Machine Learning (2025) 114:57
https://doi.org/10.1007/s10994-024-06701-0

Conformalised data synthesis

Julia A. Meister[1] · **Khuong An Nguyen**[2]

Received: 15 December 2023 / Revised: 15 October 2024 / Accepted: 4 December 2024

Abstract
With the proliferation of increasingly complicated Deep Learning architectures, data synthesis is a highly promising technique to address the demand of data-hungry models. However, reliably assessing the quality of a 'synthesiser' model's output is an open research question with significant associated risks for high-stake domains. To address this challenge, we propose a unique synthesis algorithm that generates data from high-confidence feature space regions based on the Conformal Prediction framework. We support our proposed algorithm with a comprehensive exploration of the core parameter's influence, an in-depth discussion of practical advice, and an extensive empirical evaluation of five benchmark datasets. To show our approach's versatility on ubiquitous real-world challenges, the datasets were carefully selected for their variety of difficult characteristics: low sample count, class imbalance, and non-separability. In all trials, training sets extended with our confident synthesised data performed at least as well as the original set and frequently significantly improved Deep Learning performance by up to 61% points F_1-score.

Keywords Conformal prediction · Uncertainty quantification · Statistical confidence · Synthetic data · Data generation

Mathematics Subject Classification 68T37

Editors: Henrik Boström, Eyke Hüllermeier, Ulf Johansson, Aaditya Ramdas.

✉ Julia A. Meister
J.Meister@brighton.ac.uk

Khuong An Nguyen
Khuong.Nguyen@rhul.ac.uk

[1] Computing and Maths Division, University of Brighton, Lewes Road, Brighton, East Sussex BN2 4AT, UK

[2] Department of Computer Science, Royal Holloway, University of London, Egham Hill, Egham, Surrey TW20 0EX, UK

Published online: 06 February 2025

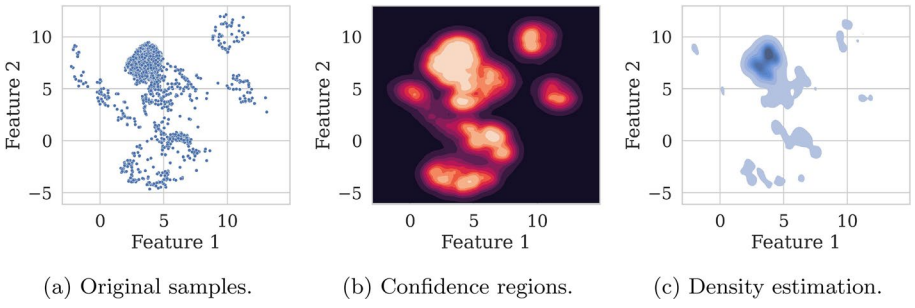
1 Introduction

Specialised and data-hungry Deep Learning implementations are increasingly faced with small and unrepresentative datasets, a well-established challenge in the literature (Brigato & Iocchi, 2021; Moreno-Barea et al., 2020; Sarker, 2021). Unfortunately, increasing the sample size by collecting more data is challenging in many real-world applications. Prohibitors include high data collection costs, low data availability, and the lack of expertise in ground-truth labelling.

Data synthesis is a highly sophisticated approach to combat small datasets. Improving on techniques that modify and remix existing samples (e.g., data resampling, randomisation, and augmentation), data synthesis generates entirely new and unseen examples based on the original data (Zhuang et al., 2019). Similarly to classification, synthesis relies on accurately modelling the data's distribution to extrapolate plausible new feature vectors (Liu et al., 2022). In Deep Learning, these synthesised samples may be included in the training set to improve model generalisation and, consequently, prediction performance.

Generative Adversarial Networks (GANs) are a state-of-the-art synthesis technique, leveraging a zero-sum game between a Deep Learning generator and discriminator to synthesise realistic samples (Aggarwal et al., 2021). However, a major challenge and open research question shared among most existing data synthesis techniques is how to quantify the produced synthetic data's quality. Evaluating a generator model's distributional fit is fundamentally difficult because there is no inherent quality metric for previously unseen data (Grnarova et al., 2019).

To address this challenge, we have designed a unique confident data synthesis algorithm based on a novel extension of the Conformal Prediction framework (Shafer & Vovk, 2008). Inspired by Cherubin et al.'s innovative conformal clustering paper (Cherubin et al., 2015), we rely on the confidence of feature space regions to guide our data generation. Figure 1 illustrates the effect of our confidence estimation approach on the feature space compared to traditional density estimation. Section 1.1 details our contributions in more depth and describes the article's structure.



(a) Original samples. (b) Confidence regions. (c) Density estimation.

Fig. 1 Visualisation of two techniques to identify feature space regions based on the original samples (**a**) from which data is synthesised. Compared to traditional density estimation of the samples (**c**), our proposed algorithm determines conformal high-confidence regions based on a user-selected threshold (**b**). The lower the threshold, the wider the confidence regions become from which samples are synthesised

1.1 Contributions

In this article, we propose conformal data synthesis to significantly improve Deep Learning prediction performance on small and imbalanced datasets. Our algorithm builds on Conformal Prediction, a foundational confidence framework founded on hypothesis testing. To the best of our knowledge, this is the first extension of Conformal Prediction to data generation.

Inspired by the innovative conformal clustering work presented in Cherubin et al. (2015), our algorithm relies on the confidence of feature space regions to synthesise new data points. Our three key contributions are:

- We incorporate ground-truth labels, making the confidence modelling process *supervised*,
- We identify *label-conditional* confidence regions in the feature space,
- And, most decisively, the confidence regions are an intermediate stage from which we *synthesise new data points*.

In the following sections, we will discuss the context and motivation of our data synthesis solution (Sect. 2); Present its theoretical foundation and inspiration (Sect. 3); Introduce our proposed synthesis algorithm (Sect. 4); Systematically evaluate the empirical advantages on real-world datasets of varying difficulty (Sect. 5); And discuss practical advice and future directions of the proposed algorithm (Sect. 6).

2 Related work in data synthesis

A model's performance is strongly dependent on its underlying data. Consequently, state-of-the-art Deep Learning models are frequently trained on vast datasets to optimise prediction performances. Although a large sample count is not necessarily required for high performance, it increases the likelihood of a representative collection of samples (Althnian et al., 2021). Unfortunately, real-world applications frequently suffer under small and unrepresentative datasets due to the shortage of high-quality annotated samples (Alauthman et al., 2023). However, artificially extended datasets have been shown to improve results. Data synthesis is the most sophisticated approach to extend datasets with entirely new data. The core concept is to generate previously unseen data points from the existing data automatically. The most popular use cases are to improve model inference with synthetic samples (Muramatsu et al., 2020; Salazar et al., 2021; Koshino et al., 2021) and to replace a dataset with synthetic samples (Li et al., 2020; Yoon et al., 2020; Thambawita et al., 2021). The former is achieved by extending the training set with new samples in keeping with the original data to support model learning.

In more detail, a successfully trained classification model h_θ will have learned a close distribution estimation of the classes $\mathbf{Y}$ in the feature space $\mathbf{X}$, $\mathrm{P}(Y|X)$ (Bashir et al., 2020). In other words, it will map latent variable relationships, and therefore certain feature space regions, to a particular class $y \in \mathbf{Y}$:

$$h_\theta(\mathbf{X}) \approx \mathbf{Y}. \tag{1}$$

The more densely a class y is represented in a region during training, the more it will be reinforced in the model's feature space representation. However, if there are too few training samples, the model is underfit and will have learned a skewed representation of the dataset (Bejani & Ghatee, 2021). Unfortunately, collecting more data to increase the sample size is difficult in many real-world situations. Prohibitors include the high data collection cost, low data availability, and lack of expertise for ground-truth labelling (Whang et al., 2023).

GANs are the most popular and ubiquitous synthesis technique to address this concern. Developed by Goodfellow et al. they encapsulate a zero-sum game between two models (Goodfellow et al., 2020). The generator attempts to fool the discriminator with new samples synthesised from random noise. Simultaneously, the discriminator attempts to distinguish between original and generated samples. Over many training iterations, the generator's output becomes more and more realistic (Kammoun et al., 2022). However, the final synthesised output is fundamentally difficult to evaluate because there is no inherent quality metric for the GAN's distributional fit (Borji, 2022; Navidan et al., 2021; Brophy et al., 2023). We must rely on purely qualitative metrics and empirical results to estimate the samples' quality, which may be problematic for high-risk domains such as healthcare, finance, and security (Saxena & Cao, 2022).

Nonetheless, multiple approaches have been developed that show improved performance with empirical confidence measures. For example, Bhattarai et al. interpreted the probabilistic output of the GAN's generator and discriminator as a confidence measure (Bhattarai et al., 2020). Based on this information, the synthetic samples were filtered for high scores, and empirical results for facial emotion recognition showed minor improvements. Similarly, Nie and Shen developed a difficulty-aware attention mechanism based on the model's confidence for medical image segmentation (Nie & Shen, 2020). They aimed to improve overall performance by reducing the training weights of easy samples based on a 'confidence' Convolutional Neural Network (CNN). The authors estimated the trustworthiness of image segments with the CNN's confidence maps and thereby took local confidence into account. In contrast, Du et al. developed a sophisticated confidence model to generate pseudo-healthy images of skin lesions (Du et al., 2022). Using a GAN, the authors generated subject-specific healthy images paired with a pathological sample to support classification performance. As part of the framework, they included a confidence check such that only segments with a confidence greater than a pre-defined threshold were considered for prediction. More recent works combine evaluation metrics to quantify different desirable characteristics that a well-fitted GAN should portray. For example, Abdusalomov et al. evaluated the synthesised samples in two stages: their distributional similarity to the original data during training and their diversity and proximity to real images of different classes after training is complete (Abdusalomov et al., 2023). While each of these examples improved prediction performance on specific datasets and tasks, there were no quality guarantees since the confidence measures were empirically derived from probabilistic neural network outputs.

There is no question that GANs can be a powerful synthesis technique (Wang et al., 2023; Shi et al., 2023; Zhao & Bilen, 2022). Unfortunately, however, they can be notoriously difficult to train. In particular, they frequently inherit the Deep Learning requirement for relatively large and balanced datasets to enable strong generalisation (Huang & Jafari, 2023). Consequently, density-based generative models are a popular alternative for small datasets requiring extension (Plesovskaya & Ivanov, 2021). For example, Kernel Density Estimation (KDE) is a state-of-the-art non-parametric approach to approximate the distribution of random variables (Park & Pardalos, 2024). Part of KDE's versatility is the

underlying kernel K, which may be defined to suit the underlying data (Bauer et al., 2024). Consequently, using n observations x_i, a random variable x's probability density estimation $p(x)$ is defined as:

$$p(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i, w). \tag{2}$$

The bandwidth w acts as a smoothing parameter, regulating the estimation's bias-variance trade-off (Belhaj, 2024). The fitted KDE model may be repeatedly sampled to synthesise new data points. For example, Pozi et al. developed a privacy-preserving data synthesis algorithm building on this technique (Pozi & Omar, 2020). After learning the probability density function of the original features with a KDE-based generative model, the distribution is carefully and deliberately shifted to obscure personally identifiable information. Through extensive classification experiments, the authors found that the shifted data's utility remained equal to the original data, maintaining downstream prediction performances.

Similar to GANs, empirical evaluations are a common occurrence for density-based generative models because they offer no inherent performance guarantees. Additionally, synthesis performance may be highly variable depending on the underlying data and user-selected parameters (e.g., the bandwidth) (Falxa et al., 2023). In contrast, theoretically supported confidence frameworks such as Conformal Prediction have an enormous potential to increase trust in synthesised data. However, to the best of our knowledge, Conformal Prediction has not previously been utilised for data synthesis. Apart from conformal clustering (discussed in Sect. 3.2), the closest related work is presented by Liu et al. The authors proposed the conformal framework for semi-supervised learning (Liu et al., 2021). In particular, Inductive Conformal Prediction was used to measure the quality of augmented samples. As a first step, the highest-confidence original samples were pre-filtered for augmentation. After augmentation, the highest-credibility adjusted samples were retained to extend the training dataset. The proposal was tested on a small dataset for herbal medicine classification, and the resulting prediction performance improved on traditional augmentation techniques. The downside was the high computational inefficiency of the approach, which was further improved in Liu et al. (2022). As the foundation of our confident synthesis proposal, Sect. 3 further discusses the Conformal Prediction confidence framework in more depth.

3 Conformal Prediction background

How confident are we that a model's prediction is correct? This is the core question that uncertainty quantification techniques such as Conformal Prediction attempt to answer. To contextualise our conformal synthesis proposal (Sect. 4), we introduce the conformal framework for classification (Sect. 3.1) and summarise the conformal clustering work that inspired our approach (Sect. 3.2).

3.1 The conformal framework

Conformal Prediction (CP) is a highly versatile confidence framework that acts as a wrapper to any underlying point prediction model (Zhang et al., 2021). Based on hypothesis testing, CP's uncertainty measures for individual predictions are statistically supported to a user-selected

significance level. This section focuses on conformal classification because of its relevance to the proposed synthesis algorithm. Interested readers are referred to Johansson et al. (2014) for conformal regression.

Under minimal exchangeability assumptions, the conformal validity property guarantees that prediction mistakes are made up to a maximum error rate (Angelopoulos et al., 2020). To achieve this strong guarantee, an underlying model's point predictions are transformed into a prediction set Γ including all plausible labels $y \in \mathbf{Y}$. Label inclusion is driven by the significance-level ϵ, making the prediction sets ϵ-dependent (Messoudi et al., 2020). An error is defined as a prediction set missing a sample's true label y^*. Due to the hypothesis testing background, the probability of a mistake being made on each sample i is capped at ϵ, subject to statistical fluctuations (Eq. 3). By the law of large numbers, the overall probability of an error occurring approaches ϵ with increasing predictions, while the individual error probability is unchanged (Zhan et al., 2020):

$$P\left(y_i^* \notin \Gamma_i^\epsilon\right) \leq \epsilon, \tag{3}$$

$$\lim_{i \to \infty} \text{avg}\, P\left(y_i^* \notin \Gamma_i^\epsilon\right) \approx \epsilon. \tag{4}$$

Originally designed for an online setting, transductive CP requires retraining the conformal model for every new test sample. Therefore, an inductive variant (ICP) was proposed to remove the need for leave-one-out retraining (Papadopoulos, 2008). However, similar to the original approach, the guarantees are valid over all predictions but not necessarily per class. For example, 'difficult' samples belonging to a minority class may average higher error rates than 'easier' samples (Ashby et al., 2022). Therefore, the Mondrian variant (MICP) builds on ICP to address this challenge by extending the guarantees to label-conditional validity (Löfström et al., 2015):

$$P(y^* \notin \Gamma^\epsilon \mid y^* = y) \approx \epsilon, \qquad \forall \quad y \in \mathbf{Y}. \tag{5}$$

Due to its unique combination of advantages, the following section describes the MICP method of prediction set construction.

The strong validity property is guaranteed through carefully constructing the prediction sets Γ^ϵ. To prepare for inductive inference, we split the original training set Z_{train} with samples $x \in \mathbf{X}$ and true labels $y^* \in \mathbf{Y}$ into the disjoint proper training set Z_{prop} and calibration set Z_{calib} :

$$Z_{train} = \{i = 1, \ldots, n | \left(x_i, y_i^*\right)\}, \tag{6}$$

$$Z_{prop} = \{i = 1, \ldots, m | \left(x_i, y_i^*\right)\}, \tag{7}$$

$$Z_{calib} = \{i = m+1, \ldots, n | \left(x_i, y_i^*\right)\}. \tag{8}$$

To make a prediction, a test sample x_{n+1}'s plausibility is evaluated through extension with each possible label $y \in \mathbf{Y}$ (Meister et al., 2023). To measure the likelihood of the postulated label given the original data, a non-conformity measure A evaluates a test extension's 'strangeness' with an underlying point prediction model U trained on Z_{prop}:

$$\alpha_{n+1}^y = A_U\left((x_{n+1}, y), Z_{prop}\right), \qquad \forall \quad y \in \mathbf{Y}. \tag{9}$$

The calibration non-conformity scores are calculated similarly to the test sample's. However, instead of extending each calibration point with every possible label, we only require values for their true labels y^*:

$$\alpha_i^{y^*} = A_U\big((x_i, y_i^*), Z_{prop}\big), \qquad \forall \quad (x_i, y_i^*) \in Z_{calib}. \tag{10}$$

An example of a straightforward non-conformity measure uses the K-Nearest Neighbour algorithm (KNN) as its underlying model:

$$A_{KNN} \quad : \quad \alpha_j^y = \sum \min_k \ \text{dist}\big(x_j, \{x_i \in Z_{prop} | y_i^* = y\}\big). \tag{11}$$

Here, α quantifies an extended sample x_j's similarity to the observed data by summing the distances to its k nearest neighbours with the same true label y^* as the postulated class y. The larger α is, the further away the test sample x is from a class y, and the less likely it is that y is a plausible label (Ndiaye, 2022).

Given a test sample's and the calibration set's non-conformity scores, we may calculate the probability of each label y being the true test label y_{n+1}^* via p values (Meister, 2020). Note that through the hypothesis testing background, all true p values p^{y^*} are automatically guaranteed to be uniformly distributed (Sesia & Romano, 2021):

$$p_{n+1}^y = \frac{\#\{i = m+1, \ldots, n | y_i^* = y, \alpha_i^{y^*} \geq \alpha_{n+1}^y\} + 1}{\#\{i = m+1, \ldots, n | y_i^* = y\} + 1}. \tag{12}$$

Finally, the prediction set Γ_{n+1}^ϵ is constructed by including all labels y whose confidence levels exceed the significance level ϵ:

$$\Gamma_{n+1}^\epsilon = \{y \in \mathbf{Y} | p_{n+1}^y > \epsilon\}. \tag{13}$$

The larger ϵ is, the narrower the prediction set becomes, and the more likely it is that a sample's true label y^* is not included. Therefore, the trade-off between low error rates and precise prediction sets must be carefully balanced. The optimal classification prediction set includes exactly one label (Vovk et al., 2016). So far, Conformal Prediction has been presented in the context of prediction tasks. Section 3.2 discusses an extension to clustering that inspired our proposed algorithm.

3.2 Conformal clustering

Conformal Prediction was originally developed to measure confidence and provide performance guarantees for prediction tasks. However, the framework has since been extended to a variety of application domains, and the one most relevant to this article is unsupervised clustering.

Introduced in Cherubin et al. (2015), conformal clustering is founded on measuring the conformal confidence of each point in a feature space. A grid of points represented the feature space with equal spacing to make the task discrete. By treating each grid point as a test sample, Cherubin et al. measured their unsupervised similarity to the observed training data with conformal p values (Sect. 3.1). All grid points with $p > \epsilon$ were considered part of the clusters, where ϵ was a user-specified threshold between 0 and 1. The clusters themselves were defined via the neighbouring rule: two grid points were part of the same cluster if they were neighbours. Therefore, in addition to setting the confidence threshold, ϵ may

be considered a regularisation factor of the clusters' connectedness. The smaller ϵ was, the more connected the high-confidence grid points were, and the fewer distinct clusters were formed.

Building on this work, Nouretdinov et al. further underpinned the understanding of conformal clustering (Nouretdinov et al., 2020). The authors extended the approach to develop multi-level conformal clustering, introducing a dendrogram construction similar to traditional hierarchical clustering methods. Additionally, a new technique for identifying out-of-distribution anomalies was established by testing whether new samples fell within a conformal cluster. More recently, Jung et al. developed a novel conformity measure for clustering that is applicable to circular variables (Jung et al., 2021). The authors demonstrated their approach by performing clustering on a dataset containing torsion measurements of different proteins.

In a further recent development, Ding et al. proposed Clustered Conformal Prediction to incorporate a clustering aspect into conformal classifiers to improve class-conditional coverage in the many-class scenario (Ding et al., 2023). The authors cluster classes with similar conformal score distributions based on the Mondrian CP variant. Calibration is then carried out within each cluster, achieving stronger 'cluster-conditional' coverage over marginal coverage. However, for the purpose of our proposed synthesis algorithm, the previously mentioned conformal clustering concept of measuring feature space confidence is particularly interesting. For a successful synthesis, we would expect the new samples to overlap with the original data in the feature space. By limiting synthesis to high-confidence regions, outliers should be minimised. With this inspiration, Sect. 4 introduces our proposed confident data synthesis algorithm.

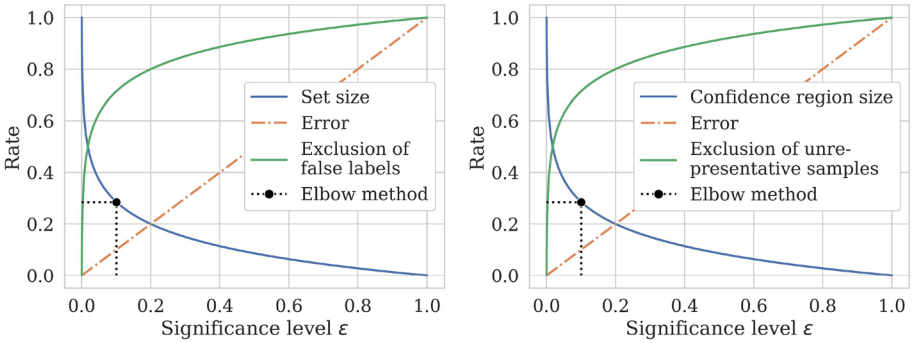
4 Conformalised data synthesis

In this section, we propose a unique data synthesis algorithm that measures the feature space confidence during the generation process. The basis of our confidence measure is the foundational Conformal Prediction framework, traditionally used for prediction and distribution testing tasks (Sect. 3). Our work presents a novel extension of the conformal technique to data generation, a previously unexplored domain. After examining our proposal's implications for the synthesised data (Sect. 4.1), we comprehensively discuss our algorithm's design and characteristics (Sect. 4.2).

4.1 Implications for data generation

Inspired by the innovative conformal clustering work summarised in Sect. 3.2, our novel conformal data synthesis algorithm relies on the confidence of feature space regions to generate synthetic datasets. Consequently, unlike traditional synthetic data generation techniques (Sect. 2), our conformal synthesis algorithm models a confidence-aware distribution of the original dataset in the feature space.

With traditional Conformal Prediction (CP), the significance level ϵ provides a statistically guaranteed error rate. Consequently, it directly regulates the trade-off between two opposing desires: low error rates and informative (i.e., narrow) prediction sets (Sect. 3.1). A common approach for optimising ϵ is via the elbow method heuristic (Jung et al., 2021), identifying the point of diminishing return on improved set sizes at the cost of more frequent prediction errors (Fig. 2a). For conformal synthesis, ϵ instead



(a) Conformal classification. (b) Conformal synthesis.

Fig. 2 Intuition of the ϵ trade-off for traditional conformal classification (**a**) and conformal synthesis (**b**). A small ϵ implies low error rates while increasing the inclusion of false labels or 'unrepresentative' samples, respectively. Note that the latter and the graphs in (**b**) must be inferred from downstream model performances due to the data generation domain. The elbow method is a straightforward heuristic to select a value for ϵ, balancing the opposing desires

acts as a threshold to identify the high-confidence feature space regions from which samples are synthesised. However, it provides only limited guarantees due to the extension to data generation, discussed further in Sect. 4.2. Nonetheless, we assume and empirically investigate a close relationship between ϵ and the overall prediction performance on synthesised datasets in Sect. 5. Therefore, we must reinterpret the (approximated) ϵ trade off in the context of data generation.

Our synthesis priority is to identify the high-confidence feature space regions, e.g., to support Deep Learning training. Consequently, an error is interpreted as the exclusion of 'representative' synthetic samples. Intuitively, the smaller ϵ is, the wider the confidence regions must become to increase the likelihood of 'representative' samples being included. However, the trade off is the inclusion of 'unrepresentative' samples, reducing the synthetic dataset's effectiveness for model training (Fig. 2b). Similarly to traditional CP, we must balance the two desires by carefully optimising ϵ. Note that because we cannot directly measure the rate of excluding 'unrepresentative' samples, we assume a close relationship between excluding 'unrepresentative' samples and a model's improved performance after training on the synthesised data. Therefore, we propose using the model's performance curve to identify ϵ via the elbow method to optimise downstream model performance, investigated in depth in Sect. 5.2.1.

The definition of the non-conformity measure (NCM) is an additional factor to consider for conformal synthesis. Feature space confidence regions are identified by comparing relative differences between non-conformity scores. As a core component to the calculations, the NCM may significantly impact the confidence regions' shapes and, consequently, how the original data's distribution is modelled for synthesis. For example, we could prioritise maintaining intra-class relationships by incorporating distances between classes or reducing noise by lowering outliers' importance. This flexibility is an inherent advantage of the CP framework and allows adjustments to be made that best suit the dataset for improved performance. With this understanding of our proposal's implications, Sect. 4.2 details the algorithm's steps and characteristics.

4.2 Proposed algorithm

Algorithm 1 illustrates the logical flow of our proposed algorithm. Given a feature space $\mathbf{X} \in \mathcal{R}^d$, we construct a grid point representation $\mathbf{G}^\gamma \in \mathcal{R}^d$ of $\mathbf{X}$ with grid step γ such that all observed samples $x \in \mathbf{X}$ fall within its boundaries. $\mathbf{X}_{grid}$ represents the collection of all grid points in $\mathbf{G}^\gamma$ (Line 1). To prepare for Mondrian Inductive Conformal Prediction (MICP), the training data $Z_{train} = \left(\left(x_1, y_1^*\right), \ldots, \left(x_n, y_n^*\right)\right)$ with $x \in \mathbf{X}$ and $y^* \in \mathbf{Y}$ is split into the proper training and calibration subsets $Z_{prop} = \left(z_1, \ldots, z_m\right)$ and $Z_{calib} = \left(z_{m+1}, \ldots, z_n\right)$ in Line 2. Then, we calculate the non-conformity scores $\alpha_{calib}^{y^*}$ of each point in the calibration set with their true label y^* (Line 3). In principle, any non-conformity measure A_U may be used. For example, Eq. (11) in Sect. 3.1 defines a neighbour-based non-conformity measure A_{KNN}. Similar to the original CP framework, the choice may have a significant impact on the size of the prediction sets (i.e., the number of synthesised samples).

Algorithm 1 Logical flow of the proposed conformal synthesis algorithm based on Mondrian inductive conformal prediction (MICP, Sect. 3.1). Each point in a discretised feature space is treated as a conformal test sample. Its class-conditional confidence dictates whether it is sampled as a synthetic point.

Require: A non-conformity measure A with underlying predictor U, a grid step γ, a significance level ϵ, and a labelled training set $Z_{train} = (x \in \mathbf{X},\ y^* \in \mathbf{Y})$.
Ensure: $\mathbf{X} \in \mathcal{R}^d,\ \mathbf{Y} \in \mathcal{N},\ 0 \leq \epsilon \leq 1$.
1: $\mathbf{X}_{grid} \Leftarrow \mathbf{G}^\gamma \in \mathbf{X}$
2: $Z_{prop}, Z_{calib} \Leftarrow \text{split}\left(Z_{train}\right)$
3: $\alpha_{calib}^{y^*} \Leftarrow A_U\left(Z_{calib}, Z_{prop}\right)$ ▷ For example, A_{KNN} in Equation (11).
4: **for** $y \in \mathbf{Y}$ **do**
5: $\quad \alpha_{grid}^{y} \Leftarrow A_U\left(\left(x \in \mathbf{X}_{grid},\ y\right), Z_{prop}\right)$
6: $\quad p_{grid}^{y} \Leftarrow \text{calculate\_p\_values}\left(\alpha_{calib}^{y^*},\ \alpha_{grid}^{y}\right)$ ▷ Following Equation (12).
7: $\quad \mathbf{R}_y^\epsilon \Leftarrow \left\{\ x_{grid} \ \middle|\ p_{grid}^{y} > \epsilon\ \right\}$ ▷ Feature space confidence regions.
8: **end for**
9: $Z_{syn} \Leftarrow \bigcup_{y \in \mathbf{Y}} \left\{\ (x, y) \ \middle|\ x \in \mathbf{R}_y^\epsilon\ \right\}$ ▷ Sample synthesis from the confidence regions.

Lines 5 to 7 contain the core of our confident data synthesis logic. To synthesise data points, we must first evaluate the confidence of each point in the feature space, represented by $\mathbf{X}_{grid}$. In Conformal Prediction terminology, we treat the grid points as test samples, extending them with each possible class label $y \in \mathbf{Y}$. Given the label-conditional non-conformity scores α_{grid}^{y}, we calculate p_{grid}^{y} following the Mondrian Inductive Conformal scheme (Eq. 12 in Sect. 3.1). The p values represent each grid point's likelihood of being assigned to class y, assuming it represents class y. In other words, these p values establish our confidence in a region's representation of a particular class. The threshold for the label-conditional confidence regions $\mathbf{R}_y^\epsilon$ is the user-selected significance-level ϵ. With this information, all grid points falling into the label-conditional high-confidence regions $\mathbf{R}_y^\epsilon$ are sampled as synthetic data points with their matching class. Finally, the set of all synthetic points Z_{syn} is constructed as the union of the label-conditional synthetic subsets (Line 9).

Derived from the validity property of CP (Eq. 14), we may draw some conclusions about the synthesised data's characteristics. In parallel with the probability of the true label $y^* \in \mathbf{Y}$ being included in the prediction set Γ^ϵ, the synthesised data $\mathbf{R}_y^\epsilon$ will include a grid point with the true label ($y = y^*$) with a marginal probability of $1 - \epsilon$:

$$\Gamma^\epsilon = \{y \in \mathbf{Y} | p^y > \epsilon\}, \mathrm{P}(y^* \notin \Gamma^\epsilon) \leq \epsilon, \tag{14}$$

$$\mathbf{R}_y^\epsilon = \{x_{grid} \in \mathbf{X}_{grid} | p_{grid}^y > \epsilon\} = \{x_{grid} \in \mathbf{X}_{grid} | y \in \Gamma^\epsilon\}, \tag{15}$$

$$\mathrm{P}(y^* \notin \Gamma^\epsilon) = \mathrm{P}\left(x_{grid}^{y^*} \notin \mathbf{R}_y^\epsilon\right), \tag{16}$$

$$\mathrm{P}\left(x_{grid}^{y^*} \notin \mathbf{R}_y^\epsilon\right) \leq \epsilon, \tag{17}$$

$$\mathrm{P}\left(x_{grid}^{y^*} \in \mathbf{R}_y^\epsilon\right) \geq 1 - \epsilon. \tag{18}$$

However, inherited from the CP framework, no theoretically-founded conclusions can be drawn about the 'false-class' predictions where $y \neq y^*$, or grid points being synthesised with a false label in conformal synthesis terms. Additionally, the focus lies on the feature space regions. Consequently, the synthesised dataset $\mathbf{R}_y^\epsilon$ does not necessarily follow the object distribution of the original data.

Regarding practical properties, the algorithmic complexity is strongly influenced by the Inductive Conformal Prediction variant (Papadopoulos et al., 2007). Assuming the underlying algorithm's training and application complexities U_t and U_a, the number of original training samples n, the number of calibration samples m, the number of grid points g, and the number of classes c, the algorithmic complexity can be described as:

$$\Theta\big(m \cdot U_t + (n - m + c \cdot g) \cdot U_a\big). \tag{19}$$

The largest individually contributing step is evaluating the per-class confidences of each point in the grid space (Lines 5 and 6 in Algorithm 1). However, due to the inductive variant, this step is fully parallelisable both within and between classes. Furthermore, as long as the original data set is unchanged, the grid's p values may be reused to generate confidence regions for any significance level ϵ. Finally, the proposed algorithm's parameters may strongly influence the synthesised output and are investigated in depth in Sect. 5.2.

In summary, our proposed conformal algorithm is a unique approach to employ feature space confidence for the data synthesis process.With this background, Sect. 5 investigates our algorithm's empirical results on real-world datasets.

5 Empirical results

In this section, we comprehensively evaluate our proposed algorithm's performance on real-world data and against a state-of-the-art density-based generative model. In particular, we investigated the following research questions:

- What influence do the conformal synthesis parameters have on the synthesised data? (Sect. 5.2)
- To what degree does conformal synthesis improve Deep Learning performance on small and large datasets? (Sect. 5.3.1)
- How effective is conformal synthesis at equalising Deep Learning performance on highly imbalanced classes? (Sect. 5.3.2)
- Can we successfully improve the quality of a dataset with overlapping classes by extending it with synthetic data points? (Sect. 5.3.3)
- How accurately can we replace a real-world dataset with entirely synthetic examples? (Sect. 5.3.4)
- How does our proposed synthesis algorithm perform compared to a state-of-the-art, density-based generative model? (Sect. 5.3.5)

5.1 Experimental setup

We conducted a rigorous empirical evaluation of the synthetic data points produced by our proposed conformal synthesis algorithm (Sect. 4.2). Because the expected use case is to bolster datasets with few samples for Machine Learning, we evaluated the effectiveness of our synthesised data by measuring the performance of a Deep Learning model trained on it.

5.1.1 Neural network architecture

We chose a feedforward neural network with three hidden layers for the model, as shown in Fig. 3. The model's architecture and parameters were chosen for their versatility and robustness to ensure a good baseline performance on all five selected datasets. The activation functions are the popular ReLu and Softmax functions for a classification output. The

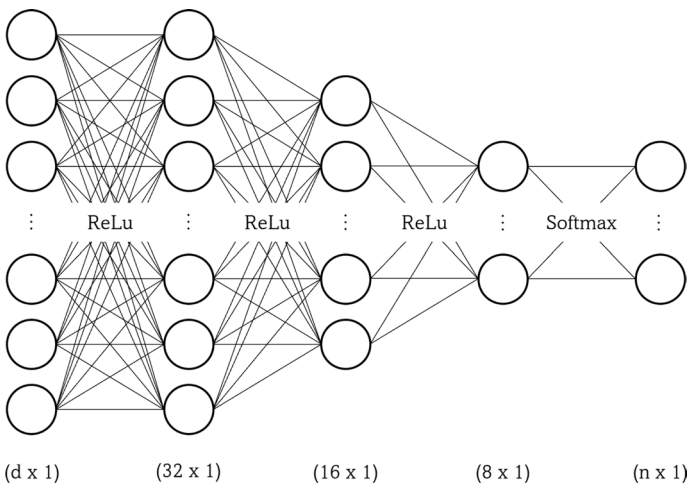


Fig. 3 The Deep Learning model architecture used in all trials. Design decisions were made to improve the versatility on different real-world datasets and the robustness of the results. The input layer size d and output layer size n are driven by the dataset's dimensionality and number of classes

number of output nodes was adjusted based on the evaluated dataset's number of classes. The model was trained for ten epochs with early stopping enabled in all trials.

5.1.2 Data subsets and synthesis parameters

Each trial was evaluated on the same test set for a meaningful comparison of the Deep Learning model's performance on the original data vs our synthesised data. Figure 4 illustrates in more detail how the data was split and the evaluated training data configurations. To generate the synthetic samples, $\text{Train}_{\text{orig}}$ was temporarily divided into a calibration (40%) and a proper training set (60%). Each subset maintained the original class proportions. Primarily, one data split was tested and reported. However, the data split may affect the synthesis and model performance results. Therefore, three additional data splits were tested in some cases to support identifying high-level trends invariant to the randomness of how samples were assigned to the data subsets.

Among a handful of other parameters, the conformal synthesis algorithm relies on defining a non-conformity measure (NCM) to identify the confidence regions of the feature space. We employed a KNN-based NCM that measures the sum of distances to the $k = 5$ nearest neighbours using the generalised Minkowski distance (Eq. 11, Sect. 3.1). KNN was chosen as the underlying algorithm due to its simplicity, robustness, and popularity in the Conformal Prediction literature (Renkema et al., 2024; Hernández-Hernández et al., 2022; Liu et al., 2021). A more common KNN-NCM additionally divides Eq. (11) with the sum of k minimum distances to instances of a different class, reducing confidence in regions with class overlap. However, our reasoning for employing the simplified NCM is to ensure that those overlapping regions are sufficiently represented in the synthesised datasets. Since feature space overlap can be common in real-world datasets, a larger volume of "difficult" synthesised samples may help Deep Learning models better generalise and distinguish them. Section 6 discusses the potential of alternative NCMs on synthesis performance. The remaining synthesis parameters' intuition and influence on the generated data are investigated in depth in Sect. 5.2.

5.1.3 Performance metrics and statistical tests

Our primary performance metrics were the total and per-class F_1-scores. Unlike alternatives like accuracy and ROC-AUC, the F_1-score is robust against dataset imbalances. The per-class F_1^c is defined as the harmonic mean of precision P^c and recall R^c, which in turn

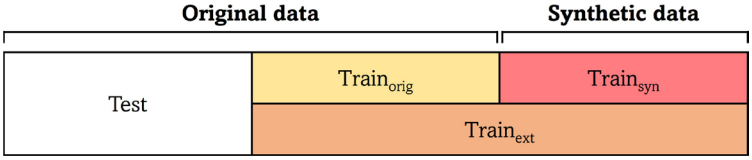


Fig. 4 The original, synthesised, and extended training sets evaluated with Deep Learning on the same held-out test set. Initially, $\text{Train}_{\text{orig}}$ is temporarily divided into the proper training (60%) and calibration sets (40%) for conformal synthesis

are calculated from the true positive TP, the false positive FP, and the false negative FN rates:

$$P^c = \frac{TP}{TP + FP}, \tag{20}$$

$$R^c = \frac{TP}{TP + FN}, \tag{21}$$

$$\mathrm{F}_1^c = 2 \cdot \frac{P^c \cdot R^c}{P^c + R^c}. \tag{22}$$

To ensure that the results were representative regardless of variance in the model's training, we repeated each experiment five times and reported the macro-average of all per-class scores:

$$P = \frac{\sum_{c \in C} P^c}{|C|}, \tag{23}$$

$$R = \frac{\sum_{c \in C} R^c}{|C|}, \tag{24}$$

$$\mathrm{F}_1 = \frac{\sum_{c \in C} \mathrm{F}_1^c}{|C|}. \tag{25}$$

The Wilcoxon signed-rank test was employed to ascertain whether the Deep learning results improved, following (Toccaceli & Gammerman, 2019; Liu et al., 2022; Johansson et al., 2017; Norinder et al., 2021; Campagner et al., 2024). The non-parametric test compares two paired groups to investigate whether they are statistically different. The null hypothesis assumes that the median of differences of matched pairs is equal to 0 (Rainio et al., 2024). It was selected after the Kolmogorov-Smirnov test revealed that the results were overwhelmingly not normally distributed. The Wilcoxon test was carried out repeatedly, comparing a pair of models each time:

- Comparing parallel model test results after training on the original $\mathrm{Train}_{\mathrm{orig}}$ and the extended $\mathrm{Train}_{\mathrm{ext}}$ sets;
- And comparing two models trained on equivalent datasets generated from different splits of the original data (e.g., $\mathrm{Train}_{\mathrm{orig}}$ from two data splits).

5.2 Illustrating the parameters' influence

We employed a simple toy dataset to demonstrate the influence of our proposed data synthesis algorithm's three parameters: the significance level ϵ, the number of original training samples n, and the grid step γ.

Visualised in Fig. 5, the 2-dimensional classification dataset was generated with 2000 samples and a 1:9 class imbalance. The two classes slightly overlap in the feature space, making the classification task more challenging but with an otherwise relatively clear class distinction. We limited the training set to 1000 samples to simulate a small dataset, and the

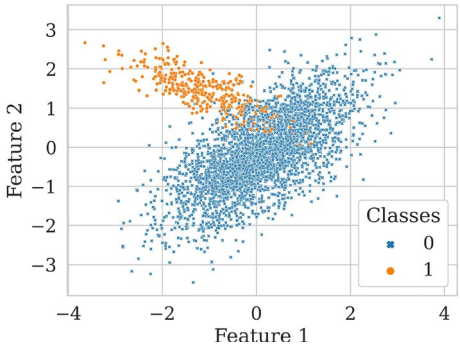


Fig. 5 A straightforward 2-dimensional toy dataset used to illustrate the influence of the proposed algorithm's parameters

Table 1 Toy dataset sample counts. The 1:9 class ratio is maintained in each data subset. 60% of the training data is temporarily allocated to the proper training set and 40% to the calibration set for synthesis

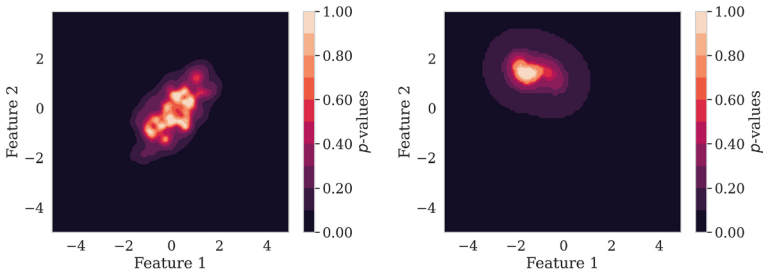
Subset	Class	Test	$\text{Train}_{\text{orig}}$		
		All	Prop.	Calib.	All (n)
$n = 1000$	Class 0	900	540	360	900
	Class 1	100	60	40	100
	All	1,000	600	400	1,000
$n = 300$	Class 0	900	162	108	270
	Class 1	100	18	12	30
	All	1,000	180	120	300
$n = 150$	Class 0	900	81	54	135
	Class 1	100	9	6	15
	All	1,000	90	60	150

remaining 1000 samples were used to evaluate the Deep Learning model's performance robustly. Each data subset maintained the original class distribution, including when the training set was artificially reduced in later sections (Table 1).

5.2.1 Significance level ϵ

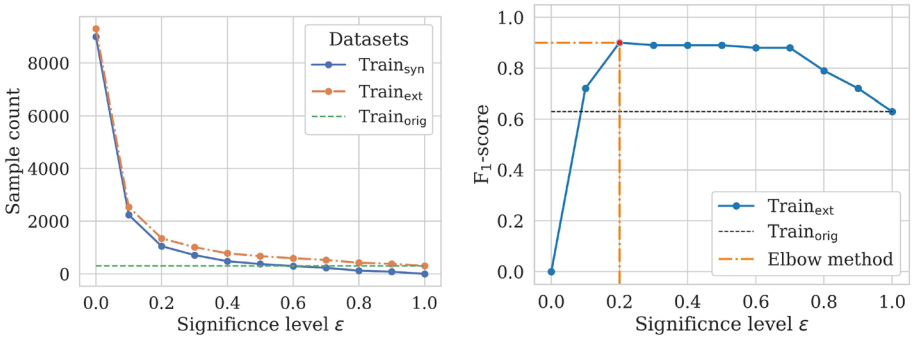
Inherited from the Conformal Prediction framework, the significance level ϵ is a central parameter of our proposed algorithm. With it, a user may set the confidence threshold for data synthesis (Sect. 4.2). Given the p values of each point in the feature space grid, we identified the per-class high-confidence regions where $p > \epsilon$. The larger ϵ was, the more carefully we controlled the high-confidence regions, tightening them around the existing data points (Fig. 6). Consequently, we see a characteristic shape in the model's F_1-score performance. The following results were generated while holding the remaining synthesis parameters steady to isolate ϵ's effect ($n = 300$, $\gamma = 0.1$). Note that because fewer minority class 1 samples were available, the confidence regions are slightly broader around the original samples than class 0.

Due to its significant impact on the synthesised data, ϵ also indirectly affected model performance, as visualised in Fig. 7. As ϵ grew, fewer synthetic samples were generated because the confidence regions became narrower. This increased the likelihood that representative samples were not included in the extended set (i.e., the synthesis error rate increases). However, the likelihood of unrepresentative samples being falsely included



(a) Class 0 confidence regions. (b) Class 1 confidence regions.

Fig. 6 Effects of the significance level ϵ on the feature space's confidence regions. The larger ϵ was, the more controlled the regions were around the original Class 0 (**a**) and Class 1 (**b**) training samples. New samples were synthesised from the high-confidence regions, defined as $p > \epsilon$



(a) Training set sizes. (b) Model performances.

Fig. 7 The effect of ϵ on the synthesised samples (**a**) and a model's F_1-score after training on the extended dataset (**b**). A characteristic performance curve was revealed, significantly increasing model performance compared to the baseline for a large range of ϵ. The elbow heuristic was employed to identify $\epsilon = 0.2$ as optimal

in the extended set was also reduced. At $\epsilon = 0$, we effectively had no confidence threshold, resulting in all possible points in the feature space being sampled to extend the training set for every class. Consequently, a model could understandably no longer distinguish the classes. However, as ϵ increased and the confidence regions became more distinguishing, we saw a sharp increase in modelling performance. Ideally, an equilibrium between the desire for improved model performance and low synthesis error rates is reached at low ϵ. An intuitive heuristic is the elbow method (Sect. 4.1), which identified $\epsilon = 0.2$ in this case. Past this point, the F_1-score after training on the $\text{Train}_{\text{ext}}$ dataset plateaued, increasing model performance by about 30% points compared to models trained on $\text{Train}_{\text{orig}}$. Finally, as the number of synthesised samples became negligible, the models' performance on both the original and extended datasets converged.

Analysing the concrete performance results in Table 2, we note that the majority of ϵ -dependent extended training sets $\text{Train}_{\text{ext}}$ improved model performances with statistical significance according to the Wilcoxon test. The best performance was indeed achieved

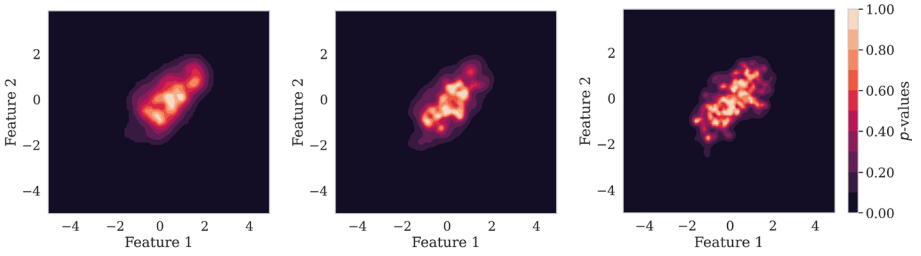
Table 2 Effects of the significance level ϵ on Deep Learning test results, the mean and standard deviation are reported. Training was carried out on the original and extended training sets. The optimal $\epsilon = 0.2$ is in bold, selected in Fig. 7b. The majority of results ($0.2 \leq \epsilon \leq 0.7$) passed the Wilcoxon test ($p_W < 0.1$) and significantly improved results, marked with *

ϵ	F_1-score			Precision			Recall		
	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W
0.1	”	0.72 (.16)	0.46	”	0.79 (.02)	0.31	”	0.91 (.00)	0.01*
0.2		**0.90** (.00)	0.02*		**0.89** (.01)	0.09*		**0.91** (.00)	0.01*
0.3		0.89 (.01)	0.03*		0.88 (.02)	0.11		0.91 (.01)	0.01*
0.4		0.89 (.02)	0.03*		0.91 (.04)	0.08*		0.88 (.03)	0.01*
0.5	0.63 (.22)	0.89 (.02)	0.03*	0.66 (.28)	0.91 (.05)	0.08*	0.62 (.17)	0.87 (.04)	0.01*
0.6	”	0.88 (.02)	0.03*	”	0.91 (.05)	0.08*	”	0.84 (.05)	0.01*
0.7		0.88 (.03)	0.03*		0.93 (.05)	0.07*		0.84 (.05)	0.02*
0.8		0.79 (.15)	0.21		0.95 (.03)	0.04*		0.75 (.15)	0.23
0.9		0.72 (.18)	0.49		0.85 (.23)	0.26		0.69 (.16)	0.54

at $\epsilon = 0.2$ with $F_1 = 90\%$ compared to the baseline of 63%, as indicated by the previously visualised F_1 curve. Increasing the training set with data synthesis improved the model's ability to distinguish between classes reliably. Note that the conformal error guarantees do not automatically apply to the model's performance.

5.2.2 Original training sample count *n*

As with Deep Learning training, we assume that a larger training dataset will improve synthesis performance. Therefore, this section investigates the effect of the original training sample count n on the feature space confidence regions. Figure 8 illustrates this relationship for class 0 on three subsets of $\text{Train}_{\text{orig}}$ with $n \in \{150, 300, 1000\}$. The confidence regions of the feature space became visibly sharper with increasing n, narrowing around the original samples (Fig. 5). Assuming that narrower prediction sets reduce the inclusion of unrepresentative samples, we expect the performance of a model trained on the synthetically extended training sets to improve with increasing n.



(a) $n = 150$. (b) $n = 300$. (c) $n = 1000$.

Fig. 8 Effects of the original training sample count n on the feature space's confidence regions. With the increase of n (**a**–**c**), our proposed algorithm could more precisely identify high-confidence regions ($p > \epsilon$). The narrower regions ensured that fewer unrepresentative samples were synthesised

Table 3 Mean and standard deviation prediction results with varying numbers of original training samples n. Larger values of n resulted in fewer synthetic samples. The increased model performance indicates that the number of unrepresentative synthesised samples was especially reduced. The Wilcoxon test confirmed that all $\text{Train}_{\text{ext}}$ and $\text{Train}_{\text{syn}}$ results statistically improved on the original data ($p_W < 0.1$)

Samples		$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$			$\text{Train}_{\text{syn}}$		
n	Syn.	F_1-score	F_1-score	Precision	Recall	F_1-score	Precision	Recall
150	2041	0.58 (.12)	0.80 (.02)	0.75 (.02)	0.90 (.00)	0.77 (.02)	0.73 (.02)	0.90 (.01)
300	1350	0.63 (.22)	0.90 (.00)	0.89 (.01)	0.91 (.00)	0.90 (.00)	0.89 (.01)	0.91 (.00)
1000	1110	0.79 (.19)	0.91 (.00)	0.92 (.00)	0.90 (.01)	0.90 (.01)	0.90 (.01)	0.91 (.00)

Table 3 presents the performance of models trained on the original and synthesised data with varying n ($\epsilon = 0.2, \gamma = 0.1$). As expected with the insights from Fig. 8, the number of synthesised samples decreased as n grew because the confidence regions became narrower. Turning our attention to the $\text{Train}_{\text{ext}}$ dataset's F_1-scores, each synthesis extension significantly improved results between 12–27% points. In particular, the potential for improvement through synthesis was the highest for the smallest datasets $n \in \{150, 300\}$ with as few as 15 minority-class samples in $\text{Train}_{\text{orig}}$ (Table 1). Examining $\text{Train}_{\text{syn}}$ revealed the same trend, indicating the improvements were likely not due to more real data in the training set. The Wilcoxon test p values p_W in Table 4 confirmed that the performance improvements related to the original training sample count n were statistically significant. Note that this article synthesises all high-confidence grid points as synthetic samples. In future, further synthesis sampling techniques could additionally investigate the precise effects of varying synthesised sample counts on the performance Sect. 6).

5.2.3 Grid step γ

The grid step γ defines the resolution of our feature space, bounded by the minima and maxima of the original dataset's features. Each grid point may be considered a potential synthetic sample with label-conditional p values tested against the significance level ϵ. Therefore, the smaller γ is, the more synthetic data points will be sampled from our high-confidence feature space regions, as shown in Fig. 9. A higher density of synthetic samples is desirable because Deep Learning models tend to generalise better with larger training sets. However, the trade-off is a higher required computing power, as each grid point must be evaluated per class. In areas where the classes' high-confidence regions overlap, both classes are assigned the same synthetic samples. Such behaviour would most likely occur for datasets with overlapping classes where the Bayes error, i.e., the probability of a perfect model making a prediction error (Salazar et al., 2023), is non-zero.

Table 4 The Wilcoxon results p_W comparing the F_1-scores of multiple test iterations on the $\text{Train}_{\text{ext}}$ and $\text{Train}_{\text{syn}}$ sets. In all cases, the n-driven improvements reported in Table 3 were found to be statistically significant ($p_W < 0.1$), marked with *

n	$\text{Train}_{\text{ext}}$			$\text{Train}_{\text{syn}}$		
	150	300	1,000	150	300	1,000
150	1.00	0.00*	0.02*	1.00	0.00*	0.00*
300	0.00*	1.00	0.00*	0.00*	1.00	0.09*
1000	0.02*	0.00*	1.00	0.00*	0.09*	1.00

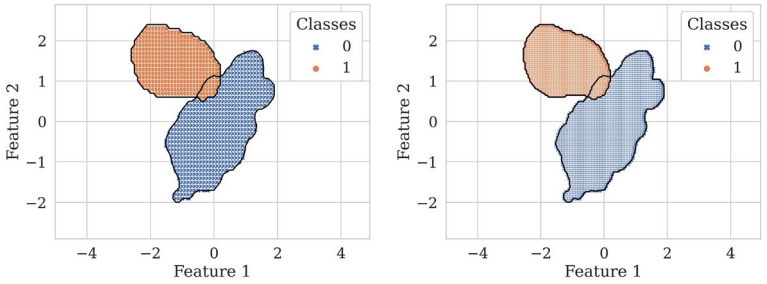


Fig. 9 Effects of the grid step γ on the feature space's resolution (**a**, **b**). The points in the graphs represent the grid points that fell into the outlined high-confidence regions ($\epsilon = 0.1$). A smaller γ means a higher density of synthetic samples may be generated from the same high-confidence regions

Table 5 shows these effects on our toy dataset. All extended training sets significantly improved on the original performance, increasing the F_1-score by up to 29% points. Additionally to the improved generalisation, the reduction of the F_1-score's standard deviation by a factor of 10 indicates that the models were trained more robustly. Varying $\gamma \in \{0.1, 0.05, 0.01\}$ while holding the other parameters steady ($\epsilon = 0.1$, $n = 150$), we found that the number of synthetic samples generated from the exact same high-confidence regions significantly increased with a decrease in the grid step. Consequently, the same Deep Learning model's generalisation performance increased by up to 9% points: from $F_1 = 78\%$ when trained on the dataset extended with $\gamma = 0.1$, to $F_1 = 87\%$ when trained on the dataset extended with $\gamma = 0.01$. In particular, recall scores were significantly improved due to the stronger representation of the minority class with an increased sample count. Investigating the Wilcoxon p values p_W confirmed that the γ-induced improvements were statistically significant for all combinations.

5.2.4 Bringing everything together

After investigating the three parameters of our conformal synthesis algorithm individually, this section illustrates the interactions between the significance ϵ, the number of original training samples n, and the grid step γ. All three influence the core of our algorithm by widening or narrowing the confidence regions in their own way. Consequently, the parameter values and the underlying original dataset define the extended training set $\text{Train}_{\text{ext}}$.

Table 5 Mean and standard deviation prediction results with varying grid step γ. The smaller γ was, the more synthetic samples were generated. Consequently, the models' generalisation was improved. The Wilcoxon test confirmed that the γ-related improvements were significant in all cases ($p_W < 0.1$), marked with *

γ	Samples	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$			p_W (F_1-score)			
	Ext.	F_1-score	F_1-score	Precision	Recall	γ	0.1	0.05	0.01
0.1	2,580	0.58 (.12)	0.78 (.03)	0.74 (.02)	0.89 (.01)	0.1	1.00	0.00*	0.00*
0.05	10,330		0.84 (.01)	0.80 (.01)	0.89 (.00)	0.05	0.00*	1.00	0.00*
0.01	258,397		0.87 (.01)	0.87 (.02)	0.87 (.01)	0.01	0.00*	0.00*	1.00

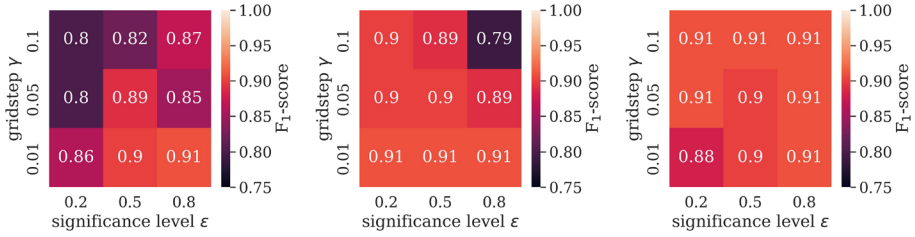
Figure 10 shows a heat map of the mean F_1-scores achieved by a Deep Learning model trained on the extended dataset $\text{Train}_{\text{ext}}$ five times. Interestingly, we observed a pattern with the changing parameters. The results tended to increase with the following:

- A larger significance level ϵ, which tightened the high-confidence feature space regions;
- A smaller grid step γ, leading to a higher resolution of the feature space;
- And a larger number of original training points n, allowing for more precise modelling of the feature space confidence.

Note that these trends are subject to randomness during the models' training and are not guaranteed results. The conformal performance characteristics discussed in Sect. 4 apply only to the synthesis process, not the downstream model prediction performances. However, Fig. 10 indicates that the relationship between model performance and ϵ are closely related: prediction performances tended to improve in parallel, subject to randomness in the model's generalisation. A notable exception is the F_1-score on data synthesised with $n = 300$, $\gamma = 0.1$, and $\epsilon = 0.8$. The significantly reduced model performance is likely related to an insufficient number of synthetic samples being generated as a consequence of narrow confidence regions (caused by high ϵ) and a low resolution of the feature space (caused by low γ).

Evaluating the performance results compared to the $\text{Train}_{\text{orig}}$ baseline results in Table 6 revealed that the majority of extended datasets strongly increased model performance by 9–33% points F_1-score across all investigated synthesis parameter settings. The standard deviation was also significantly decreased, indicating the synthetic samples enabled the Deep Learning models to generalise the dataset more robustly. This insight was further supported by the Wilcoxon test, which revealed that the improvements compared to the baseline were statistically significant. Only three models narrowly did not pass the test ($p_W < 0.1$), all occurring at $n = 1000$. These results can be traced back to the insight that the potential for dataset improvement through synthesis is highest for small datasets (Sect. 5.2.2).

Because the synthesis algorithm does not guarantee the models' prediction performance, the way the original dataset is split may impact generalisation. Therefore, additional random splits of the original samples into the test and $\text{Train}_{\text{orig}}$ datasets were evaluated, which underpin all following subsets (e.g., the proper training and calibration



(a) $n = 150$. (b) $n = 300$. (c) $n = 1000$.

Fig. 10 Overview of Deep Learning performance on the extended training sets $\text{Train}_{\text{ext}}$, synthesised with varying parameters (**a**–**c**). Each square represents the model's mean F_1-score on the same original test set. Results tended to improve with larger significance levels ϵ, smaller grid steps γ, and larger original training sample counts n

Table 6 Mean and standard deviation F_1-score results with varying conformal synthesis parameters. The best results per category are highlighted in bold. The Wilcoxon test confirmed that the improvements were largely statistically significant ($p_W < 0.1$), marked with *. The improvement potential of conformal synthesis is largest for small datasets, explaining the small number of exceptions with $n = 1000$

n	γ	$\text{Train}_{\text{orig}}$	$\epsilon = 0.2$		$\epsilon = 0.5$		$\epsilon = 0.8$	
			$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{ext}}$	p_W
150	0.1	0.58 (.12)	0.80 (.02)	0.00*	0.82 (.04)	0.00*	0.87 (.06)	0.00*
	0.05		0.80 (.03)	0.00*	0.89 (.01)	0.00*	0.85 (.06)	0.00*
	0.01		**0.86** (.00)	0.00*	**0.90** (.00)	0.00*	**0.91** (.00)	0.00*
300	0.1	0.63 (.22)	0.90 (.00)	0.02*	0.89 (.02)	0.03*	0.79 (.15)	0.02*
	0.05		0.90 (.00)	0.02*	0.90 (.01)	0.02*	0.89 (.02)	0.02*
	0.01		**0.91** (.01)	0.02*	**0.91** (.00)	0.02*	**0.91** (.00)	0.02*
1000	0.1	0.79 (.18)	**0.91** (.00)	0.08*	**0.91** (.00)	0.09*	0.91 (.01)	0.09*
	0.05		**0.91** (.00)	0.11	0.90 (.01)	0.13	**0.91** (.00)	0.07*
	0.01		0.88 (.03)	0.09*	0.90 (.00)	0.11	**0.91** (.00)	0.08*

set for synthesis). Table 7 presents the Wilcoxon test p values comparing pairs of model performances quantified by the F_1-scores. Three new data splits were compared against the results achieved on the original data split in turn. For the majority of additional data splits, we achieved the desired result of failing to reject the H_0 hypothesis, indicating the Deep Learning results were statistically similar. Therefore, we may assume that the data split had relatively little impact on the synthesis process and the subsequent model training. Now that we have systematically and comprehensively evaluated a simple toy dataset, we turn to real-world datasets to verify our proposed conformal synthesis algorithm. The insights gained in this section informed the following algorithm parameter choices.

5.3 Evaluating real-world datasets

To assess the practical benefits of our conformal synthesis algorithm, we tested its application to four realistic benchmark datasets. The datasets were carefully selected to showcase

Table 7 Wilcoxon test results evaluating the effect of the data split on the models' F_1-scores. Three random data splits of the original samples were evaluated against the previous results (Table 6) in turn, with the range of p values reported. The desirable outcome to consistently fail the test $p_W < 0.1$ was reached in the majority of cases, marked with *. Therefore, we conclude that conformal synthesis and subsequent model performances were largely invariant to the data split

n	γ	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$		
			$\epsilon = 0.2$	$\epsilon = 0.5$	$\epsilon = 0.8$
150	0.1	0.61−0.88*	0.08−0.19	0.16−0.30*	0.16−0.82*
	0.05		0.19−0.45*	0.14−0.20*	0.25−0.34*
	0.01		0.12−0.36*	0.35−0.61*	0.54−1.00*
300	0.1	0.93−0.97*	0.14*	0.36−0.55*	0.14−0.38*
	0.05		1.00*	0.09−1.00	0.17−0.19*
	0.01		0.09−0.27	0.24−0.54*	0.24−0.35*
1000	0.1	0.89–−0.96*	0.10−0.27	0.14−1.00*	0.11−0.35*
	0.05		0.29−1.00*	0.14−1.00*	0.20*
	0.01		0.24−0.57*	0.13−0.24*	0.09−0.24

our algorithm's range on some of the most prevalent data challenges limiting Machine Learning implementations: Low sample count, class imbalance and overlap, and data privacy.

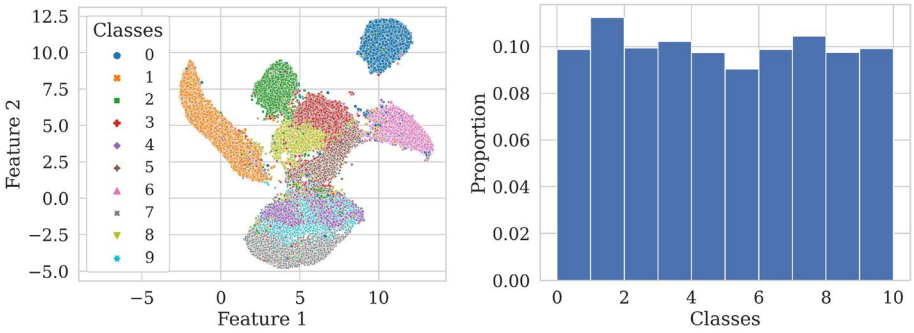
Following the details laid out in Sect. 5.1, we measured the success of our algorithm by comparing the prediction results of a feedforward neural network trained on the original data $\text{Train}_{\text{orig}}$ and our extended data $\text{Train}_{\text{ext}}$ on the same test set. The optimal parameters for data synthesis were identified with the insights gathered in Sect. 5.2. Note that we used UMAP (McInnes et al., 2018) to reduce all datasets' dimensions to two features to improve the computational complexity of synthesis, discussed in depth in Sect. 6.

5.3.1 Small dataset

Low sample counts are a ubiquitous challenge for Deep Learning, often caused by the difficulties and costs of data collection. We used the popular MNIST dataset (LeCun et al., 1998) to simulate a low training sample count. With 70,000 original samples, we could conduct an in-depth investigation of our algorithm's benefits across a range of dataset sizes. 10,000 samples were held back to evaluate the model performances on the test set. Figure 11 visualises the dataset, including a representative selection of handwritten digit samples (0–9). Each 28×28 pixel image was scaled to (0, 1) and reduced to two dimensions before synthesis. Figure 11b shows the samples' distribution in the feature space after pre-processing. While some classes were clearly linearly separated, most were adjacent with limited overlap. Overall, the classes were roughly equally distributed and this distribution was maintained for each data subset (Fig. 11c).

The core purpose of our proposed algorithm is to support Deep Learning generalisation by extending datasets with synthetic training samples. To showcase our approach's benefits across varying set sizes, we artificially under-sampled $\text{Train}_{\text{orig}}$ creating four subsets D_n with $n \in \{500, 1000, 10,000, 60,000\}$ training samples. Figure 12 reveals that all four

(a) Handwritten digit samples.



(b) Pre-processed samples.

(c) Class distribution.

Fig. 11 The MNIST dataset of handwritten digits. Each image (**a**) was reduced to two dimensions with UMAP before model training (**b**). Most classes were adjacent with some overlap, making class separation challenging. The relative class distributions were maintained in all data subsets (**c**)

subsets followed the expected model performance pattern for different ϵ when trained on the corresponding extended $\text{Train}_{\text{ext}}$ sets ($\gamma = 0.01$): A sharp increase, followed by a plateau and then converging with the baseline performance on $\text{Train}_{\text{orig}}$. The larger the original training set was, the more quickly the performance increased. We employ the elbow method to select the optimal ϵ, identifying $\epsilon = 0.2$ for D_{500} and D_{1000}, and $\epsilon = 0.1$ for $D_{10,000}$ and $D_{60,000}$.

Generally, the more original training points were available, the smaller the number of synthesised samples was, caused by more precise (and therefore more narrow) confidence regions. To illustrate the relationship between ϵ, the original training set size, and the number of synthesised samples, Table 8 reports the exact sample counts. In particular, we investigated the optimal ϵ for each training subset as well as $\epsilon = 0.9$, since models performed roughly as well on these $\text{Train}_{\text{ext}}$ sets as for their optimal ϵ (Fig. 12). Notably, the number of synthesised samples decreased as the number of original samples n and the significance level ϵ increased. Intuitively, increasing both parameters caused the confidence regions to narrow and surround the original training samples more precisely, reducing the feature space area from which new samples are synthesised. ϵ was the single most influential factor, reducing synthesised sample counts by a factor of up to around 20, going from $\epsilon = 0.1$ to $\epsilon = 0.9$.

Investigating this aspect further in Table 9, we note that in many cases, training sets extended with samples synthesised with optimal low ϵ and $\epsilon = 0.9$ confidence thresholds performed very similarly on F_1-score, precision, and recall. While we prioritised low ϵ to reduce the synthesis error in this article, a different heuristic may be to select the highest possible ϵ with the elbow method to reduce the number of required synthesis samples, constructing the most efficient extended training set to maximise model performance (discussed further in Sect. 6). Comparing the performance against the baseline, models trained on $\text{Train}_{\text{orig}}$ performed very poorly on low training sample counts as expected (under 30% F_1-score for D_{500} and D_{1000}). The score increased up to 82% with the maximum number of original training samples ($D_{60,000}$). In contrast, our synthetic dataset extension surpassed that performance from 10,000 original training samples. With only 500 original samples, our algorithm synthesised samples that improved model performance from 21 to 79% F_1-score. While the rate of improvement decreased with increasing training samples, our extended $\text{Train}_{\text{ext}}$ consistently outperformed $\text{Train}_{\text{orig}}$, reaching a maximum of 83% on the full dataset. Because the classes were roughly equally balanced in the original dataset, this increase was caused by precision and recall improving in equal measures across the classes.

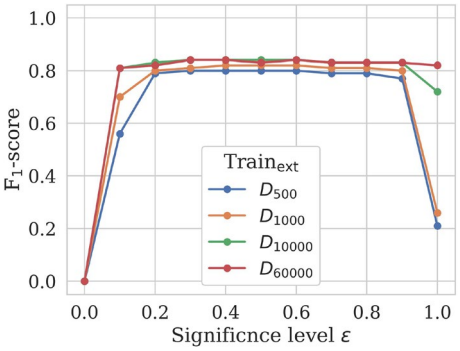


Fig. 12 Mean model performances on the extended datasets synthesised from $\text{Train}_{\text{orig}}$ subsets with 500, 1000, 10,000, and 600,000 samples. The larger the original training dataset was, the faster the performance increased, allowing us to identify lower ϵ values with the elbow method ($\epsilon = 0.2$ for the two smaller subsets and $\epsilon = 0.1$ for the two larger datasets)

Table 8 Original and synthesised sample counts. The larger the original dataset was, the narrower and more precise the confidence regions were, and the fewer new sample points were synthesised. Note that the number of synthesised samples also depended on the significance level. Therefore, the comparison is most clear when comparing variations of n for the same ϵ

Subset	Test	$\text{Train}_{\text{orig}}$			ϵ	$\text{Train}_{\text{syn}}$	$\text{Train}_{\text{ext}}$
	All	Prop.	Calib.	All (n)		All	All
$\mathbf{D_{500}}$	10,000	300	200	500	0.2	1,213,241	1,213,741
					0.9	126,190	126,690
$\mathbf{D_{1000}}$	10,000	600	400	1000	0.2	1,028,384	1,029,384
					0.9	75,067	76,067
$\mathbf{D_{10,000}}$	10,000	6000	4000	10,000	0.1	1,287,020	1,297,020
					0.9	49,908	59,908
$\mathbf{D_{60,000}}$	10,000	36,000	24,000	60,000	0.1	1,128,440	1,188,440
					0.9	44,242	104,242

Table 9 Deep Learning results on the MNIST dataset, investigating the impact of reducing the size of samples in $\text{Train}_{\text{orig}}$ before synthesis and model training. The best results per category are highlighted in bold. The mean and standard deviation of five trials are reported. As the number of original samples increased, so did the models' performance, although at different rates. The precision and recall scores stayed level with each other, implying that synthesis successfully maintained the class' original balance. The Wilcoxon test confirmed that the improvements were statistically significant ($p_W < 0.1$)

		F_1-score			Precision			Recall		
	ϵ	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W
$\mathbf{D_{500}}$	0.2	0.21 (.07)	**0.79** (.07)	0.00*	0.21 (.06)	**0.80** (.00)	0.00*	0.29 (.09)	**0.79** (.01)	0.00*
	0.9		0.77 (.01)	0.00*		0.78 (.01)	0.00*		0.77 (.01)	0.00*
$\mathbf{D_{1000}}$	0.2	0.26 (.04)	**0.80** (.01)	0.00*	0.27 (.05)	**0.82** (.01)	0.00*	0.36 (.05)	**0.80** (.01)	0.00*
	0.9		**0.80** (.01)	0.00*		0.80 (.01)	0.00*		**0.80** (.01)	0.00*
$\mathbf{D_{10,000}}$	0.1	0.72 (.04)	0.81 (.01)	0.00*	0.76 (.02)	0.82 (.01)	0.00*	0.74 (.03)	0.81 (.01)	0.00*
	0.9		**0.83** (.00)	0.00*		**0.84** (.00)	0.00*		**0.83** (.00)	0.00*
$\mathbf{D_{60,000}}$	0.1	0.82 (.01)	0.81 (.01)	0.00*	0.84 (.02)	0.82 (.01)	0.00*	0.82 (.02)	0.81 (.00)	0.00*
	0.9		**0.83** (.01)	0.09*		**0.84** (.00)	0.11		**0.83** (.01)	0.08*

These results show that our synthesis algorithm can successfully identify high-confidence class regions of the feature space with very few original training samples. This makes our proposal highly advantageous for Deep Learning, as its synthesised samples allow for significantly improved model generalisation without requiring additional data collection.

5.3.2 Imbalanced data

Class imbalances are a frequent and often inevitable occurrence in real-world datasets due to the underlying population's unequal distribution (e.g., healthy vs infected). Even if the total number of available samples is large, the minority class's under-representation may lead to skewed results. To illustrate the effects of imbalance on model performance and the combating benefits of our data synthesis algorithm, we use the MSHRM benchmark dataset (Lincoff, 1983). MSHRM contains records of around 8100 samples separated into edible (class 0) and poisonous (class 1) mushrooms. The pre-processing steps include replacing categorical variables with dummies and reducing the samples to two dimensions with UMAP (Fig. 13).

Once again, we simulated different levels of class imbalance by sub-sampling the original dataset to create four subsets D with 1:1, 1:2, 1:4, and 1:9 class ratios. Table 10 contains the tested subsets and their per-class sample counts. Following the procedure in Sect. 5.3.1, $\epsilon = 0.1$ was selected via the elbow method. Worth noting is that the level of imbalance had a drastic impact on the number and class distribution of our synthetic samples, far out-reaching the class ratio of $\text{Train}_{\text{orig}}$ (e.g., $D_{1:9}$ with 1:9 original vs 1:55 synthetic sample class ratios).

The Deep Learning results are shown in Table 11. As expected, model performance on $\text{Train}_{\text{orig}}$ decreased severely with the increasing imbalance (76%–34%). The primary contributor was the F_1-score on the minority class 1, which dropped from 75% to 0%. In contrast, the model's overall F_1-score when trained on the extended training sets remained steady at around 96%, increasing by 20–61% points with increasing class imbalance. Interestingly, although the synthetic samples reintroduced the class imbalance (albeit with class 1 now as the majority class), the imbalance was not represented in the final results. Visualising this tendency in Fig. 14, we found that over-supporting the minority class with synthetic samples did not have the same negative performance impacts as the original imbalance had.

5.3.3 Overlapping classes

Deep Learning modelling of prediction tasks relies on the separability of the classes. However, real-world datasets often do not have clear separation due to sample noise and uninformative features. We demonstrate the advantages of our proposed algorithm in these cases on the WINE benchmark dataset (Cortez et al., 2009), where we classify whether a

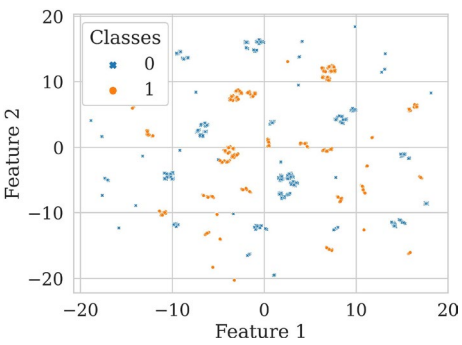


Fig. 13 The MSHRM dataset. Originally roughly equally distributed, we artificially sub-sampled the data to simulate class imbalances of varying severity for synthesis

Table 10 MSHRM sample counts with varying imbalance, differentiated by class. The original data's imbalance significantly impacted the balance of the synthetic samples ($\epsilon = 0.1, \gamma = 0.01$). The larger the imbalance was, the more samples were synthesised for the minority class 1

Subset	Class	Test	Train_{orig}			Train_{syn}	Train_{ext}
		All	Prop.	Calib.	All	All	All
$\mathbf{D_{1:1}}$	0	1394	1698	1132	2830	26,585	29,415
	1	1287	1568	1045	2613	16,926	19,539
	All	2681	3266	2177	5443	43,511	48,954
$\mathbf{D_{1:2}}$	0	1394	1696	1130	2826	29,251	32,077
	1	1287	836	557	1393	65,423	66,816
	All	2681	2532	1687	4219	94,674	98,893
$\mathbf{D_{1:4}}$	0	1394	1699	1132	2831	27,162	29,993
	1	1287	425	283	708	197,530	198,238
	All	2681	2124	1415	3539	224,692	228,231
$\mathbf{D_{1:9}}$	0	1394	1697	1131	2828	30,383	33,211
	1	1287	189	126	315	1,658,057	1,658,372
	All	2681	1886	1257	3143	1,688,440	1,691,583

Table 11 MSHRM Deep Learning results with varying class imbalance. We report the mean and standard deviation across five iterations. The best results per category are highlighted in bold. Performance on the original dataset decreased significantly with increasing imbalance. Conversely, performance on the synthetically extended training set was significantly improved compared to the baseline ($p_W < 0.1$) and remained stable across classes and as the imbalance shifted

Data	Class	F_1-score			Precision			Recall		
		$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{ext}}$	p_W
$\mathbf{D_{1:1}}$	0	0.78 (.04)	**0.96** (.01)	0.00*	0.77 (.07)	**0.97** (.01)	0.00*	0.78 (.06)	**0.95** (.02)	0.00*
	1	0.75 (.07)	**0.96** (.01)	0.00*	0.76 (.04)	**0.95** (.02)	0.00*	0.75 (.11)	**0.97** (.01)	0.00*
	All	0.76 (.05)	**0.96** (.01)	0.00*	0.77 (.05)	**0.96** (.01)	0.00*	0.76 (.05)	**0.96** (.01)	0.00*
$\mathbf{D_{1:2}}$	0	0.74 (.01)	**0.96** (.03)	0.00*	0.59 (.01)	**0.98** (.02)	0.00*	**0.98** (.02)	0.93 (.04)	0.00*
	1	0.43 (.05)	**0.95** (.03)	0.00*	0.93 (.05)	**0.93** (.04)	0.08*	0.28 (.05)	**0.98** (.02)	0.00*
	All	0.58 (.03)	**0.96** (.02)	0.00*	0.76 (.03)	**0.96** (.02)	0.00*	0.63 (.01)	**0.96** (.02)	0.00*
$\mathbf{D_{1:4}}$	0	0.70 (.02)	**0.97** (.00)	0.00*	0.54 (.03)	**0.97** (.00)	0.00*	**1.00** (.00)	0.97 (.01)	0.00*
	1	0.13 (.17)	**0.97** (.00)	0.00*	0.39 (.54)	**0.97** (.01)	0.04*	0.08 (.11)	**0.96** (.00)	0.00*
	All	0.41 (.10)	**0.97** (.00)	0.00*	0.47 (.28)	**0.97** (.00)	0.00*	0.54 (.05)	**0.97** (.00)	0.00*
$\mathbf{D_{1:9}}$	0	0.68 (.00)	**0.95** (.01)	0.00*	0.52 (.00)	**0.96** (.00)	0.00*	**1.00** (.00)	0.94 (.02)	0.00*
	1	0.00 (.00)	**0.95** (.01)	0.00*	0.00 (.00)	**0.94** (.02)	0.00*	0.00 (.00)	**0.96** (.00)	0.00*
	All	0.34 (.00)	**0.95** (.01)	0.00*	0.26 (.00)	**0.95** (.01)	0.00*	0.50 (.00)	**0.95** (.01)	0.00*

sample represents white (class 0) or red (class 1) wine based on chemical measurements. As shown in Fig. 15, this "difficult" dataset had no class separability, with both classes overlapping quite significantly in the feature space. Consequently, the high-confidence regions for data synthesis overlapped as well. Table 12 records the evaluated original, synthetic, and extended sample counts ($\epsilon = 0.2, \gamma = 0.1$).

Figure 16 visualises the mean and standard deviation performance of five models trained on the original $\text{Train}_{\text{orig}}$ and the extended $\text{Train}_{\text{ext}}$ datasets. Compared to the 43% F_1 -score achieved on the original data, including the synthesised data for training increased

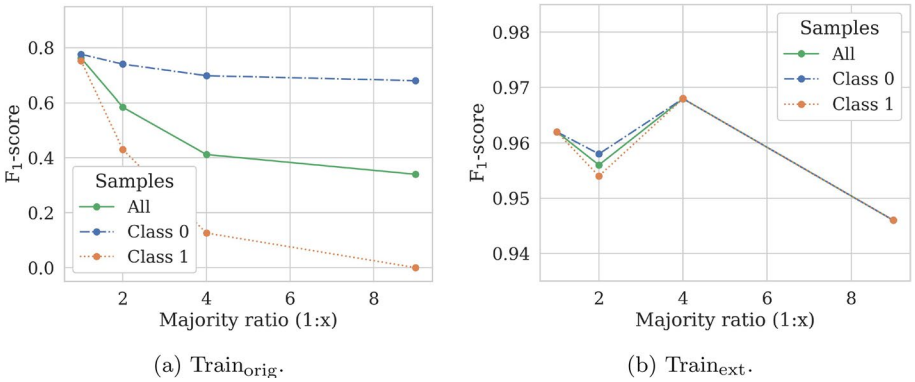


Fig. 14 Mean Deep Learning performance on the MSHRM dataset with varying class imbalance. Performance on the original dataset decreased significantly with increasing imbalance, widening class discrepancies (**a**). Conversely, performance improved on the extended training set (**b**) because the number of synthetic samples was increased exponentially (Table 10)

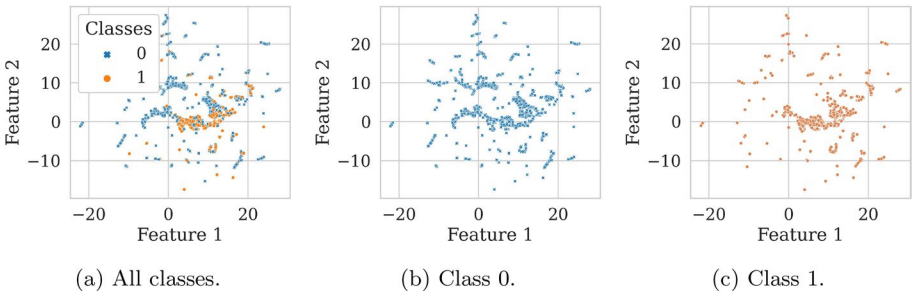


Fig. 15 The WINE dataset showed significant class overlap in the feature space (**b**, **c**). Consequently, the high-confidence regions identified by our synthesis algorithm were also overlapped. The non-separability of the classes (**a**) made accurate predictions challenging

the performance by 26% points to a total of 69%. Precision and recall results confirmed that the model's ability to identify relevant samples was indeed strengthened. In summary, even though our proposed algorithm did not increase the linear separability of the classes, it nonetheless significantly improved the model's performance. The model's learning was supported through the synthesis of a large number of new training samples.

5.3.4 Synthetic replacement

As a final show of our algorithm's ability to accurately synthesise new samples, we tested a complete synthetic replacement of the original dataset for Deep Learning training. The experiments were carried out on the USPS dataset of digits scanned from envelopes by the U.S Postal Service (Hull, 1994), shown in Fig. 17a. The classes were roughly balanced, except for a slight majority in classes 0 and 1 (Fig. 17b). The relative proportions were maintained in all sampled subsets. Of the roughly 9300 available samples, 2000 were held back for testing. The remaining 7291 training samples were temporarily split into 60% proper training and 40% calibration subsets for conformal synthesis.

Table 12 WINE original and synthetic sample counts. While synthesis generally maintained the class's non-separability, significantly increasing the number of available training samples may nonetheless support Deep Learning generalisation

Classes	Test	$\text{Train}_{\text{orig}}$			$\text{Train}_{\text{syn}}$	$\text{Train}_{\text{ext}}$
	All	Prop.	Calib.	All	All	All
Class 0	536	653	435	1088	29,169	30,257
Class 1	536	653	435	1088	31,149	32,237
All	1072	1306	870	2176	60,318	62,494

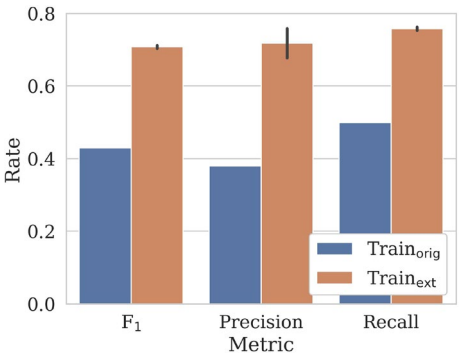
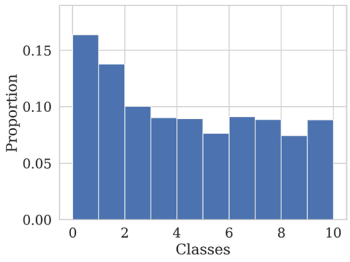


Fig. 16 Mean Deep Learning performance on the WINE dataset. The error bars show the standard deviation. Even though synthesis did not improve the classes' separability, the large quantity of additional samples significantly improved the models' generalisation. All improvements were deemed statistically significant by the Wilcoxon test ($p_W < 0.1$)

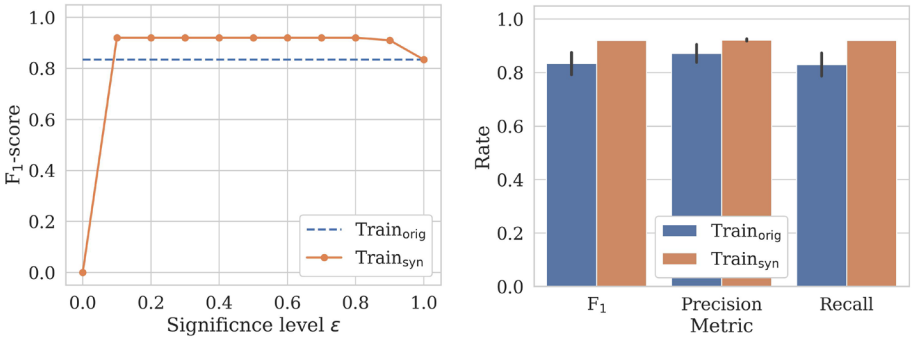
(a) Handwritten digit samples.



(b) Class distribution.

Fig. 17 Overview of the USPS dataset. Each sample is a grey-scale image of a handwritten digit in a 16×16 pixel format (**a**). The classes were roughly balanced except for classes 0 and 1 (**b**)

Conformal synthesis was performed using all available training samples and the grid step $\gamma = 0.01$. Figure 18 summarises the results as ϵ varies. Compared to the baseline on $\text{Train}_{\text{orig}}$, the mean F_1-score performance improved by around 10% points across all ϵ. Since the results were consistent on different $\text{Train}_{\text{syn}}$, we may choose a low significance level to increase confidence. Therefore, Fig. 18b visualises the models' performances for $\epsilon = 0.1$. In addition to the mean of all metrics being improved, the standard deviation was also decreased, indicating a more robust model generalisation. Precision and recall results indicated that our synthesis algorithm preserved the original classes' balance, allowing models trained on the synthetic datasets to maintain or even strengthen their ability to identify relevant samples. The Wilcoxon test confirmed that all reported performance improvements achieved by $\text{Train}_{\text{syn}}$ were statistically significant, with p_W values falling in the range 0.00−0.02.



(a) Model performance with varying ϵ. (b) Model performance with $\epsilon = 0.1$.

Fig. 18 USPS performance results on the baseline $\text{Train}_{\text{orig}}$ and a fully synthetic dataset ($\gamma = 0.01$). The mean of five Deep Learning iterations was reported (**a**), including the standard deviation as error bars in (**b**). All synthetic improvements were found to be statistically significant with the Wilcoxon test ($p_W < 0.1$)

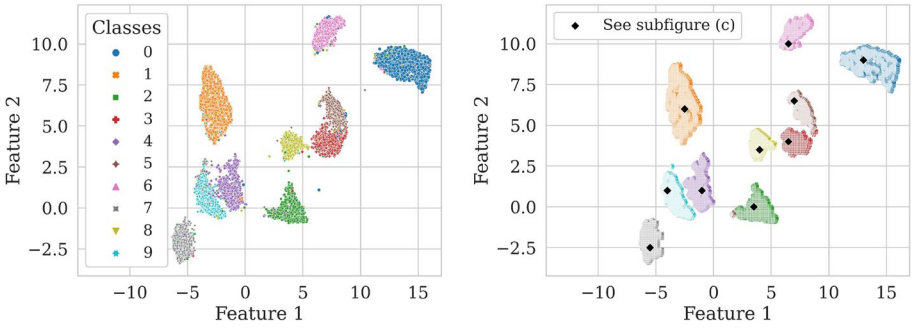
Finally, Fig. 19 investigates the synthesised samples in more detail. Comparing the original feature space and the synthesised feature space after UMAP dimensionality reduction, we see clear parallels in the sample and class distributions. Additionally, inverting a random synthesised test sample per class revealed recognisable digit images (Fig. 19c) similar to the original samples (Fig. 17a), supporting the feature space confidence approach of our proposed algorithm.

5.3.5 Comparison to density-based synthesis

In this section, we compare our conformal synthesis algorithm against a state-of-the-art density-based technique to highlight our proposal's unique properties when generating new points from small datasets. In particular, we employ KDE-based density estimation with a Gaussian kernel and the Euclidean distance metric (Sect. 2). Figure 20a illustrates the synthesis procedure. Regulated by the bandwidth parameter w, new data points may be synthesised from the data's density estimation. Intuitively, w regulates the estimator's bias-variance trade-off. Larger values lead to overly smooth estimates. In contrast, smaller values cause estimates that are too strongly influenced by the data's variance. To select a value for w, we must rely on empirical performance or heuristics that may produce sub-optimal results. We employed the Silverman rule of thumb, a popular heuristic (Belhaj, 2024). In contrast, our conformal synthesis algorithm (Sect. 4) relies on the confidence of feature space regions, thresholded by the significance level ϵ (Fig. 20b). Unlike the bandwidth parameter w, ϵ provides a statistically meaningful boundary based on hypothesis testing, discussed in Sect. 4.2.

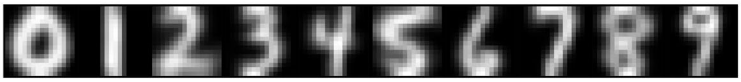
The synthesis performance was evaluated on the most difficult variant of each dataset with the conformal synthesis parameters used in Sect. 5.3:

- MNIST (D_{500}): $\epsilon = 0.2$, $\gamma = 0.01$.
- MSHRM ($D_{1:9}$): $\epsilon = 0.1$, $\gamma = 0.01$.
- WINE: $\epsilon = 0.2$, $\gamma = 0.01$.
- USPS: $\epsilon = 0.1$, $\gamma = 0.01$.



(a) $\text{Train}_{\text{orig}}$ points per class.

(b) $\text{Train}_{\text{syn}}$ points per class.



(c) Synthesised samples corresponding to the diamond markers in (b).

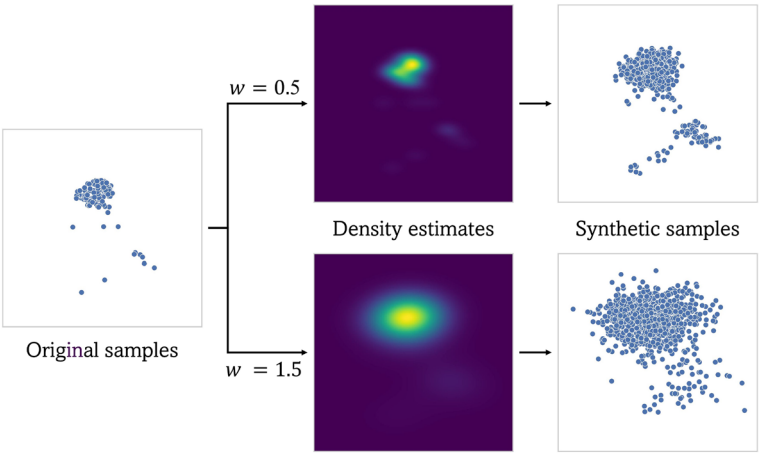
Fig. 19 Original (**a**) and generated samples (**b**) of the USPS dataset after UMAP processing. Conformal synthesis successfully maintained the distribution of samples and classes in the feature space. Additionally, inverting the UMAP transformation on random synthesised points (**c**) revealed recognisable digits similar to the original samples (Fig. 17a)

Since the number of synthesised samples follows from the conformal synthesis parameters, we synthesised the same number of samples from the KDE model to ensure a fair comparison. The sample statistics per dataset are summarised in Table 13, with class-conditional details in Sect. 5.3. All models were trained on the extended $\text{Train}_{\text{ext}}$ sets and evaluated on the same held-back test sets except for USPS models, which were trained on the synthesised samples $\text{Train}_{\text{syn}}$ instead.

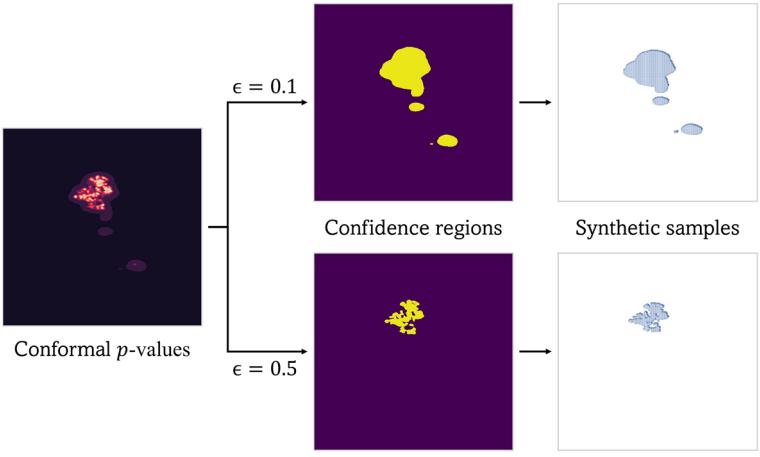
Table 14 presents the mean results across five training iterations. While KDE tended to improve on the baseline models trained on the original datasets (Sect. 5.3), our confidently synthesised training sets outperformed KDE on all datasets. The likely cause was that the KDE bias-variance trade-off was sub-optimal, resulting in under- or overfitted density estimates. However, improving the regulating parameter w is challenging because it does not have a transparent theoretical background compared to the parallel conformal ϵ (Sect. 4.2). The techniques' performance differences were most prominent on the MSHRM and WINE datasets (+53 and +28% points F_1-score, respectively), indicating the potential of conformal synthesis for challenging datasets with imbalanced and overlapping classes in particular. The Wilcoxon test confirmed that all performance improvements were statistically significant.

6 Practical advice and future work

Small sample counts, data imbalances, and overlapping classes are ubiquitous challenges that regularly reduce prediction performances on real-world datasets. Through extensive experimentation (Sect. 5), we have comprehensively showcased the potential of conformal



(a) Density estimation synthesis with KDE using bandwidth w.



(b) Confident synthesis with the proposed algorithm using the significance level ϵ.

Fig. 20 Comparison of the KDE (**a**) and proposed synthesis algorithms (**b**) on MNIST samples (class = 2). The KDE technique requires random sampling from its learned density distribution, which is heavily influenced by the freely selected bandwidth parameter w. In contrast, conformal synthesis relies on a confidence threshold ϵ, which bounds and defines the synthesis regions

Table 13 Sample counts per dataset for the KDE comparison. Models were trained on $\text{Train}_{\text{ext}}$ as identified by the proposed conformal and KDE synthesis algorithms, except for the USPS dataset, which was evaluated on $\text{Train}_{\text{syn}}$

	Test	$\text{Train}_{\text{orig}}$	$\text{Train}_{\text{syn}}$	$\text{Train}_{\text{ext}}$
MNIST	10,000	500	1,213,241	1,213,741
MSHRM	2681	3143	1,688,440	1,691,583
WINE	1072	2176	60,318	62,494
USPS	2007	7291	406,251	–

Table 14 Comparison of conformal synthesis and density-based synthesis samples. Models were trained on $\text{Train}_{\text{ext}}$ generated by the proposed algorithm (CPS) and by KDE. The mean and standard deviation across five iterations were reported with the best performance per dataset in bold. All CPS improvements were confirmed as statistically significant by the Wilcoxon test ($p_W < 0.1$), marked with *

	F_1-score			Precision			Recall		
	CPS	KDE	p_W	CPS	KDE	p_W	CPS	KDE	p_W
MNIST	**0.79** (.07)	0.62 (.08)	0.00*	**0.80** (.00)	0.66 (.10)	0.02*	**0.79** (.01)	0.67 (.05)	0.00*
MSHRM	**0.95** (.01)	0.42 (.08)	0.00*	**0.95** (.01)	0.65 (.21)	0.02*	**0.95** (.01)	0.55 (.04)	0.00*
WINE	**0.71** (.00)	0.43 (.00)	0.00*	**0.72** (.04)	0.38 (.00)	0.00*	**0.76** (.00)	0.50 (.00)	0.00*
USPS	**0.92** (.00)	0.85 (.04)	0.01*	**0.92** (.01)	0.88 (.03)	0.03*	**0.92** (.00)	0.84 (.04)	0.01*

synthesis to confidently generate new data points for these difficult datasets, ultimately improving prediction performance. It is important to note that the conformal confidence guarantees are somewhat limited by the extension to data synthesis (Sect. 4), and furthermore do not automatically guarantee downstream prediction performance. However, the extensive results analysis in this article indicates that optimising ϵ is closely paralleled by increasing model performance. An interesting avenue to explore in future would be to derive further guarantees about the synthesised data, and to combine conformal synthesis with traditional conformal predictors to investigate the interactions of confidence-aware data generation and prediction.

Apart from the original data (e.g., the number of available samples n), the type and number of generated samples are significantly influenced by a range of synthesis parameters. The primary contributors are the grid step γ and the significance level ϵ. γ can be interpreted as the resolution of the feature space, where the confidence of each grid point is evaluated for synthesis. Smaller values result in more grid points and, consequently, more potential synthesis samples (Sect. 5.2.3). Note that in this article, we synthesised all high-confidence grid points as synthetic samples. More sophisticated sampling techniques could be investigated in future to generate a pre-selected number of synthetic samples. In addition to γ, $\epsilon \in (0, 1)$ represents the confidence threshold above which grid points are included in the high-confidence regions for synthesis (Sect. 5.2.1). Intuitively, smaller ϵ lead to larger confidence areas, increasing the synthesised sample count. In this article, we prioritised low ϵ to maximise the inclusion of correctly-labelled synthesis samples (Sect. 4.2). An alternative approach that merits further investigation would be to maximise ϵ, generating an information-efficient synthetic representation of the original dataset by minimising the number of required samples. Potentially, this approach could be useful for confident data anonymisation tasks in the future.

Additionally, the underlying non-conformity measure (NCM) may significantly influence the feature space confidence regions. Inherited from the Conformal Prediction framework, the NCM drives the proposed synthesis algorithm's capacity to effectively distinguish between low and high confidence regions without affecting conformal validity (Sect. 3.1). In this article, we employed a KNN-based NCM to evaluate feature space confidence as a robust baseline (Sect. 5.1.2). Future work could investigate more sophisticated definitions to improve synthesis and subsequent prediction performances.

Theoretically, the proposed synthesis algorithm is compatible with any dimension and complexity of feature spaces. However, the confidence region computations may become prohibitively expensive for high-dimensional datasets (Eq. 19, Sect. 4.2). Therefore,

techniques to reduce the investigated feature space are vital for practical use. For example, this challenge may be addressed with dimensionality reduction techniques (e.g., UMAP), scaling features to a smaller range, and choosing an appropriate resolution of the feature space with the grid step γ. Avenues to improve the synthesis algorithm's complexity in the future include pre-selecting feature space regions of interest for which confidence scores are calculated rather than processing the entire space.

7 Conclusion

In conclusion, we have presented a unique conformal data synthesis algorithm that utilises label-conditional feature space confidence for the data generation process. In addition to a systematic investigation of our proposal's parameters and characteristics, we presented extensive empirical experiments on five benchmark datasets. The comprehensive results demonstrated our algorithm's advantages for a variety of ubiquitous real-world challenges:

- Synthesising new data to significantly boost the sample size of small datasets, as well as correcting class imbalances,
- Supporting a model's learned feature space representations of non-separable classes,
- And replacing an entire dataset with synthetic samples, maintaining Deep Learning prediction performance.

While our proposed algorithm is capable of synthesising any data, generating associated ground truths is currently limited to classification labels and may be an interesting avenue for further extension.

Author contributions All authors contributed to the study's conception and design. J.M. carried out the experimentation, performed the analysis, and drafted the manuscript under K.N.'s supervision. All authors critically reviewed and approved the final manuscript.

Funding No funding was received for conducting this study.

Data availability The benchmark datasets evaluated in this article are available from the UCI Machine Learning repository (MNIST dataset (LeCun et al., 2021), Mushroom dataset (Schlimmer, 1987), Wine dataset (Cortez et al., 2009)) and from Kaggle (USPS dataset (Hull, 1994)).

Code availability The implementation is available at https://github.com/juliameister/conformalised-data-synthesis.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Abdusalomov, A. B., Nasimov, R., Nasimova, N., Muminov, B., & Whangbo, T. K. (2023). Evaluating synthetic medical images using artificial intelligence with the GAN algorithm. *Sensors*. https://doi.org/10.3390/s23073440

Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*. https://doi.org/10.1016/j.jjimei.2020.100004

Alauthman, M., Al-qerem, A., Sowan, B., Alsarhan, A., Eshtay, M., Aldweesh, A., & Aslam, N. (2023). Enhancing small medical dataset classification performance using GAN. *Informatics*. https://doi.org/10.3390/informatics10010028

Althnian, A., AlSaeed, D., Al-Baity, H., Samha, A., Dris, A. B., Alzakari, N., Abou Elwafa, A., & Kurdi, H. (2021). Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Applied Sciences, 11*, 1–18. https://doi.org/10.3390/app11020796

Angelopoulos, A., Bates, S., Malik, J., & Jordan, M. I. (2020). Uncertainty sets for image classifiers using conformal prediction. In ICLR 2021–9th international conference on learning representations. New York: OpenReview. https://openreview.net/forum?id=eNdiU_DbM9. Accessed September 14 2024.

Ashby, A. E., Meister, J. A., Nguyen, K. A., Luo, Z., & Gentzke, W. (2022). Cough-based COVID-19 detection with audio quality clustering and confidence measure based learning. In U. Johansson, H. Boström, K. A. Nguyen, Z. Luo, L. Carlsson (Eds.), *Proceedings of machine learning research* (vol. 179, pp. 129–148). Norfolk: PMLR. https://proceedings.mlr.press/v179/ashby22a.html. Accessed September 14 2024.

Bashir, D., Montañez, G. D., Sehra, S., Segura, P. S., & Lauw, J. (2020). An information-theoretic perspective on overfitting and underfitting. In *AI 2020: AI 2020: Advances in artificial intelligence* (vol. 12576 LNAI, pp. 347–358). Cham: Springer. https://doi.org/10.1007/978-3-030-64984-5_27

Bauer, A., Trapp, S., Stenger, M., Leppich, R., Kounev, S., Leznik, M., Chard, K., & Foster, I. (2024). Comprehensive exploration of synthetic data generation: A survey. arXiv:2401.02524. Accessed September 14 2024.

Bejani, M. M., & Ghatee, M. (2021). A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review, 54*(8), 6391–6438. https://doi.org/10.1007/s10462-021-09975-1

Belhaj, E. I. (2024). A modified rule-of-thumb method for kernel density estimation. *Journal of Mathematical Problems, Equations and Statistics, 5*(1), 143–149. Accessed 14/10/2024.

Bhattarai, B., Baek, S., Bodur, R., & Kim, T. -K. (2020). Sampling strategies for GAN synthetic data. In *ICASSP 2020–2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2303–2307). New York: IEEE. https://doi.org/10.1109/ICASSP40776.2020.9054677

Borji, A. (2022). Pros and cons of GAN evaluation measures: New developments. *Computer Vision and Image Understanding, 215*, 103329. https://doi.org/10.1016/j.cviu.2021.103329

Brigato, L., & Iocchi, L. (2021). A close look at deep learning with small data. In *2020 25th international conference on pattern recognition (ICPR)* (pp. 2490–2497). New York: IEEE. https://doi.org/10.1109/ICPR48806.2021.9412492

Brophy, E., Wang, Z., She, Q., & Ward, T. (2023). Generative adversarial networks in time series: A systematic literature review. *ACM Computing Surveys, 55*(10), 1–31. https://doi.org/10.1145/3559540

Campagner, A., Barandas, M., Folgado, D., Gamboa, H., & Cabitza, F. (2024). Ensemble predictors: Possibilistic combination of conformal predictors for multivariate time series classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2024.3388097

Cherubin, G., Nouretdinov, I., Gammerman, A., Jordaney, R., Wang, Z., Papini, D., & Cavallaro, L. (2015). Conformal clustering and its application to botnet traffic. In *2015 statistical learning and data sciences (SLDS): Lecture notes in computer science* (vol. 9047, pp. 313–322). Cham: Springer. https://doi.org/10.1007/978-3-319-17091-6_26

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems, 47*(4), 547–553. https://doi.org/10.1016/j.dss.2009.05.016

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Wine quality dataset. https://doi.org/10.24432/C56S3T

Ding, T., Angelopoulos, A. N., Bates, S., Jordan, M. I., & Tibshirani, R. J. (2023). Class-conditional conformal prediction with many classes. In *Proceedings of the 36th advances in neural information processing systems conference (NeurIPS)*. New York: Curran Associates, Inc. https://papers.nips.cc/paper_files/paper/2023/hash/cb931eddd563f8d473c355518ce8601c-Abstract-Conference.html Accessed September 14 2024.

Du, Y., Quan, Q., Han, H., & Zhou, S. K. (2022). Semi-supervised pseudo-healthy image synthesis via confidence augmentation. In *2022 IEEE 19th international symposium on biomedical imaging (ISBI)* (pp. 1–4). New York: IEEE. https://doi.org/10.1109/ISBI52829.2022.9761522

Falxa, M., Babak, S., & Le Jeune, M. (2023). Adaptive kernel density estimation proposal in gravitational wave data analysis. *Physical Review D*. https://doi.org/10.1103/PhysRevD.107.022008

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM, 63*(11), 139–144. https://doi.org/10.1145/3422622. arXiv:1406.2661.

Grnarova, P., Levy, K.Y., Lucchi, A., Perraudin, N., Goodfellow, I., Hofmann, T., & Krause, A. (2019). A domain agnostic measure for monitoring and evaluating GANs. In *Proceedings of the 32nd advances in neural information processing systems conference (NeurIPS)*. New York: Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/hash/692baebec3bb4b53d7ebc3b9fabac31b-Abstract.html. Accessed September 14 2024.

Hernández-Hernández, S., Vishwakarma, S., & Ballester, P. J. (2022). Conformal prediction of small-molecule drug resistance in cancer cell lines. In *Proceedings of the 11th symposium on conformal and probabilistic prediction with applications (COPA). Proceedings of machine learning research* (vol. 179). Norfolk: PMLR. https://proceedings.mlr.press/v179/hernandez-hernandez22a.html. Accessed September 14 2024.

Huang, G., & Jafari, A. H. (2023). Enhanced balancing GAN: Minority-class image generation. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-021-06163-8

Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*(5), 550–554. https://doi.org/10.1109/34.291440

Hull, J. J. (1994). USPS dataset. https://www.kaggle.com/datasets/bistaumanga/usps-dataset. Accessed August 11 2023.

Johansson, U., Boström, H., Löfström, T., & Linusson, H. (2014). Regression conformal prediction with random forests. *Machine Learning, 97*, 155–176. https://doi.org/10.1007/s10994-014-5453-0

Johansson, U., Linusson, H., Lofstrom, T., & Boström, H. (2017). Model-agnostic nonconformity functions for conformal classification. In *Proceedings of the international joint conference on neural networks (IJCNN)*. New York: IEEE. https://doi.org/10.1109/IJCNN.2017.7966105

Jung, S., Park, K., & Kim, B. (2021). Clustering on the torus by conformal prediction. *The Annals of Applied Statistics, 15*(4), 1583–1603. https://doi.org/10.1214/21-AOAS1459

Kammoun, A., Slama, R., Tabia, H., Ouni, T., & Abid, M. (2022). Generative adversarial networks for face generation: A survey. *ACM Computing Surveys, 55*(5), 1–37. https://doi.org/10.1145/3527850

Koshino, K., Werner, R. A., Pomper, M. G., Bundschuh, R. A., Toriumi, F., Higuchi, T., & Rowe, S. P. (2021). Narrative review of generative adversarial networks in medical and molecular imaging. *Annals of Translational Medicine, 9*(9), 821–821. https://doi.org/10.21037/atm-20-6325

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2323. https://doi.org/10.1109/5.726791

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (2021). MNIST database of handwritten digits. https://doi.org/10.24432/C53K8Q

Li, Q., Zheng, Z., Wu, F., & Chen, G. (2020). Generative adversarial networks-based privacy-preserving 3D reconstruction. In *2020 IEEE/ACM 28th international symposium on quality of service (IWQoS)* (pp. 1–10). New York: IEEE. https://doi.org/10.1109/IWQoS49365.2020.9213037

Lincoff, G. H. (1983). The Audubon society field guide to North American mushrooms. *Mycologia, 75*(3), 574. https://doi.org/10.2307/3792705

Liu, S., Jiang, H., Wu, Z., & Li, X. (2022). Data synthesis using deep feature enhanced generative adversarial networks for rolling bearing imbalanced fault diagnosis. *Mechanical Systems and Signal Processing, 163*, 108139. https://doi.org/10.1016/j.ymssp.2021.108139

Liu, L., Zhan, X., Wu, R., Guan, X., Wang, Z., Zhang, W., Pilanci, M., Wang, Y., Luo, Z., & Li, G. (2021). Boost AI power: Data augmentation strategies with unlabeled data and conformal prediction, a case in alternative herbal medicine discrimination with electronic nose. *IEEE Sensors Journal, 21*(20), 22995–23005. https://doi.org/10.1109/JSEN.2021.3102488

Liu, L., Zhan, X., Yang, X., Guan, X., Wu, R., Wang, Z., Luo, Z., Wang, Y., & Li, G. (2022). CPSC: Conformal prediction with shrunken centroids for efficient prediction reliability quantification and data augmentation, a case in alternative herbal medicine classification with electronic nose. *IEEE Transactions on Instrumentation and Measurement, 71*, 1–11. https://doi.org/10.1109/TIM.2021.3134321

Löfström, T., Boström, H., Linusson, H., & Johansson, U. (2015). Bias reduction through conditional conformal prediction. *Intelligent Data Analysis*. https://doi.org/10.3233/IDA-150786

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software, 3*(29), 861. https://doi.org/10.21105/joss.00861

Meister, J. A. (2020). *Conformal predictors for detecting harmful respiratory events*. Royal Holloway, University of London. https://doi.org/10.13140/RG.2.2.28575.02728/1

Meister, J. A., Nguyen, K. A., Kapetanakis, S., & Luo, Z. (2023). A novel deep learning approach for one-step conformal prediction approximation. *Annals of Mathematics and Artificial Intelligence*. https://doi.org/10.1007/s10472-023-09849-y

Messoudi, S., Rousseau, S., & Destercke, S. (2020). Deep conformal prediction for robust models. In *Communications in computer and information science* (vol. 1237 CCIS, pp. 528–540). Cham: Springer. https://doi.org/10.1007/978-3-030-50146-4_39

Moreno-Barea, F. J., Jerez, J. M., & Franco, L. (2020). Improving classification accuracy using data augmentation on small data sets. *Expert Systems with Applications, 161*(0957–4174), 113696. https://doi.org/10.1016/j.eswa.2020.113696

Muramatsu, C., Nishio, M., Goto, T., Oiwa, M., Morita, T., Yakami, M., Kubo, T., Togashi, K., & Fujita, H. (2020). Improving breast mass classification by shared data with domain transformation using a generative adversarial network. *Computers in Biology and Medicine, 119*, 103698. https://doi.org/10.1016/j.compbiomed.2020.103698

Navidan, H., Moshiri, P. F., Nabati, M., Shahbazian, R., Ghorashi, S. A., Shah-Mansouri, V., & Windridge, D. (2021). Generative adversarial networks (GANs) in networking: A comprehensive survey and evaluation. *Computer Networks, 194*, 108149. https://doi.org/10.1016/j.comnet.2021.108149

Ndiaye, E. (2022). Stable conformal prediction sets. In *Proceedings of machine learning research* (vol. 162, pp. 16462–16479). Norfolk: PMLR. https://proceedings.mlr.press/v162/ndiaye22a.html. Accessed September 14 2024.

Nie, D., & Shen, D. (2020). Adversarial confidence learning for medical image segmentation and synthesis. *International Journal of Computer Vision, 128*(10–11), 2494–2513. https://doi.org/10.1007/s11263-020-01321-2

Norinder, U., Spjuth, O., & Svensson, F. (2021). Synergy conformal prediction applied to large-scale bioactivity datasets and in federated learning. *Journal of Cheminformatics, 13*(1), 77. https://doi.org/10.1186/s13321-021-00555-7

Nouretdinov, I., Gammerman, J., Fontana, M., & Rehal, D. (2020). Multi-level conformal clustering: A distribution-free technique for clustering and anomaly detection. *Neurocomputing, 397*, 279–291. https://doi.org/10.1016/j.neucom.2019.07.114

Papadopoulos, H. (2008). Inductive conformal prediction: Theory and application to neural networks. In P. Fritzsche (Ed.), *Tools in artificial intelligence*. IntechOpen. https://doi.org/10.5772/6078

Papadopoulos, H., Vovk, V., & Gammerman, A. (2007). Conformal prediction with neural networks. In *19th IEEE international conference on tools with artificial intelligence (ICTAI 2007)* (vol. 2, pp. 388–395). New York: IEEE. https://doi.org/10.1109/ICTAI.2007.47

Park, S., & Pardalos, P. M. (2024). Deep data density estimation through Donsker–Varadhan representation. *Annals of Mathematics and Artificial Intelligence*. https://doi.org/10.1007/s10472-024-09943-9

Plesovskaya, E., & Ivanov, S. (2021). An empirical analysis of KDE-based generative models on small datasets. *Procedia Computer Science, 193*, 442–452. https://doi.org/10.1016/j.procs.2021.10.046

Pozi, M. S. M., & Omar, M. H. (2020). A kernel density estimation method to generate synthetic shifted datasets in privacy-preserving task. *Journal of Internet Services and Information Security, 10*(4), 70–89. https://doi.org/10.22667/JISIS.2020.11.30.070

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*. https://doi.org/10.1038/s41598-024-56706-x

Renkema, Y., Visser, L., & AlSkaif, T. (2024). Enhancing the reliability of probabilistic PV power forecasts using conformal prediction. *Solar Energy Advances*. https://doi.org/10.1016/j.seja.2024.100059

Salazar, A., Vergara, L., & Safont, G. (2021). Generative adversarial networks and Markov random fields for oversampling very small training sets. *Expert Systems with Applications, 163*, 113819. https://doi.org/10.1016/j.eswa.2020.113819

Salazar, A., Vergara, L., & Vidal, E. (2023). A proxy learning curve for the Bayes classifier. *Pattern Recognition*. https://doi.org/10.1016/j.patcog.2022.109240

Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science, 2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Saxena, D., & Cao, J. (2022). Generative adversarial networks (GANs): Challenges, solutions, and future directions. *ACM Computing Surveys, 54*(3), 1–42. https://doi.org/10.1145/3446374

Schlimmer, J. (1987). Mushroom dataset. https://doi.org/10.24432/C5959T

Sesia, M., & Romano, Y. (2021). Conformal prediction using conditional histograms. In *Advances in neural information processing systems* (vol. 34, pp. 6304–6315). New York: Curran Associates, Inc. https://

proceedings.neurips.cc/paper_files/paper/2021/hash/31b3b31a1c2f8a370206f111127c0dbd-Abstract.html. Accessed September 14 2024.

Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research, 9*(3), 371–421.

Shi, J., Liu, W., Zhou, G., & Zhou, Y. (2023). AutoInfo GAN: Toward a better image synthesis GAN framework for high-fidelity few-shot datasets via NAS and contrastive learning. *Knowledge-Based Systems, 276*, 110757. https://doi.org/10.1016/j.knosys.2023.110757

Thambawita, V., Isaksen, J. L., Hicks, S. A., Ghouse, J., Ahlberg, G., Linneberg, A., Grarup, N., Ellervik, C., Olesen, M. S., Hansen, T., Graff, C., Holstein-Rathlou, N.-H., Strümke, I., Hammer, H. L., Maleckar, M. M., Halvorsen, P., Riegler, M. A., & Kanters, J. K. (2021). DeepFake electrocardiograms using generative adversarial networks are the beginning of the end for privacy issues in medicine. *Scientific Reports, 11*(1), 21896. https://doi.org/10.1038/s41598-021-01295-2

Toccaceli, P., & Gammerman, A. (2019). Combination of inductive Mondrian conformal predictors. *Machine Learning, 108*(3), 489–510. https://doi.org/10.1007/s10994-018-5754-9

Vovk, V., Fedorova, V., Nouretdinov, I., & Gammerman, A. (2016). Criteria of efficiency for conformal prediction. In *Lecture notes in computer science* (vol. 9653, pp. 23–39). Cham: Springer. https://doi.org/10.1007/978-3-319-33395-3_2

Wang, J., Xie, G., Huang, Y., Lyu, J., Zheng, F., Zheng, Y., & Jin, Y. (2023). FedMed-GAN: Federated domain translation on unsupervised cross-modality brain image synthesis. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2023.126282

Whang, S. E., Roh, Y., Song, H., & Lee, J.-G. (2023). Data collection and quality challenges in deep learning: A data-centric AI perspective. *The VLDB Journal, 32*(4), 791–813. https://doi.org/10.1007/s00778-022-00775-9

Yoon, J., Drumright, L. N., & Schaar, M. (2020). Anonymization through data synthesis using generative adversarial networks (ADS-GAN). *IEEE Journal of Biomedical and Health Informatics, 24*(8), 2378–2388. https://doi.org/10.1109/JBHI.2020.2980262

Zhan, X., Wang, Z., Yang, M., Luo, Z., Wang, Y., & Li, G. (2020). An electronic nose-based assistive diagnostic -prototype for lung cancer detection with conformal prediction. *Measurement, 158*, 107588. https://doi.org/10.1016/j.measurement.2020.107588

Zhang, J., Norinder, U., & Svensson, F. (2021). Deep learning-based conformal prediction of toxicity. *Journal of Chemical Information and Modeling, 61*(6), 2648–2657. https://doi.org/10.1021/acs.jcim.1c00208

Zhao, B., & Bilen, H. (2022). Synthesizing informative training samples with GAN. In *NeurIPS 2022 workshop on synthetic data for empowering ML research*. https://openreview.net/forum?id=frAv0jtUMfS. Accessed September 14 2024.

Zhuang, P., Schwing, A. G., & Koyejo, O. (2019). FMRI data augmentation via synthesis. In *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)* (vol. 2019, pp. 1783–1787). New York: IEEE. https://doi.org/10.1109/ISBI.2019.8759585