, but advances in digital devices, their interconnectivity and the overall digitalisation of agriculture has enabled a whole new 'digital ecosystem' which is now being used as a matter of course on many farms. Large amounts of digital data can

http://dx.doi.org/10.19103/AS.2025.0167.14
© The Authors 2026. This is an open access chapter distributed under a Creative Commons Attribution 4.0 License (CC BY).
Chapter taken from: Berckmans, D. and Norton, T. (eds.), *Advances in precision dairy and beef farming technologies*, pp. 201–246, Burleigh Dodds Science Publishing, Cambridge, UK, 2026, (ISBN: 978 1 83545 194 6; www.bdspublishing.com)

be collected in the barn or in the field, remotely or from sensors on or even within animals themselves. Satellite images can be collected to assess field conditions; animal feed, milk and meat can be analysed using chemical and/or spectroscopic methods; whilst soil and meteorological conditions can be collected (e.g. using a meteorological station on the farm). Whilst there remain challenges in collecting accurate data using suitably robust and validated methods with standardised protocols (i.e. gold standard), there is now an increasing focus on how best to aggregate and analyse the huge and growing volume of data that is now becoming available to farmers and others in order to produce actionable insights that can be used to improve production systems.

The twentieth century saw major advances in statistics, together with incredible advances in computer science and computer performance. This enabled the emergence of a new branch of statistics, commonly called machine learning (ML), defined as a field of computer science that studies algorithms and techniques used to solve complex problems (Rebala et al. 2019). ML techniques have enabled automated searching for significant patterns in large volumes of data, in a way that was virtually impossible to achieve with formal statistical models such as the analysis of variance or linear modelling.

ML models are part of a larger group of artificial intelligence (AI) models (Panesar, 2019), defined as algorithms that train machines to be 'intelligent' (Rebala et al., 2019). However, it is important to be careful of potential confusion in terminology. AI is often understood as equivalent to large language models (LLM) such as ChatGPT, Claude or Le Chat. However, the whole spectrum of AI is broader than just LLMs which are often based on techniques such as neural networks. It is important to be aware that, despite the impressive achievements of deep neuronal networks and especially LLMs in recent years, they are not always suitable for different data analysis tasks.

Neuronal network-based methods are by definition (universal approximation theorem) 'generalists' (Cybenko, 1989). In contrast, more specific 'specialist' ML methods (which rely on techniques such as cost functions and hyperparameterisation) have the potential to undertake a specific data analysis task more accurately as well as more rapidly and with less computing power (and cost) required. ML techniques provide a framework for iterative optimisation along a gradient to minimise the cost function (i.e. a measure of how effectively a model is achieving the kind of target outputs a user is looking for) and maximise the model's fit to the ground truth (i.e. how it matches to verified data that can be used as a benchmark to train ML models and validate results). In a similar way to neuronal network-based methods, ML algorithms also learn an optimal set of weights to discriminate a ground truth within complex datasets but do so using specialised algorithms to avoid the computational overhead associated with neural networks.

Published by Burleigh Dodds Science Publishing Limited, 2026.

There are a number of general data characteristics that may determine which ML technique may be most suitable. These include: the size of the dataset, the nature of data inputs (features) and their relation or the absence of responses to predict.

In terms of modelling, ML can be split into two distinct kinds of learning:

- Supervised learning and
- Unsupervised learning.

In supervised learning (SL), data contains examples of inputs (features) and outputs (labels); these training pairs help guide algorithms in inferring what desired output values to look for. There is a quantitative or qualitative (i.e. label) response associated to each record, allowing the model to be trained to estimate this target against specific quantitative and/or qualitative features.

In unsupervised learning (UL), the dataset has no response variable. UL is 'a kind of ML strategy utilised for drawing inferences from the datasets containing input data without any labelled responses' (Chander and Vijaya, 2021). In contrast to supervised learning, which aims to optimise a model to accurately predict an outcome, UL focuses on uncovering structure within a dataset without predefined labels. It simplifies data to highlight patterns or inherent features in the data in order to make it more analysable (e.g. in relating a particular feature to an outcome). A key approach is the optimisation of a mathematically-derived quantity that captures an essential property of the underlying data. A simple example is the ratio of variance within and between groups created. More complex features may involve abstract representations in a latent space (i.e. a simplified representation of multi-dimensional data such as pixels in an image), which can be derived through advanced mathematical transformations and projections. UL is often considered as exploratory data analysis to present data in a form that can then be analysed more easily in subsequent steps (James et al., 2023), although its applications extend well beyond this. Indeed, some detected patterns may have immediate practical relevance.

This chapter will discuss these algorithms and their application within the scope of dairy science. As the chapter shows, supervised and UL can also be inter-related. The first sections will present the taxonomy of UL and the concepts behind specific algorithms. Guidelines to addressing difficulties encountered in UL are explored. Finally, examples from animal production using UL are presented later in the chapter. The chapter aims to explain UL in particular within ML and its relevance in modern dairy farming. To achieve this, the chapter combined a traditional manual review process with the inclusion of AI research engines to help the authors find overlooked viewpoints as highlighted by Rane et al. (2024).

2 An unsupervised learning taxonomy

This section discusses a taxonomy of different approaches to UL based on their individual computational approaches. The aim is to give the reader a general understanding and a simplified overview of the approaches available before starting a detailed description of the most important algorithms. However, UL algorithms do not have one unique taxonomy. As discussed later, they can be categorised variously according to their objectives, their outputs or their computational approaches. The taxonomy presented here is inspired by Patel (2019) outlining two main types of UL:

- Clustering and
- Dimensionality reduction.

The classification shown in Fig. 1 is based on Ezugwu et al. (2022).

Though most approaches can be classified under the categories of clustering and dimensionality reduction, other approaches exist for specific

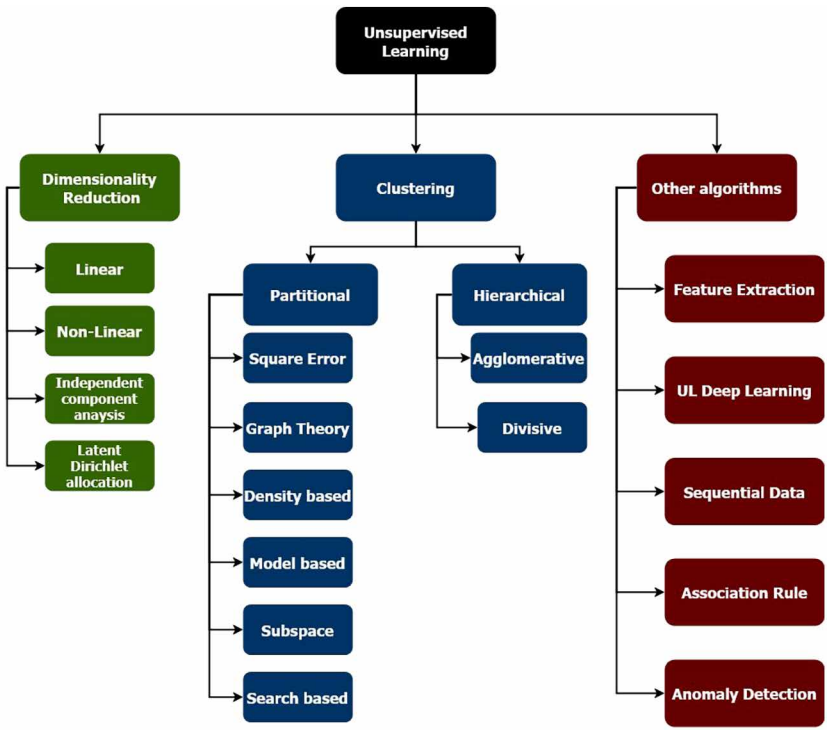

Figure 1 Unsupervised learning taxonomy based on Patel (2019) and Ezugwu et al. (2022). UL = Unsupervised learning.

objectives such as anomaly detection or feature extraction, or for specific types of data such as sequential data, or specific methodologies such as deep learning or the association rule. Whilst these are mentioned, the main focus is on clustering techniques. For each section a figure will be included to explain the concept with a theoretical example from milk recording. In this example, there are nine milk samples, three for each cow, and each milk sample is analysed using Fourier transform mid-infrared (FT-MIR) spectrometry. Each milk sample contains around a thousand different absorbance values for different wavelengths.

3 Dimensionality reduction approaches

This family of algorithms aims to reduce the dimensionality of a dataset whilst retaining the most relevant features and the intrinsic structure of the data. This approach is particularly useful for high dimensional datasets (i.e. data with a large number of variables or features compared to the number of observations). This is characteristic, for example, of data from spectroscopy. The reduction is achieved through projection of the data into lower-dimensional space, which suppresses random noise and non-essential information. This family of algorithms is further divided into four categories (Patel, 2019):

- Linear;
- Non-linear;
- Independent components analysis (ICA); and
- Latent dirichlet allocation (LDA).

Principal component analysis (PCA) is the most common linear approach (Fig. 2). This approach uses singular value decomposition (SVD) to simplify complex, high-dimensional datasets by creating smaller sets of variables called principal components. These are linear combinations of original data features optimised to represent the highest amount of data variability (Greenacre et al., 2022). Many variations of PCA are used to offset potential limitations of this approach, one of which is the requirement of continuous features as input. In this case, correspondence analysis (CA) allows for categorical features as input whilst using chi-square distances from a contingency table (Greenacre et al., 2022).

In some cases, a linear projection is not enough to capture the complexity of the dataset. A non-linear approach is then required to interrogate more complex data. Those methods, also called manifold learning, aim to capture those non-linear relations through different techniques (Meilă and Zhang, 2024). As an example, isometric mapping, also called isomap, could represent three-dimensional features on a two-dimensional plane; it computes nonlinear pairwise geodesic distances between observations before projecting them in a

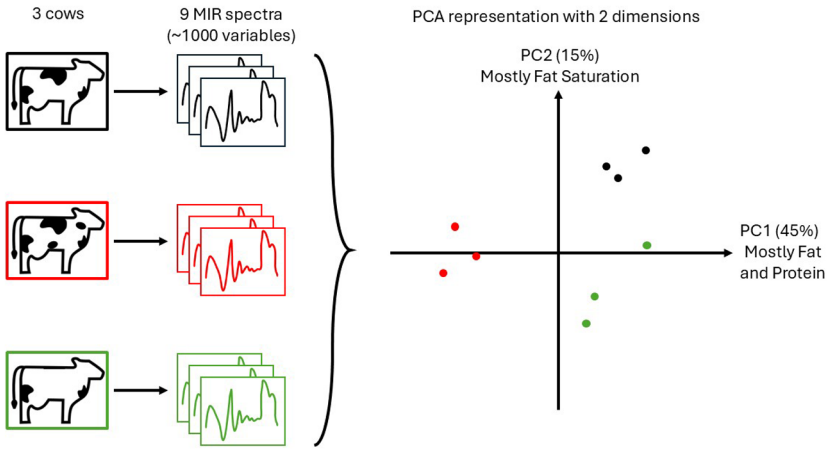

Figure 2 Concept of dimensionality reduction and its use for visualisation. Nine high dimensional spectra are simplified into two principal components (i.e. linear combinations of the absorbances) representing already 60% of the variability of those spectra which is mostly the fat and protein content.

low-dimensional space (Izenman, 2012). Among those methods, t-Distributed Stochastic Neighbour Embedding (t-SNE) (van der Maaten and Hinton, 2008) and Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) are also worth mentioning. Both algorithms aim at reducing dimensionality but also retaining key local relationships between observations. UMAP has the advantage of being repeatable (McInnes et al., 2020). A non-linear variant of PCA is the Generalised Low Rank Model (GLRM) Method (Udell et al., 2016), extending PCA by approximating a data matrix as the product of two low-rank matrices under customisable loss functions and regularisers (Udell et al., 2016). All these techniques can be used to visualise and to identify underlying patterns in 'messy' data which is particularly the case with measuring phenomena in agricultural settings which are, by definition, particularly complex and dynamic environments.

The other two categories are less commonly used. Independent component analysis (ICA) is often used as a preprocessing method to extract features from a blended signal. The general idea is to detect features that are uncorrelated and to consider those that are statistically independent (Stone, 2002). Latent dirichlet allocation (LDA) is a type of dimensionality reduction statistical model. The objective is to model count data as mixtures of topics. A typical use-case is identifying a specific research topic within a large text document (Blei et al., 2003). All those dimensionality reduction algorithms could be used for data visualisation or as a first preprocessing step in a data analytics pipeline but other applications exist.

Published by Burleigh Dodds Science Publishing Limited, 2026.

4 Clustering approaches

Clustering approaches can be helpful when analysing large datasets where data is likely to fall into natural groups (i.e. groups of data with shared characteristics) related to specific underlying processes. UL can easily be confused with clustering because it is such a commonly-used technique. However, whilst clustering is a key part of UL, other UL categories exist and should not be overlooked. Clustering algorithms aim to divide a dataset into groups, commonly called clusters, optimised to explain dataset variability. Within clustering algorithms, a commonly categorisation is between (Jain et al., 1999; Pitafi et al., 2023):

- Partitional clustering, and
- Hierarchical clustering.

Hierarchical clustering involves arranging groups with similar data points into a cluster hierarchy. Clusters are iteratively formed, either through merging from individual to population levels of records (i.e. agglomerative clustering also called bottom-up clustering) or dividing groups from the population level to the individual (i.e. divisive clustering also called top-down clustering) (Ezugwu et al., 2022).

In contrast partitional clustering organises groups without a hierarchical structure, potentially avoiding the intrinsic pitfall of distance computation required for hierarchical clustering (Ezugwu et al., 2022). Partitional strategies organise patterns into clusters (i.e. one partition), whilst hierarchical clustering consists of a sequence of nested data partitions (Fred and Leitão, 2000). Figure 3 from Gao et al. (2023) illustrates this difference. The classification for this section comes from Ezugwu et al. (2022) with some slight changes to simplify it for non-specialist readers.

4.1 Hierarchical clustering approaches

Hierarchical clustering can be useful for certain types of data, particularly data with an inherently hierarchical structure. Algorithms in this category aim to establish a hierarchical structure among data based on a measure of dissimilarity or similarity between clusters (Ran et al., 2023). These algorithms are used to solve clustering problems where relations between observations are important. For example, Xu et al. (2020) used this approach to analyse social behaviour among ten yearling cows, whilst Soyeurt et al. (2022) used this approach observe bulk tank milk samples over different months.

To group data, it is essential to define a method for measuring elements and their distances relative to one another, allowing the determination of group

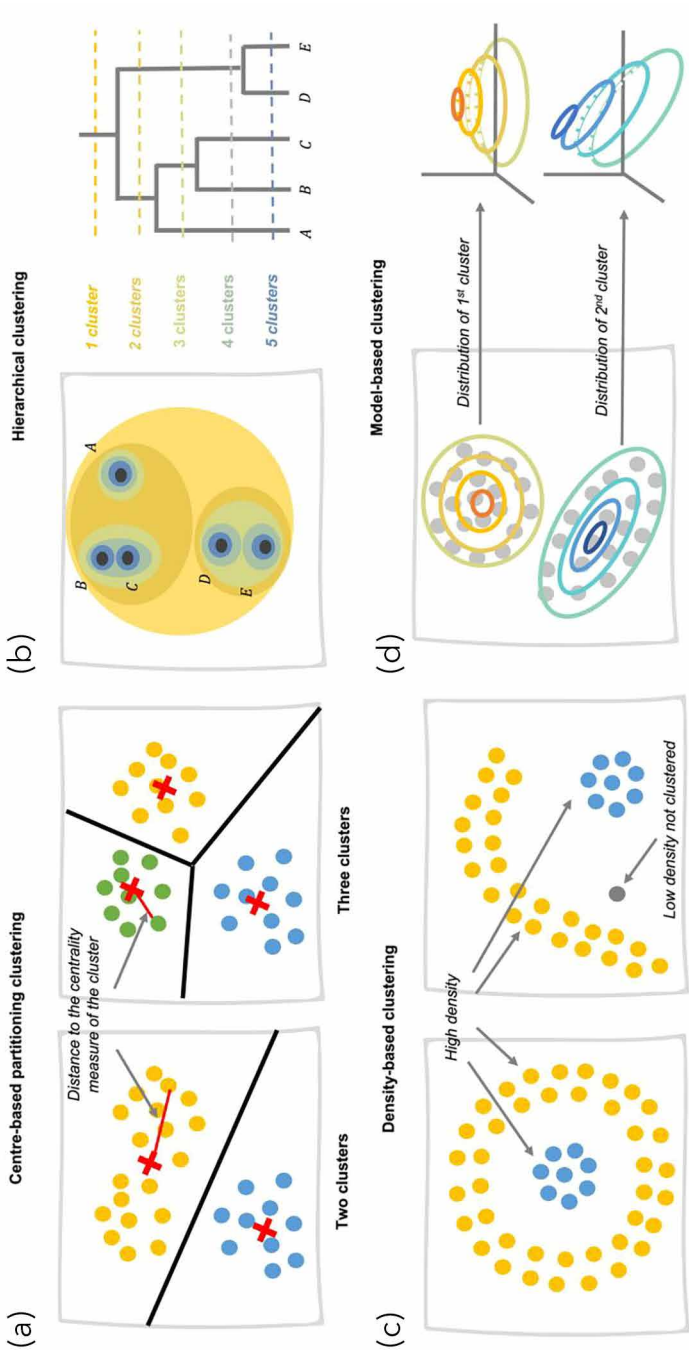

Figure 3 Different techniques of clustering and their concept. In our classification, (a), (c), and (d) are different techniques from the partition-based approach and (b) is hierarchical clustering. Figure from Gao et al. (2023).

membership (Murtagh and Legendre, 2014). The initial version of hierarchical clustering consisted of grouping records iteratively based on similarity. In most algorithms, this similarity is a distance between observations or groups of observations.

As noted, two main approaches are often used (Murtagh and Contreras, 2017; Patel, 2019; Ezugwu et al., 2022; Gao et al., 2023):

- agglomerative hierarchical clustering, and
- divisive hierarchical clustering.

Figure 4 illustrates these two clustering approaches.

The first is the agglomerative or bottom-up approach where each observation is considered individually and linked with other observations (Fig. 5). Linked similar observations are grouped to make new clusters. A dendrogram can also easily be produced to represent the relationships among clusters as output.

The agglomerative hierarchical clustering algorithm has two main hyperparameters to tune, one regarding similarity measurement and the other regarding the method to compute distance between observations and clusters, or between clusters (Murtagh and Contreras, 2012, 2017). Many algorithms are based on hierarchical clustering with enhancements to overcome limitations in the original method, including balanced iterative reducing and clustering using hierarchies (BIRCH) (Zhang et al., 1996) or clustering using representatives (CURE) (Guha et al., 1998).

The second approach is the divisive or top-down approach, where all observations are considered as a single group and splits are made iteratively based on similarity until clusters of one observation are created (Chander and Vijaya, 2021; Mahdi et al., 2021). One such algorithm is divisive analysis clustering (DIANA) (Kaufman, 2005).

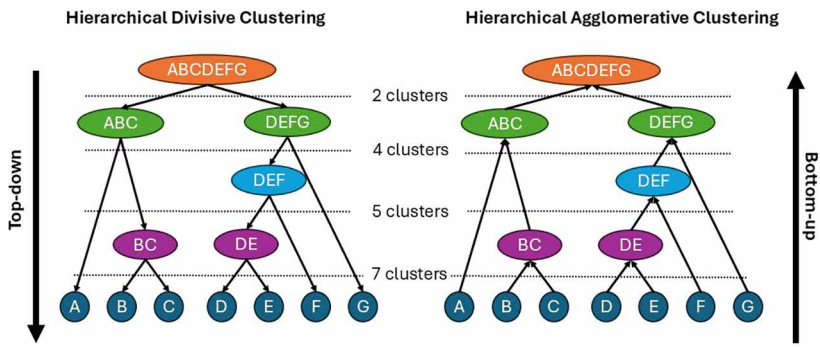

Figure 4 Divisive and agglomerative hierarchical clustering and their dendrograms.

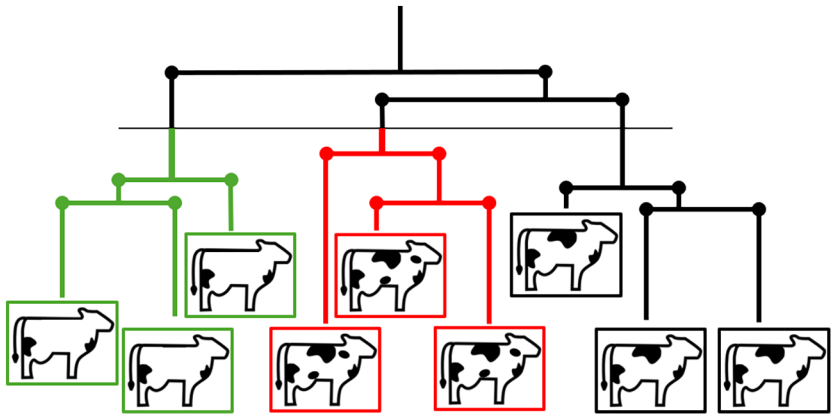

Figure 5 Agglomerative hierarchical clustering. The nine milk samples are grouped together based on their similarity. At each step, two samples or groups of samples are merged. In this example, the dendrogram representing those merges is divided into three groups. Because the main difference between the nine samples was the different cows, it is most likely to observe three groups representing the different cows.

The two hyperparameters underpinning such algorithms are similarity measurements and how these similarities are handled for groups of observations. To compute the (dis-)similarity between and within records/groups, some similarity metrics can be estimated. The most common is Euclidean distance, but other distances exist such as Manhattan distance or Gower distance. Table 1 gives some examples of similarities metrics found in the literature. These distances are often sensitive to the feature scale, hence the need for standardisation or using feature transformation into another reference space (or manifold).

For some similarities, the algorithm needs a solution to compute the distances between groups of observations as it is not simply the distance between two observations anymore. Multiple methods exist such as the single-linkage, the complete-linkage and the average-linkage. The single-linkage computes the distance between the two closest observations from two distinct groups whilst the complete-linkage computes the distances between the two farthest observations. Finally, the average-linkage computes the average distance between all observations.

Some advantage of these algorithms is that there is no need to specify the number of clusters in advance, the clustering results are visualised easily using a dendrogram, and a flat partition can be achieved at varying levels of granularity by making cuts at different heights in the dendrogram (Ran et al., 2023). The height at which the cut is made is then a reflection of the variability represented. A certain percentage of variability within and between clusters helps set the global threshold for cluster determination.

Table 1 Examples of similarities metrics

Name	Formula	Explanations
Euclidean distance*	$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$	Length of the straight line between two points corresponding to the displacement along the cartesian axes.
Manhattan distance*	$d = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$	Sum of the absolute differences of the cartesian coordinates.
Relative entropy	$d = \sum P(x) log \frac{P(x)}{Q(x)}$	Mathematical expectation of information content of a random variable (Bonilla et al., 2021), can be used to get the amount of information detected (Shahinfar et al., 2014). It is the goodness of fit of a distribution respectively to another distribution.
Gower distance	$d = \frac{\sum ws}{\sum w}$ For binary, s = 0 or 1 $s = 1 - \frac{\lvert x - y \rvert}{range}$ For continuous feature	Distance capable of dealing with mixed types of features. It can include feature weights (w). Depending on the implementation, the computation of the pair's similarities might differ.
Inertia*	$I = \frac{1}{n} \sum_{i=1}^{n} d(x_i, center)^2$	It corresponds to the sum of squared distance of points to their nearest cluster centre. Often adopted in k-means clustering to explain the variability covered by the clusters.

Similarities with a * are sensitive to feature scale.

This might be expressed as a percentage of the variance explained or inertia measurement.

However, this kind of clustering is computationally demanding, especially due to computation of the distance matrices when performing using a large-scale dataset. Indeed, the memory required is the square of the number of observations, even though some smart implementations divide this requirement by a factor near to two by using triangular distance matrices instead of square ones. This has led to approaches using smaller datasets sampled from a global database. These include a divide and conquer approach (Franceschini et al., 2022) or an iterative clustering approach (Narita et al., 2018). The divide and conquer approach consist of drawing samples from the population (also called batches), attributing them to the nearest centroids, recompute the

centroids of the clusters and repeating the sampling. Iterative clustering consists in considering batches one after another and integrating previous clusters centroids/representatives in the following run. As the amount of data is decreased, the processing time and power requirements both decrease too (Ran et al., 2023).

4.2 Partitional clustering approaches

Partitional clustering approaches can be preferable to hierarchical approaches for their computational speed, scalability and suitability for data with more clearly-identifiable clusters. Partition-based clustering algorithms divide observations into clusters, each called a data partition. Approaches are classified as hard-partitioning or soft-partitioning, depending on whether the output is a membership value or has the probability of belonging to a cluster. Multiples categories of algorithms exist, depending on the approach used to determine the group (Ezugwu et al., 2022). Some algorithms optimise a squared error, by trying to maximise the ratio between the inter- and intra-cluster variability. The most common squared error algorithm is the k-means algorithm (Bock, 2007). This algorithm divides a dataset by iteratively computing the average distance between the centre of a cluster (called a centroid) and the observations in the cluster, then updating the value of the centre and attributing new clusters to observations until it converges. The result is one partition of the dataset into k clusters. Other algorithms could use the density of the data to determine whether observations should be grouped together (Fig. 3). Those algorithms work similarly to algorithms for anomaly detection except they try to find groups instead of anomalies. Groups are formed in high density areas with cutoffs defined by low density areas. One such algorithm is Density-Based Spatial Clustering (DBSCAN) (Ester et al., 1996).

Not to be confused with density-based clustering, model-based clustering algorithms use probabilistic distributions of the data (Fig. 3). The method considers data as observations from underlying distributions and tries to find the most likely way of dividing the observations through Expectation Maximization (Fraley and Raftery, 1998). These algorithms include the Gaussian mixture model (GMM).

Subspace clustering algorithms aim to avoid the 'curse of dimensionality' inherent to high dimensionality datasets. This well-known problem in data analysis is where attempts to organise data in many dimensions (by feature or attribute) results in insufficient data in any dimension to produce meaningful results. It can also result in overfitting (i.e. fitting to irrelevant background 'noise' rather than any real underlying pattern). This is achieved by performing feature transformation or feature selection. These algorithms search for a subspace, among many possible combinations, where observations are easily divided into

groups. Different algorithms have different ways to explore possible subspaces (Parsons et al., 2004). One such algorithm is CLIQUE (Agrawal et al., 1998).

Graph-theoretic clustering algorithms work with the same concept of grouping similar observations. Differences depend on the input used to perform clustering. In this approach, the input is a graph with observations called nodes and relationships between them called edges (Schaeffer, 2007). A later section in this chapter includes an example of a study from Bonora et al. (2018) using this methodology on time series data from an automated milking system (AMS) to study herd patterns. One example of a graph-based model, more specifically message-passing-based clustering, is affinity propagation (AP) (Frey and Dueck, 2007) also discussed later in the chapter. Search-based clustering, also called automated clustering algorithms, was created to solve the issue in which most clustering algorithms have hyperparameters to optimise and thus require optimisation (Agarwal et al., 2011). These hyperparameters correspond to values used to control the learning process. However, optimisation can be slow without a priori assumptions or larger volumes of data (Ezugwu, 2020; Ezugwu et al., 2022). Those algorithms are often inspired by natural processes such as the genetic algorithm based on evolutionary processes (Fig. 6).

5 Other types of algorithm

This section reviews the categories that we cannot classify directly in dimensionality reduction or clustering categories but that are worth mentioning. Such algorithms can be suitable for specific types of dataset in animal monitoring such as sequential data – data that present an inherent order between the records, as is typically the case with time series (e.g. tracks animal body temperature over time). Sequential data requires specific algorithms as most dimensionality reduction and clustering algorithms treat successive observations as independent rather than linked, thus losing a key aspect of such data. One example is the hidden Markov Model, which is specifically designed to handle sequential data such as time series Ghahramani (2004). At each step of the sequence, the algorithm will consider that there is a hidden state from which the observation comes and then search for that state. At the next step, the observation could come from the same state or from another state related to the previous one by a probability to switch from one to another (Mor et al., 2021). As an example, Pastell et Frondelius (2018) used this approach to estimate cow feeding time from indoor monitoring data.

Anomaly detection algorithms aim to detect whether a new or following observation follows the same rules as other observations, whether based on a threshold, distribution, density, distance or dynamic pattern. Those algorithms are also called outlier, fraud or novelty detection methods. The detection of outliers is a very important aspect before starting the training of a model because

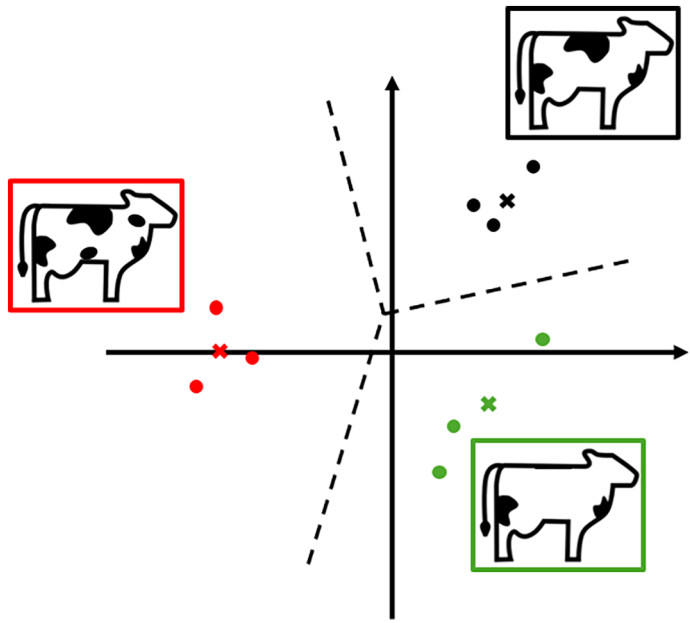

Figure 6 k-means clustering projected on a principal component analysis. The nine samples have around 1000 variables. To avoid the curse of dimensionality, they are projected into principal component (Section 2.1), then a k-means clustering is performed to find clusters that optimise the distances between the clusters (i.e. the coloured crosses) compared to the distances inside the cluster (i.e. the distances between the cross and the observation). This optimisation is equivalent to finding the minimum squared error.

their presence can impair global prediction performance. Whilst the most basic detection method is the use of quantile distribution, there are numerous methods for global or local, parametric or non-parametric, and univariate or multivariate detection (Samariya and Thakkar, 2023). Most anomaly detection algorithms are based on a common methodology: identifying a profile for nominal (non-anomalies) data points, then attributing the anomaly tag to the data points that correspond the least to that profile. Anomalies are thus defined as data points that have unexpected characteristics, which implies defining what expected characteristics are based upon process knowledge. In dairy science and more precisely in FT-MIR spectrometry applied to milk analysis, a common method is Global Hotelling Distance also called Standardised Mahalanobis Distance which is a common multivariate approach in chemometrics (De Maesschalck et al., 2000; Zhang et al., 2021).

The 52 anomaly detection algorithms described by Samariya and Thakkar (2023) include the Local Outlier Factor (LOF) and the Isolation Forest methods, which are discussed below. DBSCAN (mentioned earlier as a density-based algorithm) can also be useful for outlier detection. LOF is a density-based

algorithm using the assumption that outliers are found in low-density areas. The algorithm computes the density around an observation and compares it to the densities of the k-closest observations. If the ratio is significantly different, the observation will be considered an outlier (Alghushairy et al., 2021). The Isolation Forest works by isolating observations. The algorithm builds many random decision trees with observations divided based on randomly selected variables and thresholds. Since anomalies are typically rare and differ significantly from normal observations, they tend to be isolated faster, meaning they require fewer splits to separate them from the rest of the data.

Finally, there are the deep learning algorithms like autoencoders and generative adversarial networks (GAN). Those algorithms can perform anomaly detection but also conventional clustering and can even handle sequential data when paired with architectures such as long short-term memory (LSTM). However, those rely on deep neural networks to divide the dataset. As neural networks are based on a completely different paradigm, they are not covered in this chapter.

The overall taxonomy used so far allows us to characterise algorithms and models. However, this categorisation is quite theoretical. In many data processing pipelines, different algorithms are combined to bypass limitations and create more robust models. For instance, it is possible to achieve cluster-belonging probability from hard clustering by adding a supervised layer, or a DBSCAN algorithm could be implemented with a hierarchical aspect to create an HDBSCAN (McInnes et al., 2017).

6 Examples of unsupervised learning algorithms

6.1 Partitional clustering algorithms: k-means

As mentioned in the introduction, this chapter focuses on clustering algorithms, as most applications in the field are based on the detection of specific groups of records, e.g. detecting sick cows. We focus on some of the most utilised partitional clustering algorithms, as well as various emerging algorithms attracting particular interest. These are illustrated in Fig. 7.

Of the partitional clustering algorithms, the most used algorithm is k-means, originating from the work of Lloyd (1982) and MacQueen (1967). Examples of use include detecting milk adulteration (Sowmya and Ponnusamy, 2021), poor metabolic adaptation syndrome (Tremblay et al., 2018) and oestrus detection (Shahriar et al., 2015).

This algorithm requires the number of clusters to be fixed before starting the clustering process and primarily uses inertia for similarity measurement. First, it is necessary to set a number k of clusters we want to study. k points are then chosen randomly from the dataset. In some cases, the initialization of

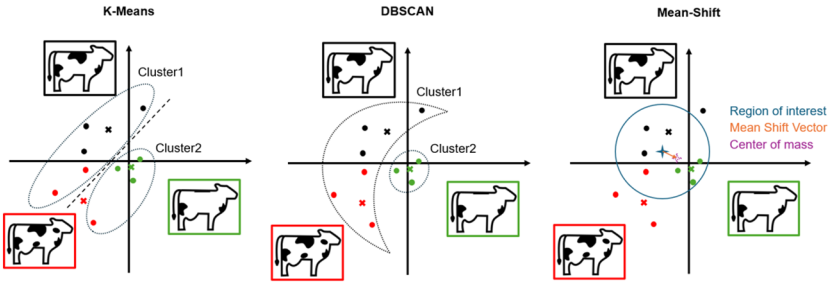

Figure 7 Illustration of k-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and mean-shift.

the cluster centroids can be set arbitrarily based on original sample probability distributions. These are considered cluster centres. All points are then assigned to the nearest cluster centre. These point clouds are subsequently processed to extract a new centre of mass, i.e. the average position of the points along each axis. These centres are referred to as centroids. If extreme values are present in the training dataset, it may be preferable to use the median instead of the mean to compute the cluster centres. This is called k-medoids instead of k-means (Oehm et al., 2022). This process of assigning points to the nearest cluster is repeated until the process stabilises and the clusters move only within a tolerable margin.

The main advantage of this algorithm is its speed and lower memory requirements compared to hierarchical clustering, making it suitable for use with a large-scale dataset. In the worst-case scenario for classical k-means clustering, the computational complexity is $(^{(+2)/}$), with O being the complexity, n the number of records, k the number of clusters, and *P* the number of features (Arthur and Vassilvitskii, 2006). In practice, the complexity is more like (n ∗ p ∗ k ∗ T), where T is the number of iterations. The 'T' highlights a drawback of this algorithm – it is overly sensitive to local minima as the algorithm will not explore alternative combinations, hence the need to repeat the process several times to ensure cluster stability. Since the number of clusters k is central to cluster identification, it should be closely aligned to the end goal. All the features entering the k-means create a hyper-space, often called a manifold. In this multi-dimensional space, clusters are assumed to be 'regular and convex', i.e. it might result in unexpected situations when data are not normally distributed along the features, exhibit non-homoscedasticity, or are uneven in group size within clusters, as the Euclidean distance is the most used and can become less reliable in highly dimensional spaces.

The major disadvantage of this algorithm is that the number of clusters must be defined in advance. As it might not be possible to predict the best number of clusters, this results in running the algorithm iteratively with a

growing number of clusters, and then applying common optimum finding methods to limit the number of clusters.

Another approach to determine the optimal number of clusters involves graphical analysis of the cluster repartition as it is not possible to make a dendrogram. Whilst it is feasible to simply monitor the cluster belonging to the original features distribution, multivariate approaches using combinations of features/manifolding, as explained above in the dimension reduction section, make it easier to discriminate between clusters. For example, PCA can help identify feature repartition and influence (Tremblay et al., 2018). From a graphical perspective, the silhouette score is a commonly used criterion to determine the optimal cluster number. This can be defined as the ratio between the difference of the mean intra-cluster distance and the mean nearest-cluster distance, and the maximum of these two values. It is the mathematical equivalent to assessing on a graph if the clusters are well grouped together and do not overlap. It has been used by Ma et al. (2024), Rebuli et al. (2024), and Tremblay et al. (2018) to detect misclassifications. Advanced refinements of these approaches relate to a current topic of interest, the automation of determining the optimal number of clusters (Shi et al., 2020).

Based on the explained variability, the traditional determination of the optimal set of clusters in k-means is the 'elbow curve'. In this approach, k-means clustering is run iteratively on a varying predefined number of clusters, and the percentage of variation explained is recorded. A graphical observation of the behaviour of the degree of variation explained against the number of clusters is then used to detect the point at which the slope changes abruptly – the 'elbow' of the curve. However, identifying the elbow can be highly dependent on the observer's perspective.

6.2 Partitional clustering algorithms: density based spatial clustering of applications with noise

Another example of partitional clustering is Density-Based Spatial Clustering of Applications with Noise (DBSCAN). This has been used by Michelena et al. (2025) to detect oestrus. This algorithm uses density-based clustering and is designed to cluster data of various shapes, even in the presence of outliers and noise, and from large datasets (Ester et al., 1996; Singh et al., 2022). One advantage it has over other traditional clustering methods (Singh et al., 2022) is that DBSCAN can identify clusters of any shape, not just spherical or convex ones, as illustrated in Fig. 6. It effectively handles noise and outliers by identifying them as points that do not belong to any cluster, enhancing clustering accuracy in real-world noisy datasets. Unlike k-means, DBSCAN does not require the number of clusters to be predefined, which is useful when this information is unknown.

In DBSCAN, the density of each sample is defined by the number of neighbouring points within a specified distance, controlled by the epsilon (ε) hyperparameter. For clustering, DBSCAN classifies each sample into three categories:

1. core points corresponding to samples with a minimum number of neighbouring points (determined by the MinPts hyperparameter);
2. border points related to samples with fewer neighbours than MinPts but still within the neighbourhoods of a core point; and
3. noise representing samples that are neither core nor border points.

To achieve this classification, the algorithm first randomly selects a core point and assigns it to a new cluster. Then, all core and border points within its ε-neighbourhood are added to the same cluster. Next, for each new point added, the previous step is repeated to find all densely connected points. This continues until no more points can be added. If unassigned core points remain, the process starts over with a new cluster. Upon completion, the algorithm forms clusters and identifies outliers (noise), making DBSCAN useful not only for clustering but also for anomaly detection.

The performance of DBSCAN relies heavily on tuning its hyperparameters, ε and MinPts. For MinPts, various heuristic approaches have been proposed, indicating that a good value for this parameter is twice the dimension of the dataset. For ε, k-nearest neighbour distance plots are often used, where the 'elbow' point indicates a suitable value (Michelena et al., 2025). A disadvantage of this algorithm is that it struggles when there is varied density in a dataset and is strongly reliant on user-defined parameters, which can be a limitation (Singh et al., 2022).

6.3 Partitional clustering algorithms: mean-shift

This algorithm is thought to originate from Fukunaga and Hostetler (1975), whilst the clustering dedicated application of the algorithm is considered to originate from Comaniciu and Meer (2002). Applied to image analysis, this algorithm is often related to image segmentation. In the context of dairy farming, it has been used by Yujuan et al. (2017) to monitor the rumination of cows.

According to the typology defined earlier, this class of clustering algorithms is partitional. Compared to k-means, it does not require the number of clusters to be defined before the fine-tuning and uses the combination of bandwidth and kernel for similarity measurement. This algorithm attempts to find the modes/peaks in the data distribution by iteratively recomputing the centres of gravity of the distribution. First, every record is considered as a potential cluster centre. Second, the algorithm defines a window/distance/radius/neighbourhood on

the manifold to inspect. Third, for each record, the algorithm investigates the window around the data point and computes the centroid of all the points within this radius. Fourth, it shifts each record to this associated recomputed centroid. This implies the displacement of points to higher density regions, hence the name 'mean-shift'. Finally, the process of defining a window until its association to a centroid is repeated until the algorithm converges, i.e. the displacement caused by the shifting of records is below a certain threshold. The local maxima highlighted with this method are the clusters. A post-processing filter might be needed to eliminate near duplicate centres to form the final set of centroids.

From a more theoretical point of view, the size of the neighbourhood to inspect is called the bandwidth (h). This is used within a kernel density estimation (KDE) function. This KDE function aims to reduce the impact of data points by giving more weight to points closer to the individual study points. Often, this KDE is a Gaussian kernel. More information on kernel choice and its constraints can be found in Comaniciu and Meer (2002).

This algorithm relies heavily on the metric of the feature space – therefore, it is preferable that all input features are converted into a Euclidean space in order to reduce constraints on the bandwidth parameters and allow the use of a consistent kernel size for all mean-shift procedures (Comaniciu and Meer, 2002). This algorithm uses a gradient approach for the displacement of the neighbourhood in question. As a result, it may be sensitive to regions that are not true local maximum likelihood but rather areas influenced by a common background noise. Another potential issue affecting convergence is the presence of 'orphan' points that lack neighbouring data points. These may be ignored or flagged as outliers. Similar to the k-means algorithm, this requires a parameter to be manually tuned – in this case the bandwidth to achieve optimal discrimination. Given the wide variation in KDE choice, the bandwidth, the method to identify cluster members and the potential sampling, defining the algorithm's complexity is not straightforward. Assuming the adoption of the scikit-learn implementation, a Python package for ML (Pedregosa et al., 2011), complexity is typically in the order of $O(T*n\hat{}2)$ with T being the number of records and n the number of samples considered.

To accelerate cluster identification, it is possible to resample the input data with a regular grid and process these sub-spaces independently first. Afterwards, these sub-spaces can be regrouped to reiterate and identify the global centres.

6.4 Partitional clustering algorithms: FASTCLUS

The so-called proc FASTCLUS procedure in the SAS® statistical software package is a lesser-known variant of the classical k-means approach. The FASTCLUS procedure was inspired by the Hartigan (1975) leader algorithm

and the MacQueen (1967) k-means algorithm (SAS, 2014, pdf. p. 2393). This algorithm was utilised by Alessio et al. (2021) to determine relationships in bulk tank milk data.

In this implementation, k-means is optimised for large datasets and speed. By default, FASTCLUS uses Euclidean distance, meaning that centres are based on least squared mean estimation, which is equivalent to k-means clustering. However, whereas k-means requires several iterations to converge, the FASTCLUS method provides good, if not necessarily optimal, clusters in as few as two or three iterations (SAS, 2014, pdf. p. 2394). This is largely due to its initialisation method. The FASTCLUS procedure employs a method known as nearest centroid sorting (Anderberg, 1973) to select initial cluster seeds in a way that promotes separation. This differs from traditional k-means, which often selects initial seeds randomly. However, this initialisation method is sensitive towards outlier data, which can be beneficial if this method is to be used for outlier detection but may be disadvantageous if robustness against outliers is required. The procedure is designed to run well with datasets consisting of more than 100 observations, can handle missing values, and requires the variables to be standardised.

Overall, Proc FASTCLUS is fast and efficient but limited in flexibility, as is common with most SAS procedures. Its main advantage lies in its integration within the SAS ecosystem, ensuring well-tested procedures and seamless compatibility with other SAS procedures. On the downside, its sensitivity towards outliers must be considered, and SAS procedures tend not to incorporate the latest developments and methods in statistics.

6.5 Partitional clustering algorithms: affinity propagation

Affinity propagation is an innovative clustering algorithm introduced by Frey and Dueck (2007) that operates by exchanging messages between data points to identify exemplars (i.e. representative points) and assign other points accordingly. It has been used to extrapolate whole crop lines from image-derived centroids by Hongbo et al. (2024) and to detect subclinical mastitis in dairy sheep farms by Kiouvrekis et al. (2024).

Unlike k-means or k-medoids, affinity propagation does not require the number of clusters to be specified in advance. Instead, it takes as its input a similarity matrix between pairs of points, where each similarity s(i,k) reflects how well point k represents point i. The core of the algorithm is based on the iterative exchange of two types of messages between data points – the responsibility message, r(i,k), indicates how well-suited point k is to serve as the exemplar for point i, considering other potential exemplars; and the availability message, a(i,k), reflects how appropriate it would be for point i to choose k as its exemplar, based on how much support k receives from other points. These

messages are updated according to simple rules, and at each iteration, each point is assigned to the exemplar k that maximises the sum of r(i,k) + a(i,k).

Rather than imposing a fixed number of clusters, the affinity propagation uses a real-valued input s(k,k) for each data point k, known as the preference. The higher this value, the more likely the point is to be chosen as an exemplar (i.e. cluster representative). The resulting number of clusters therefore depends on both the specified preferences and the message-passing process. If all data points are considered equally likely to become exemplars, a common preference value can be assigned. Adjusting this value allows control over the number of clusters – for example, using the median of the input similarities often yields a moderate number of clusters, whilst using the minimum tends to produce fewer clusters.

The algorithm offers several advantages, including not requiring a predefined number of clusters or their centres, relatively low computational complexity, consistent results across multiple runs, and a smaller amount of squared errors compared to other clustering methods.

6.6 Partitional clustering algorithms: Gaussian mixture model

The GMM is a soft-clustering method whereby the data is fitted to belong to Gaussian distributions (McLachlan et al., 2019). It has been used by Toan et al. (2019) to detect oestrus from a neck-mounted accelerometer. The term 'soft' refers to the fact that each observation is not assigned to a single cluster outright, but instead has a probability of belonging to each distribution in the mixture. This approach builds on the Expectation-Maximisation (EM) algorithm (Dempster et al., 1977), which iteratively refines the model as follows.

In the Expectation step (E-step), the algorithm estimates the probability of membership for each data point in one of the Gaussian distributions, represented by a latent (unobserved) class variable using Bayes' Theorem. In the Maximization step (M-step), these estimated class memberships are used to update the parameters of the Gaussian distributions – such as their means, variances, and mixture weights – to maximise the overall likelihood of the data. The updates are weighted by the probabilities, meaning that each data point influences the parameter estimates in proportion to its likelihood of belonging to each component. By iteratively repeating the E-step and M-step, both the distribution parameters and the class membership estimates improve progressively, leading to convergence at a (local) maximum of the likelihood function. Observation can be with a probability of P = 60% part of Cluster A but with a probability of P = 45% also part of Cluster B, and so on. This identifies which data points are confidently attributed to a certain group, and which are in the grey zone where two or more predicted groups overlap (those points will have close highest probability values).

GMM is often used in clustering tasks when it is suspected that several groups are present in the data, but no assumption can be made on the exact boundary, either because it is unknown, or the structure of the data intrinsically exposes overlap (e.g. if animal behaviour is analysed). GMM has several advantages and is especially well-suited for clustering tasks where it is known that the data is approximately normally distributed among each group, since the Gaussian distribution is explicitly assumed in the model. Because it fits several distinct distributions with their own mean and covariance, it can deal with intertwining data groups or groups that have quite different shapes or distances between their respective data points. The noise within the data does not influence the result as much as in other 'hard clustering' methods. The drawbacks of GMM are that each step of the EM requires the computation of the covariance matrix of each distribution. This can be time-consuming, especially in cases of large datasets or high dimensionality. It is also prone to overfitting, and whilst it resolves some issues found in k-means, other issues remain, such as sensitivity to initialisation and the risk of reaching local optima during fitting. Overall, GMM is a more 'realistic' approach than k-means as it models the probability distribution over all n Gaussians but may complicate interpretation of results if discrete clusters are desired.

6.7 Anomaly detection: isolation forest

The Isolation Forest assumes that anomalies are few and different and has been used by Michelena et al. (2025) to detect oestrus. This underlying assumption leads to a reflection on anomaly detection – why focus on the numerous nominal data points when the goal is to highlight a few unusual ones? Liu et al. (2008) developed a simple algorithm aimed at isolating such points in the feature space, regardless of the nominal points. To do so, they introduced a building brick called the 'Isolation Tree', a binary tree that randomly partitions the data. At the root node, a feature is randomly selected, and a split value inside the data's range is also chosen randomly. Left and right subtrees are then created, based on whether the data point's value for the selected feature falls below or above the split value. The same algorithm is run recursively in the subtrees until a stopping criterion is met. Because anomalies are different, the path through the isolation tree from the root node to a leaf node is typically shorter for anomalies than for nominal nodes. In other words, identifying leaves at the end of short paths is a proxy to identifying anomalies. Due to the randomness involved in tree building, the process is repeated multiple times to obtain convergence for each record's path length. This results in an Isolation Forest – an ensemble of isolation trees that collectively contribute to the mean path length.

Tweaks have been highlighted by Liu et al. (2008), demonstrating that the algorithm is capable of detecting outliers from a restricted dataset size, and

since anomalies are defined by the shortest path, tree depth can be limited to decrease the computational load. These tweaks allow the method to achieve linear time complexity and memory usage proportional to the dataset size.

Liu et al. (2008) created a comparison between their method and other commonly used anomaly detection algorithms, with a focus on large datasets. The reported advantages of Isolation Forest included fast execution, high AUC scores, and suitability for high-dimensional data. However, the Isolation Forest works best alongside a feature selector to reduce dimensions.

Despite its efficiency and accuracy, Hariri et al. (2021) point out a limitation related to how the data is partitioned within Isolation Tree nodes, in terms of how the Isolation Forest determines anomaly scores in the whole space. In the initial method, the splitting was performed on one feature at a time, resulting in score regions aligned with the feature directions in the feature hyperspace. In a theoretical application to dairy cows, this would be akin to observing a biomarker diagnosis, one biomarker at a time, rather than considering the animal's holistic state, potentially biasing towards the most common illness diagnosis.

To circumvent this issue, Hariri et al. (2021) proposed extending the Isolation Forest by using a decision plane to split the data, allowing the borders of the regions to go in all directions in the hyperspace. Aside from this consideration, the algorithm maintains the same workflow. As a result, the accuracy of this new algorithm, deemed the Extended Isolation Forest, is slightly improved with minimal additional processing time and memory requirements.

7 Guidelines for validation and interpretation

7.1 Validation

Whilst UL is advantageous in that it does not require a response variable in the optimisation process, it unfortunately makes the validation process more challenging once an experiment is completed. In contrast to supervised learning, in which evaluation typically involves comparing predicted outputs to known target values (i.e. ground truth), UL relies solely on various metrics that reveal structures, patterns, or similarities within the data.

Because UL methods rely solely on the intrinsic structure of the data, they do not provide direct explanations for the discovered patterns. Whilst this lack of explicit explanation can be an issue as it is a black box, the fact that the model does not rely on *a priori* knowledge from the researcher is also a huge opportunity as it avoids confirmation bias. Outputs are driven entirely by the data structure.

If underlying causal information were available to the researcher, supervised learning would be more appropriate, particularly when the dataset

is sufficiently large. Consequently, evaluating UL outcomes is inherently more challenging, since ground truth labels are typically not available in the training data.

In addition, there is no universal agreement on key modelling decisions, such as the choice of distance metric or the number of clusters. Whilst similar issues exist in supervised learning, they are typically less pronounced due to the availability of ground truth, which provides clearer guidance when selecting and evaluating algorithms. As a result, the suitability of the algorithm and design choices largely depends on the characteristics of the data and its inherent heterogeneity. It is therefore common practice to apply multiple methods and parameter settings to a dataset, followed by evaluation and validation (Gao et al., 2023).

A common evaluation strategy is to assess the relevance and interpretability of the discovered clusters. This can be approached by:

- **External validation**, by comparing clusters with external information available in another dataset, quantities similar to a ground truth (e.g. diagnostic measurements on a subset of records), or metrics such as the Rand Index or the Jaccard Coefficient.
- **Internal validation**, by examining the relationships between clusters and the original input features used to generate them, often relying on goodness-of-fit indicators.
- **Relative validation**, by criteria that assess which model or set of parameters produces the most meaningful or high-quality clusters. A variety of measures have been proposed based on cluster compactness (homogeneity within clusters), separation (distance between clusters), representativeness (how well clusters reflect the underlying data distribution), connectedness (similarity between clustered points and their nearest neighbours), and stability (consistency of clustering results across different data subsets).

Combinations of these have also been suggested as evaluation criteria. Before interpreting the clustering results, however, it is essential to assess their reproducibility. In some cases, small perturbations in the underlying sample can lead to substantially different results, indicating low reliability.

Resampling and cross-validation are therefore integral to the validation process; for example, to choose the appropriate number of clusters, as many methods are sensitive to minor variations in the data and the choice of random seed, which increases the risk of overfitting (Gao et al., 2023).

Liu et al. (2022) highlight the concept of stability as a strategy for evaluating the performance and reproducibility of data clustering. The core idea involves generating perturbed datasets that remain close to the original, then applying

the clustering process. If the clustering is stable, the original groupings will be retained across these perturbed versions. In practice, this means that if the clustering process is repeated using a different (but similar) or sub-sampled original dataset, similar or comparable clustering outcomes should be obtained. It is worth mentioning that using a part of the same dataset as validation, whilst technically an external validation, is not entirely independent and does not guarantee that it will work on any similar dataset. Numerous methods have been developed for cluster stability, including resampling for stability estimation and cluster validation via data splitting and subsampling, as well as alternative methods that diverge from these classic approaches. Ensuring consistency is a key indicator of the reliability and robustness of UL results.

7.2 Interpretation

Whilst UL provides powerful and flexible tools for discovering hidden patterns within data, interpreting the results remains challenging. Nevertheless, it is possible to attempt an indirect inference of the potential causes behind the observed patterns by leveraging external knowledge or auxiliary datasets. The following ideas and procedures offer guidelines for interpretation and validation of results.

A useful starting point is the application of general descriptive statistics for cluster interpretation. For example, threshold comparisons grounded in scientific knowledge, or the distribution of data, can be informative. Aungier et al. (2015) used such comparisons relative to blood concentrations to characterise endocrine profiles.

Another approach involves calculating correlations between features to assess their 'proximity'. Most of the time, it implies a Pearson correlation (based purely on numerical values). If the data are not normally distributed, a common alternative is the Spearman correlation, which is based on ranked values. Van Hoeij et al. (2019) applied this to analyse the link between plasma metabolites and hormones.

A more advanced approach is the analysis of variance (ANOVA), which can be used to compare least-squares means of specific variables in between clusters. Ho et al. (2019) used ANOVA to assess whether populations differed with respect to different measured traits and traits predicted from FT-MIR. However, this approach is sensitive to dataset size, performing well on sufficiently large datasets but potentially resulting in excessive statistical power, increasing the likelihood of detecting statistically significant yet practically irrelevant effects.

To solve the issue for large-scale datasets, an accessible way to interpret clusters is to estimate the Least-Square-Means (LSM) of each cluster's variables

using a simple linear model. By calculating the LSM of each variable within each cluster, one can compare, for example, concentrations of chemical compounds or metabolites within and in-between clusters. By then judging the observed (or not observed) average differences and absolute average concentrations, the scientist can apply their specialist knowledge to understand the causal relationships between the clusters (and therefore the patterns in the data). After clustering, LSMs are calculated using the following formula:

$$LSM_j = \overline{Y} + \sum_{i=-1}^{k} \beta_i X_{ij}$$

where LSM_j is the Least-Square Mean for cluster
is the overall mean of the response variable Y,
are the estimated regression coefficients of variable
represents the presence (1) or absence (0) of a variable in a cluster

As the aim of the LSM calculation is to compare concentrations and/or their differences, it is advisable in this context to remove the intercept from the calculation of the linear model. Although this step may affects the statistical analysis, it greatly facilitates subsequent interpretation.

Interpreting results in high-dimensional spaces poses significant challenges, especially if results are to be presented graphically. To facilitate intuitive visualisation, it is often necessary to reduce the number of dimensions whilst preserving the interpretability and structure of the data. This process is closely related to dimensionality reduction, as discussed previously. Projecting the data onto feature axes or onto a latent space – such as that generated by PCA – can reveal meaningful patterns that are otherwise hidden in higher dimensions. For example, Brotzman et al. (2015) visualised clusters by projecting them onto the first two principal components of PCA, thereby assessing both the separation of clusters and their interrelationships. Similarly, Multidimensional Scaling (MDS) was employed by (Xu et al., 2020) to convert pairwise distances into a Cartesian coordinate system, facilitating intuitive cluster interpretation. As highlighted in the previous section, t-SNE can also reveal apparent clusters within a dataset. However, its use is not without controversy – studies such as (Wattenberg et al., 2016) caution against a direct interpretation of the clusters, and Linderman and Steinerberger (2017) emphasize the need for careful analysis to avoid misrepresentation of the underlying structure. Despite these concerns, t-SNE can produce meaningful visualisations under certain conditions. A more recent alternative, UMAP retains the advantages of t-SNE whilst offering improved preservation of the global structure of the data. This enables a more accurate and holistic understanding of the dataset's architecture, thereby enhancing the interpretability of the clustering results.

Published by Burleigh Dodds Science Publishing Limited, 2026.

A newer methodology for interpretation and validation involves leveraging the capabilities of LLMs. These models, which have demonstrated impressive performance across a wide range of natural language processing tasks, are increasingly being explored for their potential to support UL objectives, including semantic clustering, feature extraction, and representation learning. Their ability to capture complex contextual relationships in high-dimensional textual data offers new opportunities for data interpretation and pattern discovery, particularly in domains where labelled data are scarce. Whilst the expertise of scientists remains the go-to gold standard for evaluation of scientific results, some results can be so abstract that interpretation becomes an arduous task. In addition, expert interpretation often requires years of experience and personal insight, which may only be applicable to a limited number of specialists in a respective field of science. Both issues can be eased by using LLMs. Since modern LLMs are trained on incredibly large bodies of data, including scientific papers, they can identify patterns related to real-world experiments. Although they remain, to an extent, a 'black box', particularly when it comes to questions such as, 'How does an LLM store its information?' and 'Where does the reasoning originate?' (Prince, 2024), factual evidence shows their incredible capabilities, even in terms of 'understanding' quite complex and abstract tasks.

To enable LLMs to reason about outputs from UL, it is necessary to translate cluster patterns into a structured format that they can interpret. One effective approach is to use the LSM method, which facilitates the transformation of identified clusters into a form that resembles the types of data found within the pre-training corpus of most LLMs. The rationale for using LSM in this context is clear. In line with good scientific practices, results are often summarised in tables presenting descriptive statistics such as means and standard deviations for the variables of interest. These forms of representation are not only human but also machine-readable, particularly by LLMs trained to process and analyse both natural language and quantitative data. As such, LLMs are well-suited to assist in the interpretation of LSM-derived summaries and in deriving domain-specific insights.

The procedure for querying an LLM using UL outputs involves three essential steps. The first and most crucial is to define a clear and precise context for the task. This includes specifying the scientific domain (e.g. animal science, agronomy, dairy production), detailing the experimental setup, including variables measured, timeframes, and population characteristics, and providing any contextual information that may inform the interpretation of patterns. This step is fundamental to limiting the degrees of freedom available to the LLM, thereby enhancing the relevance, accuracy, and domain alignment of its outputs. Constraining the model's scope improves both the reliability and interpretability of the resulting explanations. Next, a clear and explicit task must

be posed to the LLM. An example of an effective prompt is: *Please interpret the anonymized results of our experiment and explain what the findings mean for a dairy farmer. In addition, suggest practical actions the farmer could take to address potentially unfavourable situations.* This step helps to ensure that the LLM provides contextualised, actionable, and domain-specific interpretations rather than generic summaries. This would benefit from LLMs trained for animal science, such as the model that Liu et al. (2025) developed. By articulating a specific intent, this prompt structure improves the model's capacity to generate meaningful reasoning and recommendations. Finally, the results should be presented in a concise and structured format. This may involve a simple table that includes the list of variables, the respective cluster identifiers, and the corresponding statistical estimates, such as mean, standard deviation, or other statistical descriptors. By following these three steps (defining the framework, formulating a targeted query, and structuring the data appropriately), researchers can effectively harness LLMs to derive interpretable and actionable insights from UL outputs.

8 Combining unsupervised learning and knowledge discovery in databases

We have already identified the different categories of algorithms that can be used to develop UL. It is important to mention where these algorithms fit in a classical data pipeline, also known as KDD. This concept can be divided into five essential steps (Fig. 8) starting with 'step 0', which consists of precisely defining the objective of the study. Success is not only linked to the choice of algorithm but also to many decisions made before and after the data mining process. Figure 7 illustrates examples of operations available at each KDD step. This section reviews those steps and emphasises the use of UL algorithms, enabling discussion of certain aspects of data analysis not previously covered. In particular, the combination of SL and UL opens up many new data processing approaches. This section aims to show how UL algorithms are integrated into the process and how they complement traditional SL methods.

8.1 Selection

The first step is selection. At this stage, the priority is to integrate data from different sources and databases. ML models often treat datasets as tabular, with each record occupying one line and each column representing the monitored features. For more complex datasets, some algorithms relax this concept of 'one-line-equals-one-record' by replacing the 'line' with a processing stream, e.g. photos in image processing.

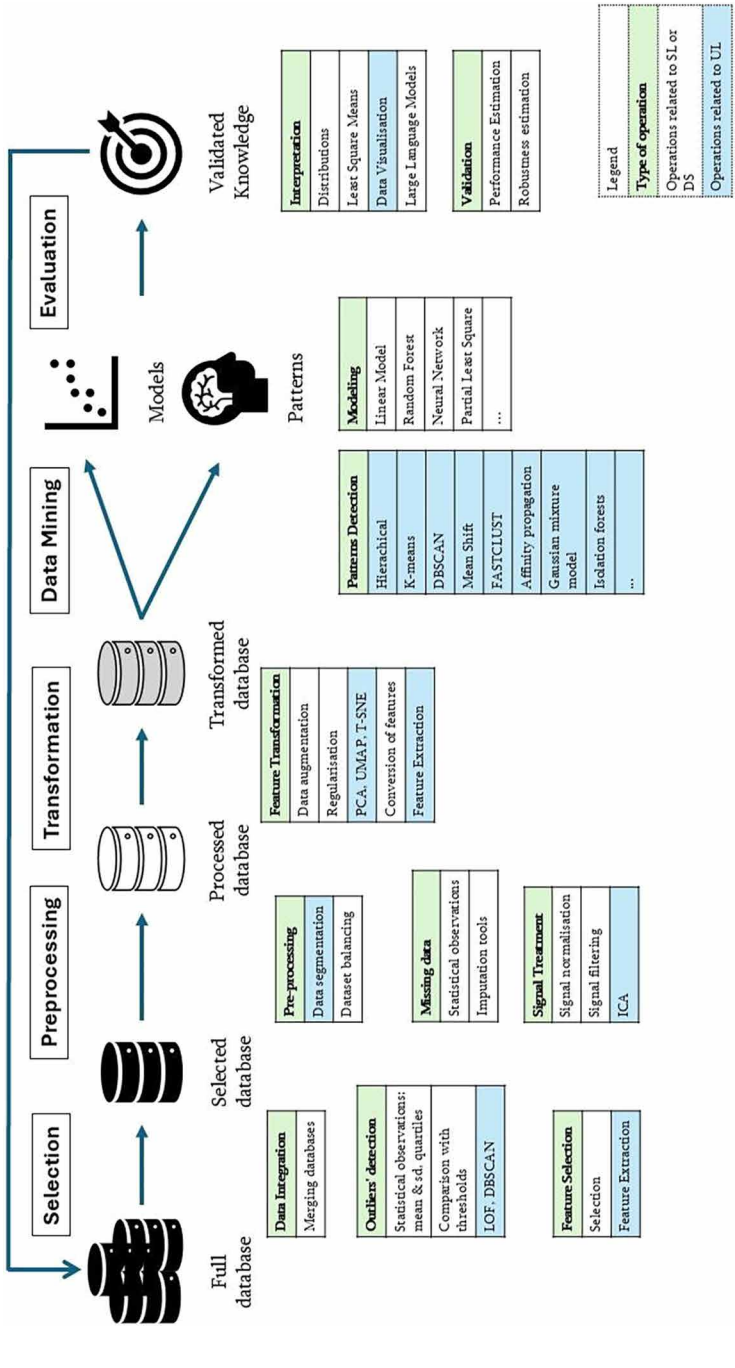

Figure 8 KDD pipelines with examples of operations available at each step.

The next task is selecting the variables (or derived features) necessary for the study. This requires a precise inventory of the available variables. Some variables may be unusable due to poor reliability or a high proportion of missing data. At this stage, domain knowledge or consultation with experts is crucial to identify the most relevant variables. Feature selection and extraction algorithms can also be useful.

Finally, outlier detection is essential to ensure that algorithms operate 'within specifications'. Outliers are values that are extremely high or low relative to the expectation from the underlying distribution and may therefore be irrelevant to the study.

These extreme values can cause several issues in the modelling and/or interpretation of results. Many algorithms – particularly those based on the Gaussian distribution – are sensitive to extreme or outlying values. Whilst this sensitivity is not a weakness itself (as it often allows for the detection of subtle data structures and can in fact improve model fit), it becomes problematic when the data contains a substantial number of outliers. In such cases, results may be misleading or less accurate, as the presence of outliers violates the underlying statistical assumptions.

This highlights a fundamental tension between algorithmic strengths and vulnerabilities in terms of the introduction of errors. Features that make an algorithm powerful can also render it fragile, depending on the context. Finding an optimal combination of data quality/properties and Matching algorithm specifications is therefore essential to minimise overall errors by accounting for such trade-offs. However, formalising such a selection process is methodologically challenging since it is hard to see which data points are outliers and why, and if the resulting interactions are harmful. The choice of strategy frequently relies on the statistician's intuition and experience.

In general, outlier detection methods often employ statistical indicators such as quantiles of a distribution or fixed thresholds of 'good' data points. UL anomaly detection algorithms, such as LOF (Satoła and Satoła, 2024) or DBSCAN (Shine et al., 2018), can also prove effective.

8.2 Preprocessing and transformation

These KDD steps are related but differ in their objectives. Preprocessing focuses on cleaning the dataset of its potential inconsistencies, whilst transformation optimises the dataset for the data mining stage. Commonly available preprocessing operations include noise treatment (particularly in signal treatment) and handling missing data. UL is less commonly used here, although an algorithm such as ICA can extract a specific signal from multiple merged signals. For missing data, several strategies can be carried out, such as deleting lines where data is missing, replacing the missing data with a variable

such as the mean, median, or most represented class, or by using any other imputation method such as k-Nearest Neighbour.

The transformation step is related to feature type conversion, scaling or the normalisation of the features, and dimensionality reduction. Some algorithms cannot handle categorical features, requiring conversion into numerical features. Vanlierde et al. (2021) used the conversion of categorical features (breed) into dummy features. In other cases, the opposite process is more beneficial, i.e. converting numerical into categorical features. Grelet et al. (2024) highlighted that, for some key biomarkers predicted from milk, quantitative performances are disappointing although trends appear to be modelled. For biomarkers that display highly skewed distribution, it can be necessary to transform quantitative measures into levels and perform discriminant analysis on these categories. However, as mentioned, this is only relevant for management purposes that require levels instead of precise values.

For feature transformation, it is common to observe features that are not uniformly or normally distributed along the range of values. A way to bypass the associated modelling complexity is to transform the features. Ghaffari et al. (2020) used log10 transformation of blood metabolite concentration, followed by a Pareto scaling for heteroscedasticity correction through skewness reduction. It is also possible to use a natural logarithm to transform input feature distribution into a normal distribution. A typical example used in milking studies is somatic cell count (SCC). As this is a marker of inflammatory response, it often explodes and thus there is a need to rationalise it, with for example the application of a log transform, or the computation of a somatic cell score, as demonstrated by Satoła and Satoła (2024).

Dimensionality reduction is a final key objective, and UL is relevant with feature extraction algorithms and dimensionality reduction algorithms. In tabular datasets, features might be interpreted as dimensions of a multi-dimensional space. As some algorithms are less efficient in highly dimensional datasets – known as 'the curse of dimensionality' – aggregating the dimensions is beneficial. This curse might be defined as the likelihood of finding observations alone in single subspaces of a manifold, categorising the lonely observation as the 'typical example' for that part of the hyperspace, and thus training the algorithm to be too specific. Some methods commonly seen in the literature include:

- PCA – this consists of linearly combining the features to create a new set of orthogonal coordinates, capturing the maximum variation in the data. Brotzman et al. (2015) used it for feature reduction before performing clustering, whilst Todde et al. (2016) used it to reduce the feature sets before k-means clustering.

- MDS – this consists of arranging points in a space based on pairwise similarities or dissimilarities of entities. The rank order of distances should match the rank order of dissimilarities. It is similar to PCA but adapted to also account for categorical data. In practice, it does not scale well on large datasets and therefore its use is restricted to simpler tasks.
- T-SNE – a non-linear dimensionality reduction technique based on probability distribution and paired comparison of high-dimensional objects. It is poor at conserving the global structure of the data and should therefore be used only for tasks that focus on local characteristics of the dataset.
- U-MAP – this is a variation of t-SNE, based on constraints on the original features (McInnes et al., 2020). Simankov et al. (2025) define it as an algorithm aiming to find a low-dimensional representation of a dataset whilst preserving its global structure by first constructing a high-dimensional graph and then, through embedding optimisation, creating a lower dimensional graph that maintains the pairwise similarities.
- Variational autoencoders (VAE) – based on neural networks, these are another way to aggregate feature information and are fed the original information as the information to process and the information modelled. This mirroring effect is enabled through multiple layers and can be seen as a symmetrical hourglass, with one end being the original data, and the other a 'perfect' reproduction of this original data, with funnels going both ways. The tightest point (the bottleneck) is where all the information can be modelled and summarised. After training it for perfect reproduction, the enlarging funnel is removed from the model and the predictions created by the 'bottleneck' are considered to be the new summarised record score in a new latent space.

8.3 Data mining: getting the best of the two approaches

Whilst the other steps are essential for building a relevant model, this is the step in which knowledge is extracted from the database. Although the data mining step is itself a full process inside the KDD, it can be summarised in two parts. The first part is modelling, i.e. supervised learning (SL), where models are used either for better understanding of a process or to make predictions. Many supervised algorithms are available, and information about the structure of the dataset or the response variable is required to choose the most appropriate one. This is covered in the section on supervised learning. The second part is pattern recognition, i.e. UL, which primarily consists of different clustering approaches. These approaches have been covered in a previous section and will not be discussed here. However, this subsection highlights an aspect that is often overlooked – although SL and UL are carried out separately, they can be combined sequentially to strengthen the

general approach. For example, in a large database, patterns may be identified but not validated. These patterns can then be predicted using a supervised model and applied to a new database where reference values related to the interpreted patterns are available. Combining UL and SL is a powerful approach, especially useful given the increasing volume of data generated today. Conversely, moving from supervised to unsupervised approaches can reveal significant differences in prediction performance in a model created by underlying groups not identified beforehand. These combinations, often referred to as stacking, are presented in the following case studies. The final step of KDD, evaluation, has been discussed in the section on UL and will not be further developed here. The evaluation of supervised models, often referred to as validation, is covered in the chapter on supervised modelling.

9 Case studies in dairy farming

Having explored the different methodologies, this section illustrates concrete examples of UL applications in dairy farming. Although this chapter focuses primarily on UL, very few studies rely exclusively on UL algorithms (denoted by a 'U' before the example). It is widespread practice to include supervised algorithms (examples beginning with 'U+S') within processing pipelines. Because UL does not rely on reference outputs, there are no accuracy/transferability measurement metrics to report for studies using only UL. In contrast, such metrics can be reported for U+S studies.

In terms of stacking, the most common approach consists of first detecting patterns using UL and then developing a supervised model to infer these patterns in new datasets, enabling the possibility of near-real-time application. For example, Xu et al. (2019) identified the metabolic status of Holstein-Friesian cows using k-means on biomarkers such as free fatty acids, glucose, β-hydroxybutyrate, insulin and insulin growth factor 1, which are costly to measure. They then predicted these statuses using supervised ML algorithms on cheaper on-farm data such as dry period length, parity and milk production traits. In this study, the output was a classification, but it could also be expressed as the probability of belonging to a specific cluster.

Another use case for stacking unsupervised and supervised learning is the interpretation of complex processes. In such cases, clustering is not the final objective but rather a means of understanding the underlying process. For example, Sowmya and Ponnusamy (2021) combined supervised and UL to detect milk adulteration using spectrometry. K-means were used to identify emerging groups and assess whether they were associated with adulteration, before predicting the adulteration using supervised ML. Beyond group identification, a predictive supervised algorithm enables transferability to new records, as shown by Soyeurt et al. (2022), where PLS- DA was applied

after clustering to infer cluster membership in new populations. Stacking is not limited to unsupervised algorithms followed by supervised applications. It can be mixed for pre-processing, using a multistep approach. Foldager et al. (2020), for example, used random forest (RF) algorithms to predict blood energy-related bioindicators (such as NEFA, cholesterol, etc.) from FT-MIR milk analysis in dairy cows. Based on these predicted values, energy balance clusters were created using a k-means algorithm and subsequently used to predict physiological imbalances.

The objective of this section is to present case studies in which the methodologies described above were applied to solve challenges in PLF.

9.1 Biomarker focused studies

'U+S' Franceschini et al. (2022) utilized 740 454 milk recording events from 114,536 primiparous Holstein cows to detect patterns in FT-MIR predicted traits, including body weight, minerals, β-hydroxybutyrate (BHB), citrate and acetone. Pattern detection was carried out using Ward hierarchical clustering, before applying partial least square discriminant analysis (PLS-DA) to predict the identified clusters. One of these clusters was associated with metabolic disorders, demonstrating that FT-MIR spectra can be used as an individual monitoring tool. A similar methodology was applied to 774 781 bulk tank milk fatty acid profiles to detect potential herd-level issues, benefiting from a higher data frequency (Franceschini et al., 2025). Although no labels for metabolic disorders were available, these studies demonstrate that strong signals exist in milk spectra. However, these studies are limited by the lack of direct feedback from farmers.

'U+S' Tremblay et al. (2018) acquired 790 observations from 312 cows between May 2015 and February 2016 to detect poor metabolic adaptation syndrome (PMAS). They used blood BHB, milk fat to protein ratio, and blood non-esterified fatty acids (NEFA) (although additional features, such as FT-MIR spectra, were available), and performed a PCA on these features and then performed a k-means cluster analysis on the first two PCs. Five clusters appeared, distinguishing different levels of PMAS, whilst additional features were used for cluster interpretation. Longer-term perspectives included the use of these clusters to develop predictive models for PMAS and to improve individual and herd metabolic health management. Regarding the validation strategy, they did not partition data beyond dataset filtering.

9.2 Health disorder-focused studies

'U+S' Ghaffari et al. (2020) primarily used supervised classifying algorithms – sequential minimal optimization, RF, alternating decision tree, and naïve

Bayes-updatable – to detect over-conditioning in dairy cows around calving, using blood metabolomic profiles. All algorithms achieved accuracies over 70%. The unsupervised component of the study relates to the clustering of the variables (i.e. the serum metabolites) using Ward Linkage for the heatmaps. Although the study applied mostly supervised learning, it illustrates that UL can be used in very specific use cases to better understand the relation between metabolites.

'U+S' Van Hoeij et al. (2019) used SAS PROC Cluster, PROC CORR, and PROC TREE to investigate the relationship between metabolic status and behaviour in 81 postpartum Holstein-Friesian cows, using computerised feeders and accelerometers. The study consisted of a Ward clustering using blood metabolite concentration (e.g. free fatty acids, BHB, glucose, IGF-1, GH) to define metabolic status; a correlation analysis between plasma metabolites and the clusters to validate those clusters; and the creation of a GLM to assess the plasma concentration of metabolites or behaviour based on integration into one of the metabolic status clusters, the dry period length, and the parity (capped at three). The study highlighted differences in behaviour between cows with better metabolic status and those with poorer status.

'U+S' Oehm et al. (2022) analysed 606 farms in Germany to detect parasites in dairy cows, using UL to avoid introducing bias. Based on a stratified sampling of farms across three regions and herd sizes occurring between 2016 and 2019, they performed clustering based on production traits in addition to antibody status using partitioning around medoids. Clustering was carried out separately for each of the study regions and each of the parasites. They successfully identified bulk tank milk contaminated by at least one of two parasites (*F. hepatica* & *O. ostertagi*). Random forest was then used to assess variable importance and better understand the underlying process.

To study hoof disorders in dairy cattle in Korea, Espiritu et al. (2025) collected data from 877 cattle across 15 farms. The 15 farms were clustered based on farm management practices, distinguishing two groups – one with higher disease prevalence and poor conditions, and the other with lower prevalence and better management. The study utilised Gower's Distance and Ward's linkage method. They then computed the correlation between the different management practices and the hoof diseases to validate the hypothesis, demonstrating that floor elevation transitions, aggressive trimming, shallow bedding, and wet flooring strongly correlated with hoof disorders.

'U+S' Kiouvrekis et al. (2024) carried out detection of subclinical mastitis in dairy sheep (2220 ewes across 110 farms) in Greece. They associated the prevalence of subclinical mastitis in farms with management, temporal and meteorological data. Assuming a bi-modality regarding the affected and non-affected populations, they explored the application of affinity propagation, birch, hierarchical, k-means and spectral clustering. On top of this detection,

they also explored supervised algorithms such as decision trees and k-nearest neighbours. The study highlighted that farmers were not always aware of the health status of their sheep, with the ultimate goal of improving awareness and increasing farmers' attention to subclinical mastitis.

9.3 Detection of heat (oestrus)

'U' Shahriar et al. (2015) used fast Fourier transform and k-means clustering to analyse accelerometer data acquired from the collars of 19 cows between October and December 2013, aiming to predict heat events in pasture-based dairy cows. Regarding the accelerometer data, mobility was defined as the sum of absolute accelerations along each axis, whilst movement intensity was defined as the square root of the sum of squared accelerations. Longer-term perspectives included the refinement of oestrus detection methods, particularly by reducing false positives to enable farm management deployment. Regarding the validation strategy, they did not partition data beyond the dataset filtering.

'U' Toan et al. (2019) combined Internet of Things technologies and ML with data acquired from a low-frequency neck-mounted accelerometer to detect oestrus. They applied UL alongside a novel heuristic post-processing algorithm to address inaccuracies caused by the low frequency. A GMM was trained on synthetic data, improving precision from 0.28 to 0.94, compared to a traditional threshold model. They then applied the model to seven cows as a proof of concept, noting that further labelled data are required.

Michelena et al. (2025) applied three unsupervised anomaly detection techniques – DBSCAN, Isolation Forest and Local Outlier Factor – to sensor data for oestrus detection in dairy cattle. The data consisted of five behaviours – resting, rumination, grazing, eating, and playing. Using 22 Holstein–Friesian cows from two dairy farms, the study found that cow-level models achieved F1-Scores above 90%, whilst herd-level models achieved 62.22% and 75.27%. These results suggest that individualised models are essential for improving performance in anomaly detection.

9.4 Cow behaviour

'U' McVey et al. (2020) focused on animal behaviour monitoring using ear-tag accelerometers and milking order. The study analysed data over a six-month period for 200 Holstein cows. Based solely on milking order, three main groups emerged – cows at the beginning of the queue, those at the end, and those moving around the middle. The findings were consistent with the ear-tag data, in terms of segmentation of the population into three sub-groups.

'U+S' Yujuan et al. (2017) monitored cow rumination during July 2013 using the application of mean-shift to 65 400 dairy cow images, combined with

jaw tracking. However, this rumination monitoring approach was hindered by behaviours such as raising or turning of the cows' heads.

9.5 Livestock monitoring: herd level

'U' Bonora et al. (2018) developed and tested a process stack to analyse the various time series recorded in AMS of dairy cows. Their study focused on a single farm with 88 cows. Data related to cows, milk production and cow collar accelerometers were monitored from June to September 2015. Their end goal was to provide automatic grouping of the cows, based on production and behavioural features, to support livestock farm management. To achieve this, they performed multiple k-means clustering and, using graph representation, created networks of cows. This approach enabled them to highlight three clusters of cows in terms of productive characteristics. As the study was primarily exploratory and group detection oriented, all data entered the pipeline without the use of an external validation set.

'U' Achour et al. (2019) monitored the activity of dairy cows using data acquired via an inertial measurement unit (accelerometers combined with gyroscopes and magnetometers) attached to the back of eight dairy cows during December 2017. They used data related to three cows for model development and all data for validation. To monitor cow behaviour and better correlate behaviour with signals, they used an 'automaton' – a pattern validator with nodes representing the states and the possible transitions validated, or, if forbidden, indicating a problem in the expected behavioural pattern. Building on this pattern detection, they developed a decision tree to classify behaviours into crisp clusters, reporting accuracies close to 98–99%.

'U' Xu et al. (2020) used location monitoring from ten cattle yearlings to identify the social structure of a small group via UL (k-means and hierarchical clustering), applied on a conversion of their dataset based on MDS. More specifically, the study focused on 'affinity', defined as a measure derived from the ranking of pairwise distance. This methodology enabled a better understanding of social interactions within the group.

'U+S' Ismail et al. (2019) analysed images of dairy cows (no information on the study period was available), using 800 images for calibration and 200 for validation. Their processing stack included image resizing and grey-scaling, applying region-based convolutional neural networks to detect cows. They then explored four clustering methods – mixture model, hierarchical clustering, k-means, and DBSCAN – in order to distinguish herds and identify outliers. Hyper-parameters were explored (notably epsilon and k) and the impact of Euclidean and Manhattan distance was also studied regarding the formation of clusters.

'U+S' To identify grass-fed cows in 2449 Walloon farms in Belgium, Soyeurt et al. (2022) used the MIR spectra alongside the date of milk payment analysis, as most Walloon cows are in pasture during the summer. They predicted the grass-fed cows using PLS-DA and then studied the average value per farm/year to discover patterns in management practices.

'U+S' Kogo (2016) presented in their master's thesis an automated approach to detect the presence of outliers in feed intake data, recorded from dairy cows from late August 2015 to late January 2016. This resulted in a dataset of 180 372 observations (primarily time-related) of 78 cows. The analysis was based on a GMM.

'U+S' Alessio et al. (2021) used a combination of methods to determine relationships in bulk tank milk composition, analysing 154 781 samples from southern Brazil. Once clusters were defined through a PROC FASTCLUS, a PROC STEPDISC was used to understand which variables most strongly discriminated between clusters. Their findings indicated that an increase in somatic cell count and total bacterial count is associated with a decrease in lactose.

9.6 Classification of farms/respondents to surveys

'U' Todde et al. (2016) characterized 285 Italian dairy cow farms through the use of k-means on PCA projected features, reflecting general information such as herd size, animal categories, land use and ownership, milk quality and production. More precisely, the study aimed to identify gradients of insensitivity, mechanisation and structurality, based on detailed descriptions of cultivated crops, farm structures, equipment, and machinery.

'U' Hyland et al. (2018) applied hierarchical clustering based on PCA scores to analyse factors influencing dairy farmers' adoption of best management practices. Data were collected between May and August. They identified groups among 256 dairy farmers with varying levels of use of a management tool – the Spring Rotation Planner (SRP). They highlighted two main groups within their dataset – low adopters, characterised by a strong sense of resource constraints, and high adopters.

'U' Vázquez-González et al. (2024) used hierarchical clustering combined with principal component factor analysis to analyse 86 survey responses, with the aim of classifying cow dairy farms in Cantabria between November 2016 and February 2017. The resulting topology was based on six pillars – milk production, land base, livestock, facilities and machinery, family labour and workforce, and economic factors. This approach identified four main production typologies, differing in levels of agricultural activity, profitability and diversification, along with a fifth 'group' comprising singular cases.

'U+S' Brotzman et al. (2015) explored trends in performance characteristics across 4190 US dairy herds (3078 after filtering, all used together without

partitioning). Based on milk composition, cow description and herd repartition, they applied PCA to 22 standardised features and identified the 16 variables most strongly associated with the first six principal components. These entered the cluster analysis consisting of Ward's partitioning method and Fourier plots. They identified six herd groups differentiated by performance, infection risk, inter-lactation timing and transitioning. A key perspective highlighted in the study is the relationship between herd management and performance cluster attribution.

10 Conclusion and future trends

Compared with supervised learning, relatively few studies focus exclusively on the application of UL. This may reflect the common perception that a model must include a target variable to be practically useful. However, as demonstrated throughout this chapter, UL offers considerable potential – not only for discovering novel patterns in data but also for improving data quality through cleaning, transforming features to reduce dimensionality, and facilitating the visualisation of both inputs and outputs.

As highlighted, when large databases are available, UL can be especially powerful – particularly when combined with supervised learning – to generate actionable insights for dairy farmers and the broader dairy industry. Such large-scale data sources are now more accessible than ever, and the required computational resources are widely available, enabling the extraction of meaningful knowledge at scale. In our view, further progress will depend on the integration of heterogeneous data sources, which could unlock new opportunities for precision dairy farming and innovation within the dairy value chain. Importantly, this perspective is not limited to dairy systems and can be extended to other types of livestock and agricultural production systems.

This chapter has aimed to provide an overview of the main techniques in UL and their practical applications in dairy science. However, given the breadth and complexity of the subject, it is not possible to cover all aspects comprehensively within a single chapter. In addition to the key references cited herein, readers seeking further detail are encouraged to consult the broader literature for deeper insights into specific methods and use cases.

11 Where to look for further information

For a longer explanation about UL and the fundamental theory behind it, the books by Hinton and Sejnowski (1999) and Bishop (2006) are highly relevant references:

Hinton, G. and Sejnowski, T.J. 1999. *Unsupervised Learning: Foundations of Neural Computation*. MIT Press, Cambridge. (Computational Neuroscience Ser).

Bishop, C.M. 2006. *Pattern Recognition and Machine Learning*. Springer, New York. (Information science and statistics).

For more information about the practical implementation in Python, the book by Patel (2019) and the documentation of Scikit-learn Python Package are of great use:

Patel, A.A. 2019. *Hands-on unsupervised learning using Python: how to build applied machine learning solutions from unlabeled data.* 1st edition. O'Reilly Media, Sebastopol.

Pedregosa, F., et al. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.

12 Acknowledgement

The authors would like to thank the EU Interreg NWE Holicow project for their funding.

13 References

Achour, B., et al. 2019. Unsupervised automated monitoring of dairy cows' behavior based on inertial measurement unit attached to their back. *Computers and Electronics in Agriculture* 167, 105068. https://doi.org/10.1016/j.compag.2019.105068.

Agarwal, P., et al. 2011. Issues, challenges and tools of clustering algorithms. https://doi.org/10.48550/ARXIV.1110.2610.

Agrawal, R., et al. 1998. Automatic subspace clustering of high dimensional data for data mining applications, In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98. Association for Computing Machinery, New York, NY, USA, pp. 94–105. https://doi.org/10.1145/276304.276314.

Alessio, D.R.M., et al. 2021. Lactose and its relationship with other milk constituents, somatic cell count, and total bacterial count. *Livestock Science* 252, 104678. https://doi.org/10.1016/j.livsci.2021.104678.

Alghushairy, O., et al. 2021. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing* 5, 1. https://doi.org/10.3390/bdcc5010001.

Anderberg, M.R. 1973. *Cluster Analysis for Applications, Probability and Mathematical Statistics: A Series of Monographs and Textbooks*. Elsevier Science, Burlington.

Arthur, D. and Vassilvitskii, S. 2006. How slow is the k-means method?, In: Proceedings of the Twenty-Second Annual Symposium on Computational Geometry. Presented at the SoCG06: 22nd Annual Symposium on Computational Geometry, ACM, Sedona Arizona USA, pp. 144–153. https://doi.org/10.1145/1137856.1137880.

Aungier, S.P.M., et al. 2015. The relationship between activity clusters detected by an automatic activity monitor and endocrine changes during the periestrous period in lactating dairy cows. .*Journal of Dairy Science* 98, 1666–1684. https://doi.org/10.3168/jds.2013-7405.

Blei, D.M., et al. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993.

Bock, H.-H. 2007. Clustering methods: a history of k-means algorithms, In: Brito, P., et al. (Eds.), *Selected Contributions in Data Analysis and Classification, Studies in Classification, Data Analysis, and Knowledge Organization*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 161–172. https://doi.org/10.1007/978-3-540-73560-1_15.

Bonilla, J., et al. 2021. Fuzzy clustering methods with Rényi relative entropy and cluster size. *Mathematics* 9, 1423. https://doi.org/10.3390/math9121423.

Bonora, F., et al. 2018. A cluster-graph model for herd characterisation in dairy farms equipped with an automatic milking system. *Biosystems Engineering* 167, 1–7. https://doi.org/10.1016/j.biosystemseng.2017.12.007.

Brotzman, R.L., et al. 2015. Cluster analysis of dairy herd improvement data to discover trends in performance characteristics in large Upper Midwest dairy herds. *Journal of Dairy Science* 98, 3059–3070. https://doi.org/10.3168/jds.2014-8369.

Chander, S. and Vijaya, P. 2021. Unsupervised learning methods for data clustering, In: Binu, D. and Rajakumar, B.R. (Eds.), *Artificial Intelligence in Data Mining*. Elsevier, San Diego, pp. 41–64. https://doi.org/10.1016/B978-0-12-820601-0.00002-1.

Comaniciu, D. and Meer, P. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 603–619. https://doi.org/10.1109/34.1000236.

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2, 303–314. https://doi.org/10.1007/BF02551274.

De Maesschalck, R., et al. 2000. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems* 50, 1–18. https://doi.org/10.1016/S0169-7439(99)00047-7.

Dempster, A.P., et al. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

Espiritu, H.M., et al. 2025. Surveillance of hoof disorders in Korean dairy cattle and the correlation of farm condition risk factors to their prevalence. *BMC Veterinary Research* 21, 169. https://doi.org/10.1186/s12917-025-04628-9.

Ester, M., et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96. AAAI Press, Portland, Oregon, pp. 226–231.

Ezugwu, A.E. 2020. Nature-inspired metaheuristic techniques for automatic clustering: a survey and performance study. *SN Applied Sciences* 2, 273. https://doi.org/10.1007/s42452-020-2073-0.

Ezugwu, A.E., et al. 2022. A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence* 110, 104743. https://doi.org/10.1016/j.engappai.2022.104743.

Foldager, L., et al. 2020. Predicting physiological imbalance in Holstein dairy cows by three different sets of milk biomarkers. *Preventive Veterinary Medicine* 179, 105006. https://doi.org/10.1016/j.prevetmed.2020.105006.

Fraley, C. and Raftery, A.E. 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal* 41, 578–588. https://doi.org/10.1093/comjnl/41.8.578.

Franceschini, S., et al. 2022. Can unsupervised learning methods applied to milk recording big data provide new insights into dairy cow health? *Journal of Dairy Science* 105, 6760–6772. https://doi.org/10.3168/jds.2022-21975.

Franceschini, S., et al. 2025. Detection of dairy herd management issues using fatty acid profiles predicted by mid-infrared spectrometry. *Animals* 15, 1575. https://doi.org/10.3390/ani15111575.

Fred, A.L.N. and Leitão, J.M.N. 2000. Partitional vs hierarchical clustering using a minimum grammar complexity approach, In: Ferri, F.J., et al. (Eds.), *Advances in Pattern Recognition, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 193–202. https://doi.org/10.1007/3-540-44522-6_20.

Frey, B.J. and Dueck, D. 2007. Clustering by passing messages between data points. *Science* 315, 972–976. https://doi.org/10.1126/science.1136800.

Fukunaga, K. and Hostetler, L. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory* 21, 32–40. https://doi.org/10.1109/TIT.1975.1055330.

Gao, C.X., et al. 2023. An overview of clustering methods with guidelines for application in mental health research. *Psychiatry Research* 327, 115265. https://doi.org/10.1016/j.psychres.2023.115265.

Ghaffari, M.H., et al. 2020. Discovery of different metabotypes in overconditioned dairy cows by means of machine learning. *Journal of Dairy Science* 103, 9604–9619. https://doi.org/10.3168/jds.2020-18661.

Ghahramani, Z. 2004. Unsupervised learning, In: Bousquet, O., et al. (Eds.), *Advanced Lectures on Machine Learning, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 72–112. https://doi.org/10.1007/978-3-540-28650-9_5.

Greenacre, M., et al. 2022. Principal component analysis. *Nature Reviews Methods Primers* 2, 1–21. https://doi.org/10.1038/s43586-022-00184-w.

Grelet, C., et al. 2024. Prediction of key milk biomarkers in dairy cows through milk mid-infrared spectra and international collaborations. *Journal of Dairy Science* 107, 1669–1684. https://doi.org/10.3168/jds.2023-23843.

Guha, S., et al. 1998. CURE: an efficient clustering algorithm for large databases. *ACM SIGMOD Record* 27, 73–84. https://doi.org/10.1145/276305.276312.

Hariri, S., et al. 2021. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering* 33, 1479–1489. https://doi.org/10.1109/TKDE.2019.2947676.

Hartigan, J.A. 1975. *Clustering Algorithms, A Wiley Publication in Applied Statistics*. Wiley, New York.

Ho, P.N., et al. 2019. Classifying the fertility of dairy cows using milk mid-infrared spectroscopy. *Journal of Dairy Science* 102, 10460–10470. https://doi.org/10.3168/jds.2019-16412.

Hongbo, L.I., et al. 2024. Seedling stage corn line detection method based on improved YOLOv8. *Smart Agriculture* 6(6), 72. https://doi.org/10.12133/j.smartag.SA202408008.

Hyland, J.J., et al. 2018. Factors influencing dairy farmers' adoption of best management grazing practices. *Land Use Policy* 78, 562–571. https://doi.org/10.1016/j.landusepol.2018.07.006.

Ismail, Z.H., et al. 2019. Efficient herd – Outlier detection in livestock monitoring system based on density – Based spatial clustering. *IEEE Access* 7, 175062–175070. https://doi.org/10.1109/ACCESS.2019.2952912.

Izenman, A.J. 2012. Introduction to manifold learning. *WIREs Computational Statistics* 4, 439–446. https://doi.org/10.1002/wics.1222.

Jain, A.K., et al. 1999. Data clustering: a review. *ACM Computing Surveys* 31, 264–323. https://doi.org/10.1145/331499.331504.

James, G., et al. 2023. Unsupervised learning, In: *An Introduction to Statistical Learning, Springer Texts in Statistics*. Springer International Publishing, Cham, pp. 503–556. https://doi.org/10.1007/978-3-031-38747-0_12.

Kaufman, L. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis, Wiley Series in Probability and Mathematical Statistics*. Wiley, Hoboken, NJ.

Kiouvrekis, Y., et al. 2024. The use of machine learning to predict prevalence of subclinical mastitis in dairy sheep farms. *Animals* 14(16), 2295. https://doi.org/10.3390/ani14162295.

Kogo, G. 2016. *Analyzing Automatic Cow Recordings to Detect the Presence of Outliers in Feed Intake Data Recorded from Dairy Cows in Lovsta Farm*. Master's thesis, Dalarna University.

Linderman, G.C. and Steinerberger, S. 2017. Clustering with t-SNE, provably. https://doi.org/10.48550/ARXIV.1706.02582.

Liu, E., et al. 2025. Agents are all you need: pioneering the use of agentic artificial intelligence to embrace large language models into dairy science. *Journal of Dairy Science* 108, 14038–14049. https://doi.org/10.3168/jds.2025-26775.

Liu, F.T., et al. 2008. Isolation forest, In: 2008 Eighth IEEE International Conference on Data Mining. Presented at the 2008 Eighth IEEE International Conference on Data Mining (ICDM), IEEE, Pisa, Italy, pp. 413–422. https://doi.org/10.1109/ICDM.2008.17.

Liu, T., et al. 2022. Stability estimation for unsupervised clustering: a review. *WIREs Computational Statistics* 14, e1575. https://doi.org/10.1002/wics.1575.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 129–137. https://doi.org/10.1109/TIT.1982.1056489.

Ma, J., et al. 2024. Time profiles of energy balance in dairy cows in association with metabolic status, inflammatory status, and disease. *Journal of Dairy Science* 107, 9960–9977. https://doi.org/10.3168/jds.2024-24680.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations, In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. University of California Press, pp. 281–298.

Mahdi, M.A., et al. 2021. Scalable clustering algorithms for big data: a review. *IEEE Access* 9, 80015–80027. https://doi.org/10.1109/ACCESS.2021.3084057.

McInnes, L., et al. 2017. Hdbscan: hierarchical density based clustering. *Journal of Open Source Software* 2, 205. https://doi.org/10.21105/joss.00205.

McInnes, L., et al. 2020. UMAP: uniform manifold approximation and projection for dimension reduction. https://doi.org/10.48550/arXiv.1802.03426.

McLachlan, G.J., et al. 2019. Finite mixture models. *Annual Review of Statistics and Its Application* 6, 355–378. https://doi.org/10.1146/annurev-statistics-031017-100325.

McVey, C., et al. 2020. Mind the queue: a case study in visualizing heterogeneous behavioral patterns in livestock sensor data using unsupervised machine learning techniques. *Frontiers in Veterinary Science* 7, 523. https://doi.org/10.3389/fvets.2020.00523.

Meilă, M. and Zhang, H. 2024. Manifold learning: what, how, and why. *Annual Review of Statistics and Its Application* 11, 393–417. https://doi.org/10.1146/annurev-statistics-040522-115238.

Michelena, Á., et al. 2025. Comparative analysis of unsupervised anomaly detection techniques for heat detection in dairy cattle. *Neurocomputing* 618, 129088. https://doi.org/10.1016/j.neucom.2024.129088.

Mor, B., et al. 2021. A systematic review of hidden Markov models and their applications. *Archives of Computational Methods in Engineering* 28, 1429–1448. https://doi.org/10.1007/s11831-020-09422-4.

Murtagh, F. and Contreras, P. 2012. Algorithms for hierarchical clustering: an overview. *WIREs Data Mining and Knowledge Discovery* 2, 86–97. https://doi.org/10.1002/widm.53.

Murtagh, F. and Legendre, P. 2014. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of Classification* 31, 274–295. https://doi.org/10.1007/s00357- 014-9161-z.

Murtagh, F. and Contreras, P. 2017. Algorithms for hierarchical clustering: an overview, II. *WIREs Data Mining and Knowledge Discovery* 7, e1219. https://doi.org/10.1002/widm.1219.

Narita, K., et al. 2018. Incremental clustering for hierarchical clustering, In: 2018 5th International Conference on Computational Science/ Intelligence and Applied Informatics (CSII). Presented at the 2018 5th International Conference on Computational Science/ Intelligence and Applied Informatics (CSII), pp. 102–107. https://doi.org/10.1109/CSII.2018.00025.

Oehm, A.W., et al. 2022. A machine learning approach using partitioning around medoids clustering and random forest classification to model groups of farms in regard to production parameters and bulk tank milk antibody status of two major internal parasites in dairy cows. *PLoS One* 17, e0271413. https://doi.org/10.1371/journal.pone.0271413.

Panesar, A. 2019. *Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes*. Apress L. P, Berkeley, CA.

Parsons, L., et al. 2004. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations Newsletter* 6, 90–105. https://doi.org/10.1145/1007730.1007731

Pastell, M. and Frondelius, L. 2018. A hidden Markov model to estimate the time dairy cows spend in feeder based on indoor positioning data. *Computers and Electronics in Agriculture* 152, 182–185. https://doi.org/10.1016/j.compag.2018.07.005.

Patel, A.A. 2019. *Hands-on Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data*, 1st edition. O'Reilly Media, Sebastopol, CA.

Pedregosa, F., et al. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.

Pitafi, S., et al. 2023. A taxonomy of machine learning clustering algorithms, challenges, and future realms. *Applied Sciences* 13, 3529. https://doi.org/10.3390/app13063529.

Prince, S.J.D. 2024. *Understanding Deep Learning*. The MIT Press, Cambridge and London.

Ran, X., et al. 2023. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artificial Intelligence Review* 56, 8219–8264. https://doi.org/10.1007/s10462-022-10366-3.

Rane, J., et al. 2024. *Smart Farming Using Artificial Intelligence, Machine Learning, Deep Learning, and ChatGPT: Applications, Opportunities, Challenges, and Future Directions*. Deep Science Publishing. https://doi.org/10.70593/978-81-981271-7-4.

Rebala, G., et al. 2019. Machine learning definition and basics, In: Rebala, G., et al. (Eds.), *An Introduction to Machine Learning*. Springer International Publishing, Cham, pp. 1–17. https://doi.org/10.1007/978-3-030-15729-6_1.

Rebuli, B., et al. 2024. *Exploring Milk Productivity and Energy Output Dynamics Across Lactation Periods in Automatic Milking Systems Through Multi-Algorithm Clustering*. ITA https://iris.unito.it/handle/2318/2027989.

Samariya, D. and Thakkar, A. 2023. A comprehensive survey of anomaly detection algorithms. *Annals of Data Science* 10, 829–850. https://doi.org/10.1007/s40745-021-00362-9.

SAS Institute Inc. 2013. "The FASTCLUS Procedure" in SAS/STAT® 9.4 User's Guide. Cary, NC: SAS Institute Inc.

Satoła, A. and Satoła, K. 2024. Performance comparison of machine learning models used for predicting subclinical mastitis in dairy cows: bagging, boosting, stacking, and super-learner ensembles versus single machine learning models. *Journal of Dairy Science* 107, 3959–3972. https://doi.org/10.3168/jds.2023-24243.

Schaeffer, S.E. 2007. Graph clustering. *Computer Science Review* 1, 27–64. https://doi.org/10.1016/j.cosrev.2007.05.001.

Shahinfar, S., et al. 2014. Prediction of insemination outcomes in Holstein dairy cattle using alternative machine learning algorithms. *Journal of Dairy Science* 97, 731–742. https://doi.org/10.3168/jds.2013-6693.

Shahriar, S., et al. 2015. Heat event detection in dairy cows with collar sensors: an unsupervised machine learning approach, In: 2015 IEEE SENSORS. Presented at the 2015 IEEE Sensors, IEEE, Busan, pp. 1–4. https://doi.org/10.1109/ICSENS.2015.7370528.

Shi, C., et al. 2020. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. https://doi.org/10.21203/rs.3.rs-58011/v2.

Shine, P., et al. 2018. Machine-learning algorithms for predicting on-farm direct water and electricity consumption on pasture based dairy farms. *Computers and Electronics in Agriculture* 150, 74–87. https://doi.org/10.1016/j.compag.2018.03.023.

Simankov, N., et al. 2025. Illuminating the path to enhanced resilience of machine learning models against the shadows of missing labels. *IEEE Journal of Biomedical and Health Informatics* 1–11. https://doi.org/10.1109/JBHI.2025.3558846.

Singh, H.V., et al. 2022. A literature survey based on DBSCAN algorithms, In: 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS). Presented at the 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 751–758. https://doi.org/10.1109/ICICCS53718.2022.9788440.

Sowmya, N. and Ponnusamy, V. 2021. Development of spectroscopic sensor system for an IoT application of adulteration identification on milk using machine learning. *IEEE Access* 9, 53979–53995. https://doi.org/10.1109/ACCESS.2021.3070558.

Soyeurt, H., et al. 2022. Prediction of indirect indicators of a grass-based diet by milk Fourier transform mid-infrared spectroscopy to assess the feeding typologies of dairy farms. *Animals* 12, 2663. https://doi.org/10.3390/ani12192663.

Stone, J.V. 2002. Independent component analysis: an introduction. *Trends in Cognitive Sciences* 6, 59–64. https://doi.org/10.1016/S1364-6613(00)01813-1.

Toan, T.V., et al. 2019. Cow estrus detection with low-frequency accelerometer sensor by unsupervised learning, In: Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019. Presented at the the Tenth International Symposium, ACM Press, Hanoi, Ha Long Bay, Viet Nam, pp. 342–349. https://doi.org/10.1145/3368926.3369683.

Todde, G., et al. 2016. A multivariate statistical analysis approach to characterize mechanization, structural and energy profile in Italian dairy farms. *Energy Reports* 2, 129–134. https://doi.org/10.1016/j.egyr.2016.05.006.

Tremblay, M., et al. 2018. Identifying poor metabolic adaptation during early lactation in dairy cows using cluster analysis. *Journal of Dairy Science* 101, 7311–7321. https://doi.org/10.3168/jds.2017- 13582.

Udell, M., et al. 2016. Generalized low rank models. *Foundations and Trends in Machine Learning* 9, 1–118. https://doi.org/10.1561/2200000055.

van der Maaten, L. and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 2579–2605.

Van Hoeij, R.J., et al. 2019. Relationship between metabolic status and behavior in dairy cows in week 4 of lactation. *Animal* 13, 640–648. https://doi.org/10.1017/S1751731118001842.

Vanlierde, A., et al. 2021. Improving robustness and accuracy of predicted daily methane emissions of dairy cows using milk mid-infrared spectra. *Journal of the Science of Food and Agriculture* 101, 3394–3403. https://doi.org/10.1002/jsfa.10969.

Vázquez-González, I., et al. 2024. A combined multi-variate statistical analysis to establish dairy farm typologies in Cantabria. *Computers and Electronics in Agriculture* 221, 109007. https://doi.org/10.1016/j.compag.2024.109007.

Wattenberg, M., et al. 2016. How to Use t-SNE Effectively. *Distill* 1. https://doi.org/10.23915/distill.00002.

Xu, H., et al. 2020. Analysis of Cattle Social Transitional Behaviour: Attraction and Repulsion. *Sensors* 20, 5340. https://doi.org/10.3390/s20185340.

Xu, W., et al. 2019. Prediction of metabolic status of dairy cows in early lactation with on-farm cow data and machine learning algorithms. *Journal of Dairy Science* 102, 10186–10201. https://doi.org/10.3168/jds.2018-15791.

Yujuan, C., et al. 2017. Intelligent monitoring method of cow ruminant behavior based on video analysis technology. *International Journal of Agricultural and Biological Engineering* 10(5), pp.194–202. https://dx.doi.org/10.25165/j.ijabe.20171005.3117.

Zhang, L., et al. 2021. Appropriate data quality checks improve the reliability of values predicted from milk mid-infrared spectra. *Animals* 11, 1–17. https://doi.org/10.3390/ani11020533.

Zhang, T., et al. 1996. BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Record* 25, 103–114. https://doi.org/10.1145/235968.233324.

Published by Burleigh Dodds Science Publishing Limited, 2026.