IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 27, 2020, accepted December 17, 2020, date of publication December 28, 2020, date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047838

Learning With Imbalanced Data in Smart Manufacturing: A Comparative Analysis

YASMIN FATHY[1], MONA JABER[2], (Member, IEEE), AND ALEXANDRA BRINTRUP[1]

[1]Department of Engineering, University of Cambridge, Cambridge CB3 0FS, U.K.
[2]School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4FZ, U.K.

Corresponding author: Yasmin Fathy (yafa2@cam.ac.uk)

This work was supported by the Research England for the Pitch-In Project (Promoting the Internet of Things via Collaborations between HEIs and Industry).

ABSTRACT The Internet of Things (IoT) paradigm is revolutionising the world of manufacturing into what is known as Smart Manufacturing or Industry 4.0. The main pillar in smart manufacturing looks at harnessing IoT data and leveraging machine learning (ML) to automate the prediction of faults, thus cutting maintenance time and cost and improving the product quality. However, faults in real industries are overwhelmingly outweighed by instances of good performance (faultless samples); this bias is reflected in the data captured by IoT devices. Imbalanced data limits the success of ML in predicting faults, thus presents a significant hindrance in the progress of smart manufacturing. Although various techniques have been proposed to tackle this challenge in general, this work is the first to present a framework for evaluating the effectiveness of these remedies in the context of manufacturing. We present a comprehensive comparative analysis in which we apply our proposed framework to benchmark the performance of different combinations of algorithm components using a real-world manufacturing dataset. We draw key insights into the effectiveness of each component and inter-relatedness between the dataset, the application context, and the design of the ML algorithm.

INDEX TERMS Manufacturing analytics, generative modeling, smart manufacturing, imbalanced data, limited failure data, generating synthetic data.

I. INTRODUCTION

Manufacturing process is a broad terminology that encompasses the production of final products either handmade, machine-made, or hybrid. This often entails the transformation of raw material into the final product through various mechanical, chemical or other industrial processes. Smart Manufacturing (SM), also referred to as *Industry 4.0*, has transformed the traditional linear manufacturing into a dynamic and digital ecosystem to improve product quality, operations efficiency, and production yield. The Internet of Things (IoT) is a network of connected devices, such as sensors and actuators, that gives operators access to data from the real world in real-time whilst working from the comfort of their desks. In the context of SM, IoT offers infinite possibilities in terms of remote monitoring, maintenance, and control of operations. A manufacturing process is typically a chain of complex tasks where the quality of the final product depends on the entire chain of production. By embracing the IoT paradigm, SM leverages IoT data to drive and automate intelligent decisions, thus improving the efficiency and quality of each task in the chain, not the least the final product.

In traditional manufacturing, intermediate quality tests are performed during the production process to detect quality issues and identify defective batches. However, these tests are time-consuming and not conclusive as they do not allow for full physical inspections of a production line. On the other hand, the early detection of defects at early product processing stages is one of the most effective ways of reducing costs, saving time, and boosting operational efficiency [1]. SM empowered with IoT and machine learning (ML) techniques offers the ability of early defect detection and the opportunity to capitalise on its benefits. IoT sensor data has been used for predictive maintenance in industrial machines [2] and automotive manufacturing [3]. These works; however, are not designed to deal with data imbalance. For instance, authors in [3] use cluster-based

The associate editor coordinating the review of this manuscript and approving it for publication was Qingchao Jiang.

methods for detecting and disregarding data outliers in order to improve fault detection during the manufacturing process. Other existing works such as [4] show that classification performance for imbalanced data can be improved when data outliers are eliminated.

Modern prediction systems rely on using IoT data and ML techniques to predict the expected quality level of forthcoming products. Predictive analytics is one application of ML that analyses current and historical data to make predictions about future events. With predictive analytics, SM manufacturers can discover intricate patterns from collected IoT data, identify processing batches that drop below a defined quality level and perform the best course of actions among multiple options. Consequently, quality engineers can use this information to either immediately adjust the process parameters or stop a particular defective batch processing.

In today's multimode manufacturing processes, not all anomalies or faults would affect the product's quality. Thus, in order to make the process of quality monitoring more purposeful and accurate, a tailored performance indicator method has been proposed by Song *et al.* [5]. The proposed method considers the influence of the fault on product quality and process safety by constructing sub-spaces that enable distinguishing between various types of quality-related and safety-related faults in complex industrial systems. Predicting quality in a real-time is essential for process monitoring and control, but measuring quality variables often require offline analysis. However, quality variables can be measured by exploring the correlation between the changes in process variables and the collected quality information [6]. To this end, a multi-subspace elastic network method is employed in [6] to construct the correlation relationship between process variables and quality variables for the detection and diagnosis of faults. Recently, a novel data-driven method has been proposed in [7] using multi-subspace orthogonal canonical correlation analysis for identifying quality-affecting faults in a real-time fashion.

Despite the recent success of collecting IoT data for process monitoring [8], this data mirrors the actual manufacturing process and its intrinsic bias towards good performance. It is, of course, fortunate that instances of faultless samples largely outnumber the defects and faults in the manufacturing process. This characteristic is a major factor in today's economy as it renders the manufacturing of complex products cost-effective and the end product accessible to masses. For instance, two-thirds of the world population today can afford a mobile phone.[1] On the other hand, data imbalance is a major challenge for ML-based predictive analytics which rely on data for learning.

In a binary classification ML task (e.g., faulty/faultless), imbalanced data is where the number of negative samples (faultless or samples that conform to the quality control process) outweighing the number of positive samples (faulty samples). Canonical ML algorithms often assume that each class has roughly the same number of objects [7], [9]. Manufacturing datasets, however, often have a dramatically skewed distribution. Thus, their application to canonical ML algorithms fails to deliver reliable results. Indeed, when positive samples are limited, predictive models tend to be biased towards the majority (negative) class. This leads to a high probability of misclassifying samples from the minority class. In the context of manufacturing, this bias in predictive models results in the majority of faults going unnoticed and significantly impacting the quality and efficiency of production. In fact, the impact of not predicting a fault in manufacturing is much more detrimental to the quality and process than misclassifying a faultless sample as faulty.

To this end, there is a dominant incentive in SM to improve the performance of quality predictive models that deal with imbalanced datasets. There are various efforts to address this issue, where some propose to remove the bias by manipulating the dataset, referred to as data-based, and others propose to implement a positive bias in the ML algorithm, referred to as algorithm-based. Data-based methods look at generating new synthetic data samples from the minority class to reach similar numbers as the majority class. These methods are often referred to as *data augmentation* tools as they increase the number of samples by adding the synthetic data. For instance, authors in [10] use such a method (Synthetic Minority Over-sampling TEchnique (SMOTE) [11], [12]) to mitigate data imbalance while predicting product quality.

The second group relates to algorithm-based methods, also referred to as cost-sensitive learning. An artificial bias is implemented in the existing classification process through a cost matrix that amplifies the penalty value for misclassifying minority samples. An algorithm-based method was recently used with an imbalanced dataset to predict failure in air pressure systems in [13]. A promising direction looks at combining both approaches to form a hybrid technique that alters the dataset as well as the algorithm bias to circumvent the imbalance obstacle.

However, these methods have not been evaluated in the context of predictive analytics for manufacturing problems. In fact, there is no one-solution-fits-all, and it is critical to have a framework in which various methods can be assessed in a data-centric and contextual fashion. In this work, we present the first such framework that covers the process of predictive analytics starting at the original dataset and finishing at the context-aware performance evaluation, as shown in Figure 1. In this study, we present a comprehensive comparative analysis of data-based, algorithm-based and hybrid methods for improving the prediction accuracy of manufacturing faults. To this end, we propose the first statistical analysis framework for measuring the effectiveness of each method by quantifying four key aspects. The first looks at measuring the bias embedded in data by adopting entropy concepts. It is evident that the optimum predictive analytics method majorly depends on the dataset and its

[1] https://datareportal.com/global-digital-overview

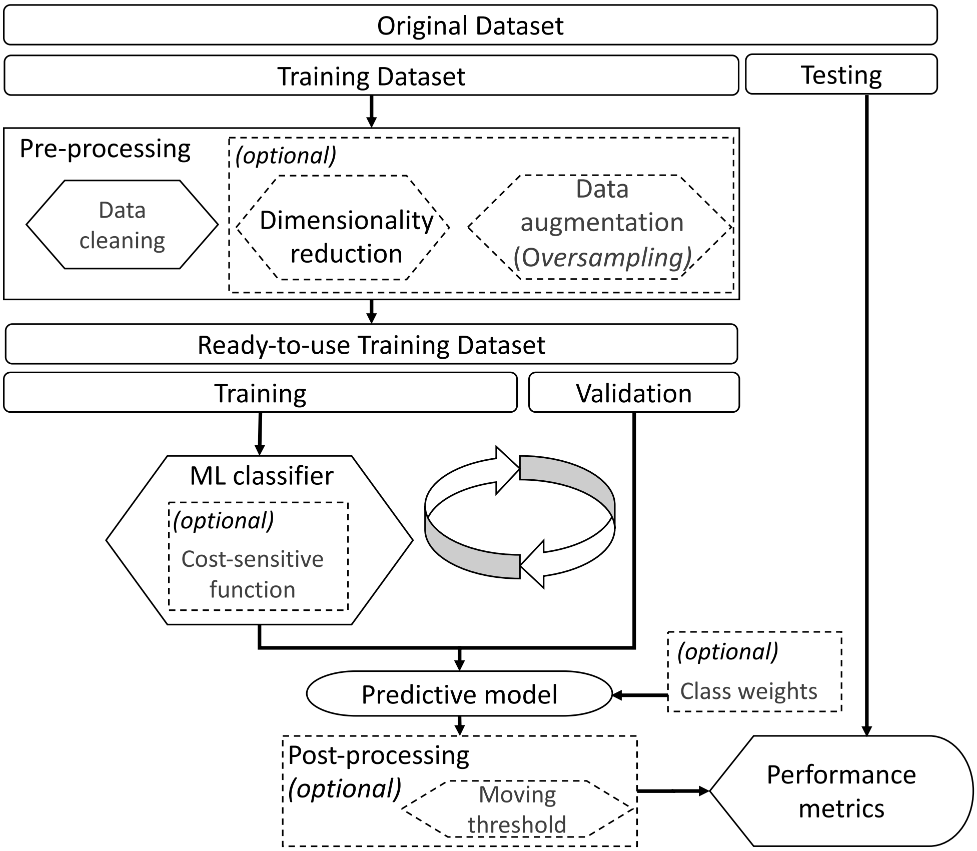


FIGURE 1. Proposed predictive analytics framework for dealing with imbalanced datasets.

intricate features. It is, therefore, of pivotal importance to capture the level of imbalance in the dataset and the resulting behaviour with different re-balancing methods. The second statistical method gauges the goodness of the associated labels (where each represents a cluster or class) using the Silouhette coefficient. Indeed, this metric is essential for understanding the dataset and the source of bias in order to identify a suitable remedy. The third aspect measures the effectiveness of the proposed method (be it data-based, algorithm-based, or hybrid) in predicting defects and identifies relevant metrics such as *Precision* and *Recall* which are used to calculate the *F1-score*. The last aspect only relates to data augmentation methods and aims at measuring the goodness of generated synthetic data in comparison with the real data samples.

Data augmentation is an optional step in the pre-processing phase and may include many methods such as SMOTE and Generative Adversary Networks (GANs) and related variants (see Figure 1). Algorithm-based methods are implemented in the ML classifier, which is selected from a pool of potential classifiers. In addition, data bias can be addressed in the post-processing phase, as shown in Figure 1, by tuning the classification threshold to counter-balance the imbalance.

A. CONTRIBUTIONS

This article aims to provide a better understanding of predictive analytics methods that employ imbalanced datasets with particular focus on manufacturing applications. Our research study aims at assessing the performance of some of the widely used techniques at three levels:

- Pre-processing: Data-augmentation techniques (e.g., SMOTE, GAN) and feature reduction techniques (e.g., Principal component analysis).
- Classifier algorithm: Classifier type (e.g. Linear regression, Random forest) that might incorporate class weights while being trained and tuned or optimised for a given cost-sensitive function (i.e., misclassification cost).
- Post-processing: moving threshold for counter-balancing the data bias.

The performance evaluation phase employs context-aware metrics from a wide range of indicators for measuring the data imbalance, the goodness of class labels, the context-aware success of the classification, and the usability of synthetic data. We have conducted multiple experiments that combine hybrid options within the framework in Figure 1 using a real-world SM dataset.

Although the literature review reveals various works that address the challenge of handling imbalanced datasets in the context of smart manufacturing, none offers a comprehensive study on how intrinsic characteristics of datasets can be exploited in the selection of the ML-based predictive analytics. To this end, the overarching contributions in this work can be summarised as follows:

- Comprehensive review of the challenge of classification using imbalanced datasets and corresponding widely used techniques.
- The first complete evaluation framework to enable the assessment of various combinations of techniques.
- A context-aware set of performance indicators to gauge the effectiveness of predictive models and their impact on the application at hand.
- The first comparative analysis of multiple experiments in which we apply the designed framework using a real manufacturing dataset and draw novel insights.

B. OUTLINE

The paper is structured as follows. The problem formulation is explained in Section II. Section III provides the required background and related work. The adopted methodology in the implementation of various techniques for data augmentation and cost-sensitive functions is detailed in Section IV. The performance metrics for context-aware evaluation of predictive models are described in Section V. In Section VI, we present the manufacturing dataset that is used in the comparative analysis, and we devise four cases that encompass hybrid combinations of data-based, algorithm-based, and post-processing techniques. In Section VII, we present the results from each of the cases and draw context-aware insights in the discussion. We finally conclude the paper and explain the future directions of our research in Section VIII.

II. PROBLEM FORMULATION

Consider a network of N sensor nodes that are deployed across the manufacturing process to predict quality issues such as defective batches, component failure, among others. Each sensor node s_n, where $n = 1, 2 \cdots N$, collects data steams of particular type, e.g. temperature, flow, and motor rotation. Data streams are a sequence of numerical measurements in consecutive order. At each time instance $t \geq 0$, each sensor node s_n gathers a single data stream. Let $x(t) \in \mathbb{R}^N$ be a data sample or instance that is collected from N sensors at a time t. Let $X \in \mathbb{R}^{M \times N}$ be a sequence of M data samples (i.e., features/variables) that are collected by N senors where each has a label or class y_j to indicate whether a failure has occurred ($y_j = 1$) or not ($y_j = 0$) at the machine or component being observed such that $y_j = \{0, 1\}$, where $j = \{1, 2 \cdots M\}$. Let $Y \in \mathbb{R}^M$ be discrete labels whose values are to be modelled and predicted by the input data X. X and Y can be formulated as follows:

$$X = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix}}_{M \text{ data samples from } N \text{ sensors}} \; Y = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}}_{\text{labels}} \tag{1}$$

where the data samples $X = \{x_i, \cdots, x_M\}$ such that the first data sample $x_1 = \{a_{11}, a_{12}, \cdots a_{1N}\}$ is collected by N sensor; a_{ij} is a single measurement value for a particular sensor at a certain collected sample and i and j are sample id, and sensor id, respectively. Moreover, $\{a_{11}, a_{21}, \cdots a_{M1}\}$ are data streams; a sequence of numerical observations collected by sensor s_1.

The problem can be formulated as an imbalanced classification predictive modelling; a process of predicting quality issues that are categorised into classes/labels (e.g. failure/non-failure) by approximating a mapping function f from input data samples $X = \{x_i, \cdots, x_M\}$ into discrete output labels Y. We assume that there is a high-imbalanced ratio between the minority and majority classes in Y. Minority class refers to the class that has few samples in the data X, while the majority class refers to the class that has many samples in the same data. The ratio between these two types of samples is referred to as *imbalanced ratio*. Imbalance Ratio (IR) is defined as a proportion samples in the number of majority class ($y_j = 0$) to the number of minority class ($y_j = 1$).

In this article, our dataset consists of data samples X and their corresponding discrete labels Y. To this end, we refer to the training samples in our dataset as $\mathcal{D} = \{(x_i, y_j), \cdots (x_M, y_M)\}$ in a binary classification task, where the i-th example pair in $\mathcal{D}$ denotes a data sample (i.e., feature vector) taken by N sensors at a time t and is labelled (through y_j) as either a normal sample (when $y_j = 0$) or faulty one (when $y_j = 1$). To this end, in our binary classification case, we denote data samples subset containing all faulty samples (i.e., minority class) as $\mathcal{D}_f \subset \mathcal{D} = \{(x_i, y_j)|y_j = 1\}$. Similarly, data samples subset containing all normal or faultless samples (i.e., majority class) as $\mathcal{D}_s \subset \mathcal{D} = \{(x_i, y_j)|y_j = 0\}$.

III. LEARNING WITH IMBALANCED DATA

Recent advancements in Artificial Intelligence (AI) stemming from deep neural networks (DNN) have helped to establish manufacturing predictive quality as an essential category in Smart Manufacturing (SM). Despite their success in many data-driven real-world, the problem of learning from imbalanced data is still a challenging task [14]. Most of the manufacturing applications suffer from having highly-skewed class distributions where there are usually few collected samples about defective batches, or component failures (i.e., minority class) and it is costly to collect more failure samples [1], [14]. As a result, rare instances and events in manufacturing applications aren't easily identified, and it is difficult to apply standards classification techniques while attaining high accuracy in predicting the minority class [15]. Approaches that have been developed to tackle the imbalanced problem can be grouped into three main categories: data-driven methods, algorithmic-based methods and hybrid methods.

A. DATA-DRIVEN METHODS

Data-driven approaches are data preprocessing methods to enhance the learning from imbalanced data models by modifying the training data-set to balance the class distributions (see Figure 1). This modification is based on generating new samples for the minority class or removing samples of the majority class. The former is referred to as Over-sampling, while the latter is referred to as under-sampling. The basic sampling approach is Random Over-sampling, and under-sampling (ROU) aim to balance class distribution by the random elimination of majority class samples or duplicate samples of minority class samples in the training data-set [16]. This results in discarding useful information about the data and performance degradation due to removing potentially useful samples or duplicating samples that might cause over-fitting for the learned models [9], [17]. Over-fitting occurs when the learned model fits too closely to the training data such that it becomes unable to generalise to new data samples [18]. To address this problem, advanced sampling techniques have been proposed to maintain the underlying distribution of the natural grouping in the data while adding new data samples.

Advanced under-sampling strategies aim to preserve important information while learning from imbalanced data. Lin *et al.* [19] have proposed an under-sampling cluster-based technique to maintain a good representation of the majority class in the dataset. The approach is based on clustering the data samples from the majority class such that the number of clusters in the majority class is set to be equal to the number of data samples in the minority class. Moreover, Mani and Zhang [20] have used a K-Nearest Neighbours (KNN) classifier as an under-sampling preprocessing step. KNN tends to reduce the overlapping between minority and majority samples such that majority samples are classified, and samples are selected to be removed based on their distances from minority samples. Density-based under-sampling methods have been developed to retain useful information while reducing the number of majority classes' samples such as Density-based under-sampling (DBU) technique [21], [22]. DBU assumes that similar examples are relatively close to each other, while noisy examples tend to be far from other examples that are associated with the same class in the feature space.

Several informed over-sampling techniques have also been developed to reduce over-fitting and strength class boundaries. Over-sampling methods tend to be more efficient than under-sampling techniques when handling extremely imbalanced big data problem with large imbalanced ratio [12], [23]–[25]. Chawla *et al.* [11] have introduced a Synthetic Minority Over-sampling Technique (SMOTE); a method for creating synthetic data of minority samples by identifying the feature space for the minority samples and considering their k nearest neighbours. In principle, SMOTE creates artificial data samples for the minority class by artificially linear interpolating new samples between already existing minority samples and their nearest minority neighbours.

SMOTE has shown great success for addressing imbalanced data in different industrial applications including, manufacturing process [24], predictive maintenance and failure prediction [26]. Several extensions have been developed to improve upon the original SMOTE algorithm such as Safe-Level-SMOTE [27], and Borderline-SMOTE [28], among others. Safe-Level-SMOTE aims to define safe regions to prevent overlapping between classes and generate less noisy minority samples, while the primary goal of Borderline-SMOTE is to limit the number of generated samples near minority class borders. Similar to SMOTE, ADASYN [29] is an over-sampling method that creates synthesis data samples of the minority classes. However, ADASYN improves the over-sampling process by further reducing the bias introduced by the class imbalance and force the learning algorithm to learn the minority class boundaries adaptively for enhancing the quality of generated minority samples.

Yet, powerful generative models including, Generative Adversarial Networks (GAN) [30] have been successful in generating new samples that are similar to real samples for improving the performance of imbalanced classification tasks. In [31], Decision Tree (DT) classifier using the training data-set generated by GAN achieves comparable results to DT trained on original data-set. Conditional GAN (CGAN) has been developed in [32] for creating synthetic data for prognostics under the conditions of limited failure data availability.

TABLE 1. A cost matrix for binary classification [36]
C: Cost, TP: True Positive, FP: False Positive (i.e., false alarm), TN: True Negative, and FN: False Negative.

Predicted	Actual/True	
	Faulty	Faultless
Faulty	$C_{1,1}$ or $C_{TP} = 0$	$C_{1,0}$ or C_{FP}
Faultless	$C_{0,1}$ or C_{FN}	$C_{0,0}$ or $C_{TN} = 0$

B. ALGORITHMIC-BASED METHODS

Algorithmic-based approaches are often discussed under cost-sensitive methods in the literature. Unlike data sampling methods, cost-sensitive methods do not alter training data-set. Instead, they modify the existing learning algorithms or decision process (e.g. for classification tasks) through a cost matrix such that each class is assigned a misclassification penalty value [9], [33], [34] (see Figure 1). In principle, this family of algorithms alleviate their bias towards majority classes by increasing the cost value of minority groups. This results in increasing the importance of these groups and decreasing the likelihood that the learning algorithm will misclassify them [18]. An example of a cost matrix of a binary classification problem such as failure prediction is shown in Table 1 where $C_{i,j}$ is the misclassification cost, i.e. penalty cost for predicting samples as a class i when their true class is class j. Intuitively, there is no penalty for classifying the samples correctly (i.e., True Positive (TP) and True Negative (TN)). To this end, the diagonal of the cost matrix, where $i = j$ (i.e., $C_{1,1}$ and $C_{0,0}$) is 0. The optimisation process of the predictive models for imbalanced data shifts from maximising the overall accuracy or minimising error rate, to minimising total cost such that:

$$T_{Cost} = m\, C_{FP} + n\, C_{FN} \tag{2}$$

where T_{Cost} is the total cost that should be minimised, and m and n are the number of errors for false positive (FP) and false negative (FN) classification, respectively. More preciously, m is the number of faultless samples that are predicted as faulty samples, and vice versa for n. In real-world manufacturing applications, FN errors cost more than FP errors i.e., $C_{FN} \gg C_{FP}$ [18]. For instance, $C_{FP} = 10$ and $C_{FN} = 500$ for predicting air pressure system failures in Scania trucks [13]. In such a case, C_{FP} refers to the cost that an unnecessary check needs to be done for a truck by a mechanics. In contrast, C_{FN} refers to the cost of missing a faulty truck, which may cause a breakdown and put drivers and their road fellows at high risk.

The total cost T_{Cost} in Eq. 2 is non-normalised cost. Without loss of generality, we can express the normalised cost as $\hat{T}_{Cost}$ with respect to n as follows:

$$\hat{T}_{Cost} = \lambda\, C_{FP} + \; C_{FN} \tag{3}$$

where the coefficient λ indicates the relative importance of various misclassification costs such that ($\lambda = \frac{m}{n}$) [35]. If the cost-sensitive classifier produces posterior probability estimates P for test samples instead of discrete labels, the cost matrix will rely on defining a classification threshold p^* [36] such that:

$$p^* = \frac{C_{FP}}{C_{FP} + C_{FN}} \tag{4}$$

In such a case, the learning algorithm classifies test samples as faulty samples if their posterior probability estimates $P \geq p^*$. This type of threshold moving methods is sometimes called *Relabelling* and used as a post-processing step for relabelling the output class of test samples [18], [36]. The performance of cost-sensitive algorithms mainly relies on incorporating an effective cost matrix that modifies the learning process. However, the actual cost matrix is often unknown in most of the real-world applications, and it can be empirically defined or by domain experts [9], [18].

Ensemble-based learning algorithms are another type of cost-sensitive methods for tackling imbalanced class distributions. Krawczyk *et al.* [37] introduce a one-class ensemble learning algorithm for improving predictive classifier of a multi-class imbalanced problem with complex and imbalanced class distribution. The proposed ensemble learning algorithm aims to create individual descriptions of each class, and then combining them to have a classifier that outperforms each of them. Such classifier reduces the bias towards one of the classes introduced by standard classifiers [38]. Bagging [39], AdaBoost [40] and Gradient Boosting [41] are the most common ensemble classifier algorithms. Boosting is considered an iterative algorithm that associates different weights on data distribution. The learned algorithm is forced to focus more on the misclassified samples at each iteration. This is achieved by increasing the associated weights for the misclassified samples and decreases the weights associated with correctly classified samples. Other different ensemble approaches are also discussed in [42].

C. HYBRID METHODS

Hybrid methods take the advantages of data-driven and algorithmic-based methods. These two categories have been integrated in various ways. For instance, data-driven solutions are combined with classifier ensembles to mitigate the effect of the imbalanced data [43]. Other approaches such as [44], [45] combine cost-sensitive and over-sampling approaches based on data density to generate better samples around each minority group and eliminating the noise effect for imbalance learning. It is worth-noting that over-sampling approaches have shown little sensitivity when misclassification costs change [46].

Yang *et al.* [25] introduce a hybrid optimal ensemble classifier (HOEC) framework that outperforms other conventional and ensemble classifier methods for learning from imbalanced real-world data sets. HOEC combines density-based under-sampling and cost-effective methods through a multi-objective optimisation process for overcoming the limitations of traditional under-sampling and cost-sensitive algorithms. Several cost-sensitive methods have been developed based on traditional decision trees to improve the imbalanced classification performance of the minority class. Li *et al.* [47] developed a hybrid decision tree that incorporated both a misclassification cost and a set of selected attributes. The attributes selection criterion is based on a linear combination of the Gini index and information gain.

A hybrid framework that incorporates data clustering, data sampling and ensemble is proposed in [48] for improving the performance evaluation of binary classifier. The proposed hybrid framework outperforms the traditional over-sampling techniques including, SMOTE.

Overall, this article studies the improvement of the performance of predictive quality analytics under the condition of limited faulty data availability. We present a comparison of state-of-the-art over-sampling approaches for generating samples for minority groups (e.g., faulty data). We also study a hybrid approach between sampling and cost-sensitive model. With the use of statistical analysis, we measure the quality of over-sampling data techniques and their effects on alleviating the bias towards majority groups (e.g., faultless or normal samples) by increasing importance and the cost values (i.e., penalties) of minority groups. To the best of our knowledge, so far, such a comparison spanning various over-sampling techniques for predicting quality analytics in manufacturing has not been carried out.

IV. METHODOLOGY FOR PREDICTIVE QUALITY ANALYTICS

In this section, we present the methodology adopted in the evaluation of various combinations of data-based and algorithm-based methods for dealing with imbalanced datasets. As discussed in Section II, our work focuses on a manufacturing problem that aims to predict defects in the end product. To this end, the dataset considered in this evaluation contains samples that are labelled as either positive (minority class) or negative (majority class).

In the following paragraphs, we summarise the standard synthetic minority over-sampling technique (SMOTE), and generative models, including Generative Adversarial Network (GAN) and their variants in terms of their objective functions and their architectures. We evaluate these techniques for creating high-quality minority data samples and measure their effectiveness in conjunction with cost-sensitive learning algorithms on improving the classification performance when presented with an imbalanced dataset. There are very few works that use some of these presented techniques as data augmentation such as [26], [49] while learning from an imbalanced dataset, especially in industrial settings. To the best of our knowledge, this is the first comprehensive evaluation of different data augmentation and cost-sensitive methods for predictive analytics in manufacturing applications.

A. SYNTHETIC OVER-SAMPLING TECHNIQUES

As mentioned previously in Section III-A, SMOTE [11] is one of the main over-sampling methods for creating synthetic data samples of the minority class (e.g., faulty samples). SMOTE firstly selects a random data sample x_1 from minority samples then finds a k nearest neighbours minority samples. An example is shown in Figure 2 to demonstrate the process followed in SMOTE in which we assume that $k = 6$. In this example, the neighbouring samples of faulty sample x_1 are indicated as $\{x_2, x_3, \cdots, x_7\}$. Then, for each of the pairs $\{(x_1, x_2), (x_1, x_3), \cdots, (x_1, x_7)\}$, a synthetic faulty data sample is created along the line segments joining them, shown as red triangle in Figure 2.

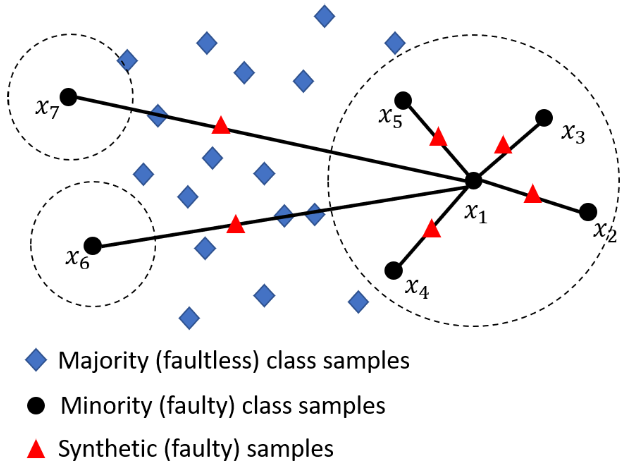

x_7
x_5
x_3
x_1
x_2
x_6
x_4
Majority (faultless) class samples
Minority (faulty) class samples
Synthetic (faulty) samples

FIGURE 2. **An example of SMOTE for $k = 6$.**
- Before over-sampling: $M = 24$ with $M_{min} = 7$ (black circle) and $M_{maj} = 17$ (blue diamonds)
- After over-sampling: $M = 30$ with $M_{min} = 13$ (black circle and red triangles) and $M_{maj} = 17$ (blue diamonds).

Thus, the newly generated faulty samples are at random positions on the linear vector space from x_1. The number of generated samples relies on $\beta = \frac{M_{min}}{M_{maj}}$, where M_{min} is the number of faulty samples in the minority class after over-sampling and M_{maj} is the number of faultless (i.e., normal) samples in the majority class. In a binary classification task, the total number of samples M is the sum of the number of samples of the minority and majority classes $M = M_{min} + M_{maj}$.

B. GENERATIVE ADVERSARIAL NETWORKS (GAN)

SMOTE synthetically generates new and non-replicated minority samples to alleviate the over-fitting caused by random over-sampling. However, SMOTE tends to neglect the characteristics of the local distribution of data samples as it considers the neighbourhood parameter k globally [50]. This results in generating overlapped and noisy samples [51]. Consequently, SMOTE is not guaranteed to create realistic faulty data samples for manufacturing applications [32].

Goodfellow *et al.* [30] propose Generative Adversarial Networks (GAN) as an alternative over-sampling method for creating synthetic data samples. GAN is a minimax two-player game with an objective function $V(G, D)$ between generator G and discriminator D. The generator and discriminator are two neural networks defined by Multilayer Perceptrons (MLP) with weight vectors θ_g and θ_d, respectively. These two models compete against each other during the training process, and they are trained simultaneously.

GAN-based methods emerged originally as an over-sampling technique to create realistic images to improve the performance of learning algorithms in different applications [52]. In manufacturing applications, GAN-based methods were recently used to create faulty synthetic samples under an imbalanced dataset for improving prediction of faults [32]. The main objective of GAN-based methods is to augment the original training data such that the number of available faulty samples (after generating new synthetic faulty samples) for the training models is increased.

In standard GAN, generator G is trained to fool the discriminator D by capturing the underlying distribution of the real faulty data $\mathcal{D}_f = \{x^{(i)}\}$ of variables $x^{(i)} \sim p_{data}(x^{(i)})$, so that it can create synthetic faulty samples that are intended to come from the same distribution of real faulty data $p_{data}(x^{(i)})$. The discriminator D is trained to recognise fake and real faulty data by estimating the probability that a given data sample originates from the real faulty samples. This zero-sum game between the generator and discriminator motivates both of them to improve their functionalities. The basic architecture of GAN is depicted in Figure 3 (left).

Formally, given faulty data $\mathcal{D}_f = \{x^{(i)}\}$ of a variable $x^{(i)} \sim p_{data}(x^{(i)})$, we wish to estimate $p_{data}(x)$. To this end, we transform a prior white noise variable $z \sim p(z)$ through a generator $G(z; \theta_g)$, parametrised by MLP parameters θ_g, to produce a new synthetic (i.e., fake) faulty data sample. To this end, G implicitly defines a probability distribution

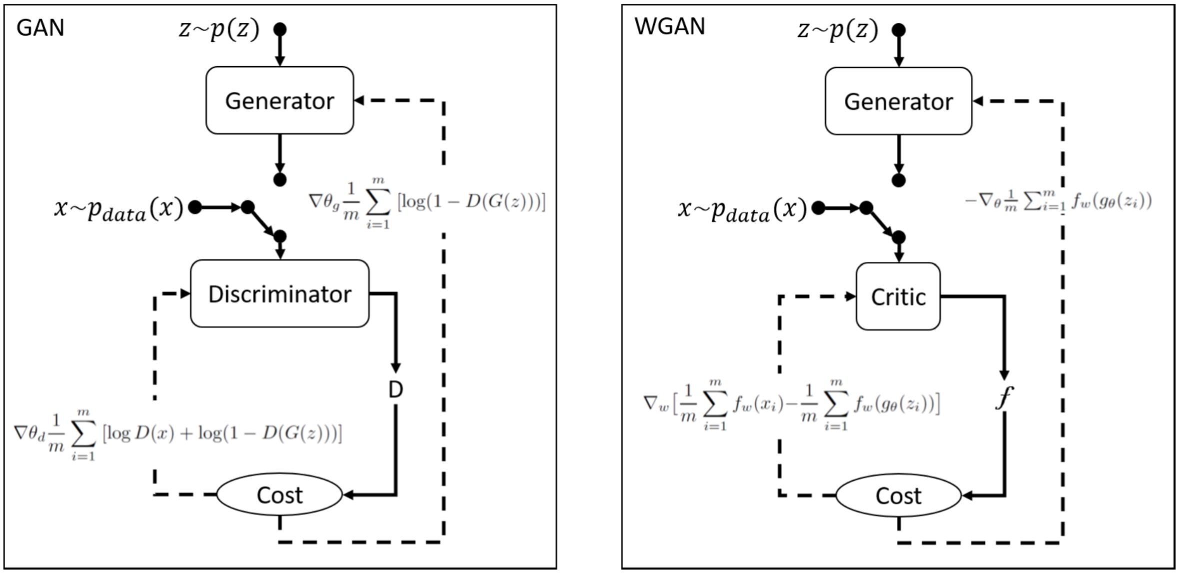

GAN
$z \sim p(z)$
Generator
$x \sim p_{data}(x)$
$\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}[\log(1-D(G(z)))]$
Discriminator
D
$\nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}[\log D(x)+\log(1-D(G(z)))]$
Cost
WGAN
$z \sim p(z)$
Generator
$x \sim p_{data}(x)$
$-\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z_i))$
Critic
$\nabla_w[\frac{1}{m}\sum_{i=1}^{m} f_w(x_i) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z_i))]$
f
Cost

FIGURE 3. Basic GAN (Left) and WGAN (Right) architectures.

p_g as the distribution of the faulty samples $G(z)$ obtained when $z \sim p_z$. The discriminator $D(x; \theta_d)$, parametrised by MLP parameters θ_d is to output the probability estimation that any x comes from the data distribution $p_{data}(x)$. In principle, the discriminator D is a scalar function that is trained to maximise the probability to assign the correct labels to faulty samples in the training data and generated from $G(z)$. In such a case, the discriminator is typically a traditional supervised learning method that is optimised to identify whether any given sample x is a real faulty data sample (i.e., $x \sim p_{data}$) or fake sample (i.e., sampled from generator distribution $x \sim p_g$).

Overall, the main goal of GAN is to learn a distribution p_g over the faulty data samples such that p_g is as close as possible to the original faulty data distribution p_{data}. In such a case, the generator represents a distribution over the distributions of the original data. The training procedure for the generator G is to maximise the probability that the discriminator D makes a mistake in identifying fake and real faulty samples. This is done by increasing the chances of D to produce a high probability for fake examples, thus to minimise $\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$. At the same time, the training process for the discriminator D aims to teach it to identify real faulty data samples accurately by maximising $\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)]$. Meanwhile, given a fake sample sampled from the generator $z \sim p_g(z)$, the discriminator is expected to output a probability, $D(G(z))$, close to zero by maximising $\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$. More specifically, the loss functions of generator G and discriminator D are defined in Eq. 5 and Eq. 6, respectively.

$$\min_G V(G,D) = \mathbb{E}_{z\sim p_z(z)} \overbrace{[\log(1 - D(G(z)))]}^{\text{Optimise G to generate better fake faulty samples to fool D}} \tag{5}$$

$$\max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)} \overbrace{[\log D(x)]}^{\text{D to better identify real faulty samples}} + \mathbb{E}_{z\sim p_z(z)} \underbrace{[\log(1 - D(G(z)))]}_{\text{D to better identify generated fake faulty data samples}} \tag{6}$$

Formally, both the discriminator D and the generator G play a two-player minimax game with the following main objective function $V(G, D)$ such that the generator tries to minimise it while the discriminator tries to maximise it. More specifically, $V(G, D)$ (shown in Eq. 7) incorporates the loss functions of Eq. 5 and Eq. 6 as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)} \overbrace{[\log D(x)]}^{\text{D output for real faulty data}} + \mathbb{E}_{z\sim p_z(z)} \underbrace{[\log(1 - D(G(z)))]}_{\text{D output for generated fake faulty data G(z)}} \tag{7}$$

In Eq. 7, the generator does not have a direct effect on the first term $\log(D(x))$ in the objective function. For the generator, minimising the loss is equivalent to minimising $\log(1 - D(G(z)))$.

1) GLOBAL OPTIMALITY IN GAN

The global optimality of $V(D, G)$ is only achieved when both D and G are at their optimal values. In such a case, p_g becomes very close to p_{data}. The training objective for the discriminator D can be described as maximising the log-likelihood for estimating a conditional probability $P(Y = y|x)$, where $Y = \{0, 1\}$ indicates whether x comes from real data failure p_{data} (with $y = 1$) or a synthetic or fake failure sampled from p_g (with $y = 0$) [30]. The optimal discriminator should be able to identify the real failure data and generated fake failures. This can be achieved when the real failure data distribution p_{data} and generated failure data distribution p_g are known. In such a case, the optimal discriminator $D^*_G(x)$ for any fixed generator G is expressed in Eq. 8. Indeed, when $p_g = p_{data}$, the optimal discriminator $D^*_G(x) = \frac{1}{2}$.

$$D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{8}$$

For the optimal discriminator to maximise the quantity $V(G, D)$, Eq. 6 can now be reformulated as in Eq. 9:

$$\begin{aligned}\max_D V(D^*, G) = \mathbb{E}_{x\sim p_{data}(x)} & \overbrace{[\log D^*_G(x)]}^{\text{D to better identify real faulty samples}} \\ + \mathbb{E}_{x\sim p_g(x)} & \underbrace{[\log(1 - D^*_G(x))]}_{\text{D to better identify generated fake faulty data samples}}\end{aligned} \tag{9}$$

In such a case, $V(D, G)$ in Eq. 9 has a value of $-\log 4$ (proof is included in [30]). Using Eq. 8, Eq. 9 can now be reformulated as:

$$\begin{aligned}\max_D V(D^*, G) = & -\log 4 + KL\big(p_{data}||\frac{p_{data} + p_g}{2}\big) \\ & + KL\big(p_g||\frac{p_{data} + p_g}{2}\big)\end{aligned} \tag{10}$$

where KL is Kullback–Leibler divergence [2] that measures how the probability distribution p_{data} diverges from a second expected probability distribution p_q. Intuitively, KL has a minimum value of zero when $p_{data} = p_g$. When $p_data(x)$ gets closer to zero and $p_g(x)$ is non-zero, the effect of $p_g(x)$ will then be ignored and disregarded [53]. However, both distributions are equally important. Moreover, it is very clear that KL divergence is asymmetric measure. Jensen–Shannon Divergence (JSD) [3] is another measure of similarity between two probability distributions. It is a symmetric measure and is bounded by [0, 1]. The advantage of using symmetric JS divergence instead of asymmetric KL divergence while training GAN has been discussed in [53], [54]. It turns out that KL divergence is hard to optimise and that the minimax converges to its equilibrium between the polices of both generator and discriminator when the polices can be updated during the training process while minimising the JSD [55]. To this end, Eq. 10 can be reformulated as follows:

$$V(D^*, G) = -\log 4 + 2\ .\ JSD\ (p_{data}||p_g) \tag{11}$$

As explained previously, the global optimality of $V(D, G)$ is achieved when $p_g = p_{data}$. In such a case, $V(D^*, G)$ is obtained as in Eq. 11.

Algorithm 1: Minibatch Stochastic Gradient Descent Training of GAN for Generating Synthetic Faulty Samples

```
for number of training steps on faulty samples do
    for number of k steps to train discriminator do
        • Sample a minibatch of $m$ noise samples
          $\{z_1, \cdots, z_m\}$ from noise prior $p_g(z)$
        • Sample a minibatch of $m$ samples
          $\{x_1, \cdots, x_m\}$ from data distribution $p_{data}(x)$
        • Update the discriminator models $\theta_d$ by
          ascending its stochastic gradient:
              $\nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}[\log D(x) + \log(1 - D(G(z)))]$
    end
    • Sample a minibatch of $m$ noise samples
      $\{z_1, \cdots, z_m\}$ from noise prior $p_g(z)$
    • Update the generator model parameters $\theta_g$ by
      descending its stochastic gradient:
          $\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}[\log(1 - D(G(z)))]$
end
Generate $N$ synthetic faulty data samples from
learned Generator model
```

2) GAN TRAINING PROCESS

The value functions of both players are typically defined in terms of their model parameters θ_g and θ_d. The discriminator aims to maximise $V(\theta_d, \theta_g)$ while it has only control on θ_d, while the generator aims to minimise $V(\theta_d, \theta_g)$ while only controlling θ_g. The GAN training process (shown in Alg. 1) for cost function $V(G, D)$ (in Eq. 7) includes two gradient steps simultaneously: one updating θ_d to maximise $V(D)$ and one updating θ_g to minimise $V(G)$. Adam [56] is often used as a gradient-based optimisation algorithm for learning both models' parameters. To this end, the discriminator model parameters θ_d is learned for a minibatch of m failure examples by ascending its stochastic gradient in Eq. 12, while generator model parameters θ_g is learned by descending its stochastic gradient in Eq. 13 (see Figure 3 (Left)).

$$\nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}[\log D(x) + \log(1 - D(G(z)))] \tag{12}$$

$$\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}[\log(1 - D(G(z)))] \tag{13}$$

C. CONDITIONAL GAN (CGAN)

In the standard GAN (shown in Alg. 1 and from [30]), the generative model $G(z)$ is trained without having control on the type of faulty data being generated. In Conditional GAN (CGAN) [57], the generator and discriminator are conditioned on auxiliary information y, where y is the failure

[2] $KL(p||q) = \log\frac{p(x)}{q(x)}$ is a term that quantifying the KL divergence between two distributions p and q; it measures how one probability distribution p diverges from a second expected probability distribution q.

[3] $JSD(p||q) = \frac{1}{2}KL(p||\frac{p+q}{2}) + \frac{1}{2}KL(q||\frac{p+q}{2})$ is a term that quantifying the JS divergence between two distributions p and q

class labels. The conditioning on class failure label is formed by feeding the label y into both generator and discriminator. This allows the generator $G(z|y)$ to learn to generate synthetic faulty data samples for a particular type of failure labelled through y. In such a case, the objective function of the minimax game of the standard GAN (in Eq. 7) can be formulated to incorporate class failure label y as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)} \overbrace{[\log D(x|y)]}^{\substack{\text{D output for real}\\ \text{faulty data of type y}}} + \mathbb{E}_{z\sim p_z(z)} \underbrace{[\log(1-D(G(z|y)))]}_{\substack{\text{D output for generated}\\ \text{fake faulty data of type y}}} \tag{14}$$

Similar to the standard GAN, the CGAN training process for cost function $V(G,D)$ includes two gradient steps simultaneously. In principle, Eq. 12 and Eq. 13 are reformulated for CGAN as follows:

$$\nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}[\log D(x|y) + \log(1-D(G(z|y)))] \tag{15}$$

$$\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}[\log(1-D(G(z|y)))] \tag{16}$$

D. WASSERSTEIN GAN (WGAN)

The traditional GAN may result in model collapse whereby the generator reaches a state in which it always produces the same synthetic output (e.g., same faulty samples or image). Wasserstein GAN (WGAN) is an alternative to traditional GAN. It employs the Wasserstein distance measure (W) between two probability distributions in the training process, which allows a smoother gradient. The main goal of WGAN is to provide an efficient approximation of the W distance that provides high synthetic sample quality. As such, the WGAN averts model collapse as it follows a more stable training process and offers better learning for hyperparameter search [58]–[60]. In general, Wasserstein distance $W(p_{data}, p_g)$ between the two distributions is defined as follows:

$$W(p_{data}, p_g) = \inf_{\gamma\sim\Pi(p_{data},p_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \tag{17}$$

where $\Pi(p_{data}, p_g)$ is the set of all joint probability distributions of $\gamma(x,y)$ whose marginals are p_{data} and p_g, respectively. In principle, $\gamma(x,y)$ describes the percentage of faulty data distribution that should be transported from x to y such that the distribution of real faulty samples p_{data} is transformed into p_g that will be used to generate synthetic faulty samples. More precisely, Eq. 17 indicates the cost of such an optimal transport plan. However, the infimum in Eq. 17 makes it intractable. Using Kantorovich-Rubinstein duality [61], Eq. 17 can be simplified to Eq. 18:

$$W(p_{data}, p_g) = \sup_{||f||_L\leq 1} \mathbb{E}_{x\sim p_{data}}[f(x)] - \mathbb{E}_{x\sim p_g}[f(x)] \tag{18}$$

where supremum is over f that is 1-Lipschitz functions where the general form of K-Lipschitz function is $||f||_L \leq K$ for some a Lipschitz constant K. K-Lipschitz function has a constraint to satisfy $|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$, where $K \geq 0, \forall x_1, x_2 \in \mathbb{R}$ and K is independent from x_1 and x_2 [58]. In such a case, $||f||_L \leq 1$ can be replaced (in Eq. 18) by $||f||_L \leq K$ where $K = 1$. In order to solve Eq. 18, we suppose that function f comes from a parameterised family of K-Lipschitz continuous functions $\{f_w\}_{w\in W}$. An assumption has been made in [58] that by attaining the supremum in Eq. 18 for some $w \in W$, $W(p_{data}, p_g)$ can then be calculated. In principle, we can calculate the Wasserstein distance by finding a 1-Lipschitz function that can be learned by DNN parameterised on weights w in a compact space W. By backproping via Eq. 18, $W(p_{data}, p_g)$ can be differentiating via estimating $\mathbb{E}_{z\sim p(z)}[\nabla_\theta f_w(g_\theta(z))]$, where p_g is the distribution of $g_\theta(Z)$ with Z a random variable with density p and f_w is the set of 1-Lipschitz function.

In the view of the Kantorovich-Rubinstein duality for calculating the Wasserstein distance, the value function $V(G,D)$ of minimax game for WGAN can be written as follows:

$$\min_G \max_{D\in F} V(D, g_\theta) = \mathbb{E}_{x\sim p_{data}(x)} \overbrace{[D(x)]}^{\substack{\text{D output for real}\\ \text{faulty data}}} + \mathbb{E}_{z\sim p_z(z)} \underbrace{[D(g_\theta(z))]}_{\substack{\text{D output for generated}\\ \text{fake faulty data}}} \tag{19}$$

where F is the set of 1-Lipschitz functions. Similar to GAN, WGAN has a discriminator D; however, it is not a classifier, but instead, it is a model with a critic that scores the realness or fakeness of a given sample, as shown in Figure 3 (right). More precisely, it's a real-valued function that aims to learn w in a compact space W to find a good f_w [58]. To train the discriminator, Arjovsky *et al.* [58] specify a family of functions f_w by DNN, and a weighted clipping is then applied that aims to enforce the Lipschitz continuity. More precisely, to maintain the Lipschitz continuity of f_w during the training, upon every gradient update, the weights w are clipped to be within a small window, e.g., $[-0.01, 0.01]$. This results in having a compact parameter space W and obtaining the bound of f_w that preserves the Lipschitz continuity.

Analogous to GAN where Jensen-Shannon (JS) divergence implicitly employed, the learning in WGAN is based on minimising the Wasserstein distance between the real faulty sample distribution p_{data} and leaned generator distribution p_g that can be represented as in [58] as follows:

$$W(p_{data}, p_g) = \max_{w\in W} \mathbb{E}_{x\sim p_{data}}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))] \tag{20}$$

Given an optimal discriminator which is also known as a critic (i.e., because it is not trained to classify), minimising the value function in Eq. 19 with respect to generator's parameters minimised $W(p_{data}, p_g)$ in Eq. 20 [62]. The WGAN training procedure to generate faulty synthetic samples is summarised in Alg. 2, and the proof of optimality of WGAN is detailed in [58].

Algorithm 2: WGAN for Generating Synthetic Faulty Samples

```
Require: $\alpha$: learning rate, $c$: clipping parameter, $m$: batch size, $n_{critic}$: number of iterations of the critic per generator iteration, $w_0$: initial critic parameters, and $\theta_0$: initial generator's parameters
while $\theta$ has not converged do
    for $t = 0, \cdots n_{critic}$ do
        • Sample a minibatch of $m$ noise samples $\{z_1, \cdots, z_m\}$ from noise prior $p_g(z)$
        • Sample a minibatch of $m$ samples $\{x_1, \cdots, x_m\}$ from data distribution $p_{data}(x)$
        $g_w \leftarrow \nabla_w[\frac{1}{m}\sum_{i=1}^{m} f_w(x_i) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z_i))]$
        $w \leftarrow w + \alpha.\ \mathrm{RMSProp}(w, g_w)$
        $w \leftarrow \mathrm{clip}(w, -c, c)$
    end
    • Sample a minibatch of $m$ noise samples $\{z_1, \cdots, z_m\}$ from noise prior $p_g(z)$
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z_i))$
    $\theta \leftarrow \theta - \alpha.\ \mathrm{RMSProp}(\theta, g_\theta)$
end
Generate $N$ synthetic faulty data samples from learned Generator model
```

To enforce the Lipschitz constraint without clipping the discriminator's weights, Gulrajani *et al.* [62] have introduced the WGAN gradient penalty (WGAN-GP) which incorporates a penalised gradient such that the norm of the discriminator's output with respect to its input is constrained. This results in reformulating Eq. 19 as follows:

$$L(D) = \overbrace{\mathbb{E}_{x\sim p_g(x)}[D(x)] - \mathbb{E}_{x\sim p_{data}(x)}[D(x)]}^{\text{Original critic loss}} \\ + \underbrace{\lambda \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]}_{\text{Gradient penalty}} \quad (21)$$

where the last term is a soft version of the constraint with a penalty on the gradient norm for random samples $\hat{x} \sim p(\hat{x})$. The training procedure for WGAN-GP incorporating the gradient's penalty is summarised Alg. 3, and the detailed discussion on how WGAN can be improved using Eq. 21 is discussed in [62].

E. WASSERSTEIN CGAN (WCGAN)

Wasserstein CGAN (WCGAN) is often discussed under WCGAN as in [63] or CWGAN as in [64], [65]. Similar to CGAN discussed in Section IV-C, the generator and discriminator in WCGAN can be conditioned on the failure class labels y, as auxiliary information. WCGAN incorporates a penalised gradient to constrain discriminator's output, analogous to the WGAN-GP shown in Alg. 3. However, differently from WGAN-GP in Alg. 3, the objective functions in WCGAN are conditioned on the type of failure y. As such, the objective function of the minimax game of the WGAN-GP (in Eq. 21) can be formulated to incorporate class failure label y as follows:

Algorithm 3: WGAN-GP: WGAN With Gradient Penalty λ for Generating Synthetic Faulty Samples

```
Require: $\lambda$: gradient penalty coefficient, $n_{critic}$: number of iterations of the critic per generator iteration, $m$: batch size, $w_0$: initial critic parameters, $\alpha$: Adam learning rate (i.e., step-size), $\beta_1$, $\beta_2$: Adam exponential decay rates, and $\theta_0$: initial generator's parameters
while $\theta$ has not converged do
    for $t = 1, \cdots n_{critic}$ do
        for $i = 1, \cdots m$ do
            • Sample a real sample $x \sim p_{data}$
            • Sample a latent variable $z \sim p_g(z)$
            • Select a random number $\epsilon \sim U[0, 1]$
            $\tilde{x} \leftarrow G_\theta(z)$
            $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$
            $L_i \leftarrow \mathbb{E}_{x\sim p_g(x)}[D(x)] - \mathbb{E}_{x\sim p_{data}(x)}[D(x)] + \lambda\ \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]$
        end
        $w \leftarrow \mathrm{Adam}(\nabla_w \frac{1}{m}\sum_{i=1}^{m} L_i, w, \alpha, \beta_1, \beta_2)$
    end
    Sample a minibatch of $m$ noise/latent samples $\{z_1, \cdots, z_m\}$ from noise prior $p_g(z)$
    $\theta \leftarrow \mathrm{Adam}\ (\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} -D_w(G_\theta(z)), \theta, \alpha, \beta_1, \beta_2)$
end
Generate $N$ synthetic faulty data samples from learned Generator model
```

$$L(D) = \overbrace{\mathbb{E}_{x\sim p_g(x)}[D(x|y)] - \mathbb{E}_{x\sim p_{data}(x)}[D(x|y)]}^{\text{Original critic loss}} \\ + \underbrace{\lambda \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x}|y)||_2 - 1)^2]}_{\text{with Gradient penalty}} \quad (22)$$

where $\hat{x}$ is a penalty on the gradient norm for random samples $\hat{x} \sim p(\hat{x})$. The main objective function of the generator $L(G)$ is expressed conditionally on a label y as follows:

$$L(G) = -\mathbb{E}_{x\sim p_g(x)}[D(x|y)] \quad (23)$$

Zheng *et al.* [64] have used WCGAN-GP as an oversampling approach to generate realistic minority samples based on learning the real distribution of available minority samples. In comparison to WGAN-GP, the authors show that incorporating the class label in the WCGAN-GP increases the quality of synthetic generative data. The same observation is drawn in [66], where the authors have constrained their WCGAN-based model conditionally on some features (e.g. colours) to generate better cartoon images from sketch images. Furthermore, Qin and Jiang [63] have discussed the main shortcoming of WGAN model in speech enhancement task; the task that aims to improve the performance of speech systems in a noisy environment. Although WGAN learns well the characteristics of the speech data, the model tends to overfit while training on low-data environments. To this end, the authors have introduced an improved objective function upon WCGAN-GP to improve the performance of speech enhancement task. The improved WCGAN-GP conditionally on a variety of features in voice data makes it possible to improve the speech quality. However, there was no comparison with other existing GAN-based models.

Liu *et al.* [67] have tackled the limited availability of training dataset to predict the illegal accesses for the Internet of Vehicles (IoV) applications. In particular, WCGAN has been used to generate synthetic illegal access data such that there is a balance between the legal and illegal access dataset. The simulation results show that WCGAN converges faster than traditional GAN and thus, improves the prediction accuracy and reducing false-negative rate. However, there was no comparison between WCGAN and WGAN.

Recently, WCGAN-based model has also been developed to generate faulty samples conditioned on fault categories to improve the performance of fault diagnosis in industrial applications [65]. In comparison to traditional GAN, the proposed WCGAN-based generates good quality data of faulty synthetic samples. Hence, it improves the prediction accuracy (by 3% after over-sampling faulty examples) for faults and avoids over-fitting. However, there is no comparison to other GAN-based approaches.

Based on these studies, WCGAN-based methods tend to avoid over-fitting the training data and to converge faster in different applications, including industrial scenarios. In this work, we present the first comprehensive evaluation of different GAN-based methods for predictive analytics in manufacturing applications, including WCGAN. Furthermore, we extend the evaluation to encompass data-based, algorithm-based and hybrid methods for improving the prediction accuracy of manufacturing faults. To the best of our knowledge, this is the first statistical analysis framework for measuring the effectiveness of dealing with data imbalance when used in predictive analytics.

V. PERFORMANCE METRICS

In the recent advances made to circumvent the challenge of an imbalanced dataset, it has become apparent that while some methods are successful for a given classification problem, they may ultimately fail in the imbalanced classification task. What are the governing factors that render a method successful? What are the dataset aspects that dictate the correct method to apply? What is an adequate metric to gauge the effectiveness of a method, particularly in the context of manufacturing? This is the first work that offers a statistical analysis framework to answer these questions, where we propose four performance metrics. The first two extract two essential dataset features: the level of skewness or imbalance ratio and the goodness of the allocated label using the Silhouette coefficient. The third measures the effectiveness of the predictive analytics by quantifying revealing indicators such as the precision, recall and F1-scores. The last, applies to data-based methods only as it measures the fit of the synthetic samples in comparison with the real data samples.

A. IMBALANCE RATIO

In a dataset with two classes, an Imbalance Ratio (IR) is defined as a proportion of the number of samples in the majority class to the number of samples in the minority class. Referring to the problem formulation in Section II, the majority class in a manufacturing problem is the number of good samples labelled $y_j = 0$, while the minority represents the defects labelled $y_j = 1$, where $J = \{1, 2 \cdots M\}$ and M is the total number of data samples. This ratio can be calculated using information entropy.

Entropy measures quantify the information about the outcome class, given the class distribution. Shannon's entropy is one of the most widely entropy measures [68]. In general, given a dataset with k number of classes, let Y be a class variable or label with k modalities (i.e., the number of classes), $Y = \{y_1, \cdots y_k\}$ with frequentist probabilities of $p = (p_1, \cdots p_k)$ where $\sum_{i=1}^{k} p_i = 1$ and $p_i \geq 0 \; \forall \; i = 1, \ldots k$. The Shannon entropy H of the probabilities distribution can be computed as follows:

$$H = -\sum_{i=1}^{k} p_i \log p_i \tag{24}$$

where $p_i = \frac{|c_i|}{M}$ is the frequentist probability of a class labelled y_i, with c_i is the cluster of all $|c_i|$ samples with y_i label out of a total of M data samples. With this metric, $H \longrightarrow 0$ if the dataset is very unbalanced and $H \longrightarrow \log k$ if the data is balanced. Normalising the entropy H in Eq. 24 by $\log k$ gives $\hat{H} \in [0, 1]$. Therefore, the imbalance ratio (IR) can be defined as follows:

$$\mathrm{IR} = \frac{-\sum_{i=1}^{k} p_i \log_b p_i}{\log_b k} \tag{25}$$

where IR tends to be 0 when the data is highly imbalanced and 1 when the data is balanced.

B. SILHOUETTE COEFFICIENT

In section III, we have discussed some of the conditional generative methods that create synthetic data samples of minority class dependent on its labels (e.g. type of quality issues or faults). In order to group the minority data samples into k clusters, clustering approaches are used to identify natural groupings of the minority class. Silhouette coefficient is a single score that is widely used for measuring the quality of clustering results [69].

Silhouette coefficient measures how well the separation between the clusters independently from the number of clusters. In principle, it measures how each sample in a cluster is close (i.e., similar) to other samples in the same cluster when comparing to other samples in other clusters [70]. The coefficient has a value $\in [-1, 1]$. A high value of the coefficient means a better structure for the clusters. The Silhouette coefficient can be obtained as follows:

$$s(i) = \frac{b(i) - d(i)}{max\{d(i), b(i)\}}, \quad \forall i = 1, \cdots M \tag{26}$$

where M is the total number of samples, and $d(i)$ is the average dissimilarity of the sample $x_i = \{a_{i,1}, \cdots, a_{i,M}\}$ to other samples with the same label y_i and $b(i)$ is the average dissimilarity of the same with respect to the closest cluster with a different label. The average value of $s(i)$ for all samples measures the quality of how well the M samples in the input data are clustered.

C. FAULT AND QUALITY PREDICTION

In a highly imbalanced dataset, predictive models tend to be biased towards the majority class; having a high misclassification rate for the minority class. Such a bias in manufacturing predictive analytics is detrimental to the quality and cost of the end product. Therefore, there is a need for the classifier to provide high accuracy in predicting the minority class (e.g. defects) without deteriorating the classification performance of the majority class [12], [71]. Using a single criterion, such as the overall accuracy or error rate, fails to discern the failed predictions related to defects when the availability of faulty data samples is limited. To this end, more informative evaluation metrics, such as precision and recall, are proposed to capture the effectiveness of the classification method in the presence of imbalanced data.

Precision quantifies the ratio of the correctly predicted faulty samples among all predicted samples in a classification model. Recall is another metric that measures the ratio of the correctly predicted faulty samples among all actual faulty samples in the dataset (regardless of the classification). Precision can be calculated by the following:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{27}$$

where TP represents the number of correctly classified faults, and FP represents the number of faultless samples that have been incorrectly classified as faulty. Recall, on the other hand, is expressed as follows, where FN is the number of faulty samples that are classified as faultless and $TP + FN$ represents the total number of actual faulty samples:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{28}$$

Precision tends to measure how well the fault prediction algorithm can reduce the number of faultless samples that are misclassified – reducing false alarms. The probability of false alarm can also be captured by computing the false positive rate (FPR) as follows:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{29}$$

On the other hand, recall (often referred to as sensitivity or TP rate (TPR)), measures how well the fault prediction model learns to predict the actual faulty samples correctly – reducing the number of undetected faults (i.e., faulty samples that are predicted as faultless samples). However, neither precision nor recall provides a conclusive evaluation of the imbalanced classification model. Good recall value often levels out a reduced precision value and vice-versa. F1-score provides a way to combine the precision and recall (Eq. 27 and Eq. 28). It is also known as F-Measure. It takes into account false alarms and undetected faults – weighting the precision and recall equally. High F1-score means perfect precision and recall scores. F1-score is given by:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{30}$$

Another informative measure is the receiver operating characteristic curve (ROC), which is a graphical plot that illustrates the diagnostic ability of a binary classifier. The ROC curve is created by plotting the recall (in Eq. 28) or TPR against the FPR (in Eq. 29) at various discrimination threshold settings. The discriminating threshold value controls the classification decision based on the probabilistic output of the classifier (Please refer to Section VI-D for more details about the discrimination threshold). Similarly, the precision-recall curve (PRC) is another interpretation of the output of a binary classifier where the precision (in Eq. 27) is plotted against the recall (in Eq. 28) at various potential discrimination threshold settings.

ROC curves are appropriate when the data is relatively balanced between classes – having an equal number of data samples for each class in the dataset. However, with severely imbalanced class distributions, PRC curves tend to be a more informative measure that offers to find an optimal threshold that achieves a right balance between precision and recall, hence considering class distributions. Thus, in the context of manufacturing predictive analytics, we propose to calculate the Precision, Recall, F1-score, and PRC curve in addition to general accuracy and success rate for an in-depth evaluation of the effectiveness of the method adopted.

D. DATA GENERATION

Data-based methods for curtailing the challenges of imbalanced dataset employ different over-sampling generative approaches for the creation of new samples from the minority class. These approaches aim to minimise the distance between the real and generated distributions. Nonetheless, it is crucial to measure the quality of generated data by quantifying how similar generated data and the original data are [57], [72], [73]. To this end, we need to develop a single score to compare the quality of generated samples of different over-sampling methods. Several metrics and statistical tests for measuring the similarity between two distributions exist such as f-divergence (e.g., Hellinger distance, Jensen–Shannon divergence) [73], Wasserstein distance [74] which is also known as Kantorovich–Rubinstein metric and the Kolmogorov–Smirnov test (KS-test) [75], [76], among others.

KS-test is widely used in hypothesis testing for the comparison of the cumulative distribution functions (cdf) of given distributions [77], [78]. Suppose the real and generated data have a size of R and G samples, respectively. Let $F_R(x)$ and $F'_G(x)$ be the cdf of real and generated data distributions, respectively. The KS-statistic is defined as the maximum distance between the two cdfs:

$$D_{MG} = \max_x |F_M(x) - F'_G(x)| \tag{31}$$

where D_{RG} is the maximum absolute difference between the cdfs of the distributions of the real and generated data samples. In principle, the KS-statistic has a value $\in$ [0, 1] that defines the overlap between the two distributions; 0 for perfect overlap and 1 for no overlap. Therefore, the KS-dissimilarity can be thought of as the fractional difference between the two distributions [79]. A null hypothesis $H_0 : F = F'$ is defined to check their overlap; if the distributions of the generated and real faulty data samples are statistically similar. To this end, a q-value is obtained representing the hypothesis probability, taking into account the comparison between D_{RG} and a critical value $C(\alpha)$ such that the null hypothesis is rejected at a level α if:

$$D_{RG} > C(\alpha)\sqrt{\frac{R+G}{R \cdot G}} \tag{32}$$

where $C(\alpha)$ is a size-independent function with α as the chosen significance level for statistical significance. For $q < \alpha$, the null hypothesis is rejected. This means the distribution of the generated data doesn't converge to the real faulty data samples, and they aren't statistically similar. In such a case, the generative method generates poor quality data samples. In contrast, for $q > \alpha$, the hypothesis is accepted, and this implies a high quality of generated data samples. In principle, the selected significance value of α impacts recognising the statistical significance between the two distributions.

For a small α value, a substantial difference between the two distributions is required for rejecting the null hypothesis, indicating a higher D_{RG} value. On the other hand, a significantly large α means that having small differences between the two distributions are magnified – leading the null hypothesis to be rejected regardless of small D_{RG} values.

VI. FRAMEWORK EVALUATION

As introduced in Section I, the proposed framework is not tailored for a particular dataset but is designed to address the challenge of data imbalance that results from an IoT-rich (smart) manufacturing environment. This is purposely a large domain and encompasses smart environments such as industrial or mechanical systems. Indeed, our main contribution is to present such a framework that is not tailored to a particular dataset but is able to extract the pertinent characteristics of any given dataset to guide the user in selecting the appropriate ML approach. This section explains the real-world dataset used in our evaluation which is based on heavy trucks' air pressure system. The details of the pre-processing phase, the classifiers parameterisation, and post-processing steps are also presented. The evaluation framework discussed in Section V is then used to assess the effectiveness of each method.

A. AIR PRESSURE FAILURE DATA

The Air Pressure System (APS) plays a vital role in heavy Scania trucks. In principle, APS is a system that generates pressurised air to be used in various functions in a truck, such as gear changes and braking. The APS Scania dataset was collected from heavy Scania trucks and was made available by the Industrial Challenge for IDA 2016.[4] Each instance in the dataset is classified as positive or negative. The APS is, thus, an example of smart mechanical systems and fits for the purpose of framework validation since it has a high degree of bias. Although the data is collected from trucks in operation, the insights drawn from analysing this data reflect on the quality of the manufacturing process and highlight manufacturing issues when linked to other datasets such as factory (e.g. factory Identification) and batch numbers.

The positive class indicates that reported failures are related to the APS system, while the negative class includes all other instances that have other types of faults. Differently from the common binary classification in manufacturing datasets (faulty and faultless), in this case, both classes reflect faults. Nonetheless, the positive class remains the less frequent type of APS-related faults, hence represents the minority class. Whereas, non-APS related faults are many and form the majority class, i.e. the positive class.

The main goal is to develop a binary predictive model that correctly identifies failures related to APS. An APS failure that is not predicted prior to its occurrence would incur a drastic cost on Scania. Thus the predictive model is expected to perform well in terms of accuracy in general but also in detecting false alarms (misclassifying other failures as APS) or missed alarms (i.e., misclassifying an APS failure as that of another component).

Scania has provided training and testing datasets for APS. The training set contains 60, 000 rows, of which only 1, 000 instances belong to the positive class (i.e., faulty samples that are related to APS) and 59, 000 for the negative class (i.e., faulty samples that are not associated with APS). Each row includes 171 anonymised features, one of which is the label (i.e., target class) column to indicate APS-related faults (i.e., positive class) or other faults (i.e., negative class). The testing set consists of 16, 000 instances, of which only 375 instances belong to the positive class.

[4] https://ida2016.blogs.dsv.su.se/?page_id=1387

The APS dataset is therefore biased with a high imbalance ratio between positive and negative classes [13], [80]. Indeed, 1.7% of the entire training set represents APS-related faulty samples (i.e., positive samples), while the remaining 98.3% belongs to other faulty samples (i.e., negative samples). It is expected that, with such imbalanced data, the predictive models would tend to be biased towards the majority class (i.e., non-APS faulty samples) and less sensitive towards the minority class (i.e., APS-related faulty samples) [18].

In addition to the dataset, Scania has provided a misclassification cost metric for the APS dataset. As discussed previously in Section III-B, the cost of predicting false negatives (C_{FN}) is much higher than false positives, i.e. false alarms (C_{FP}). In Scania APS scenario, the cost of the former is C_{FN} = €500, while the latter is C_{FP} = €10 [80]. In principle, C_{FP} refers to the cost of unnecessary checks carried by a mechanic as a result of false alarms. On the other hand, C_{FN} refers to undetected APS-related faults which may cause the truck to break down; hence, it is the incurred cost of downtime and repair. By substituting these cost values in Eq. 2, the total cost that should be minimised for predicting APS-related faulty samples is expressed as follows:

$$\begin{aligned} T_{Cost} &= m\, C_{FP} + n\, C_{FN} \\ &= m \times 10 + n \times 500 \end{aligned} \tag{33}$$

where n is the number of undetected APS-related faulty samples (missed alarms), and m is the number of false alarms.

The Scania APS dataset is suitable for the evaluation of the data-driven methods introduced in Section III-A for dealing with the imbalanced data problem in manufacturing. Furthermore, the manufacturer's cost-sensitive function gives a domain expert's perception of the impact of different misclassification errors on the manufacturing process. It can, therefore, be implemented as an effective algorithmic-based approach (as discussed in Section III-B) as well as hybrid approaches that combine both data-driven and algorithm-based methods (as discussed in Section III-C).

B. DATA PRE-PROCESSING

As shown in Figure 1, data pre-processing includes essential and optional procedures. Data cleaning is essential to any classification exercise and aims to compensate for missing values, among other issues. The APS dataset contains up to 82% of missing values per feature. We have replaced the missing values by the mean imputation method as in [80]. In such a method, all missing values in a particular column in the dataset are substituted with the mean value of the available values in that column.

The pre-processing phase also includes the optional procedures for reducing the number of features of a multi-dimensional dataset to facilitate its interpretation. The APS dataset contains 171 features, one of which is the label class (i.e., to indicate if the faulty sample is APS-related or other). To reduce the number of features (i.e., dimensions), we use Principal Component Analysis (PCA). PCA simplifies the complexity of high-dimensional data by geometrically projecting it into fewer dimensions that maximise the variance, called Principal Components (PCs). The primary goal of PCA is to obtain the best summary of the data using a limited number of PCs [81]. In this process, we found that 11 principal components are sufficient to capture 95% of the information in the dataset – having a cumulative explained variance percentage of 95%. The number of principal components was found to be a good indication to hit the point of diminishing returns (i.e., a little variance is gained by retaining additional principal components). We have included more details about the steps that yielded the number 11 in Appendix A.

Data augmentation is another optional pre-processing procedure, as discussed in Section III-A. We describe the methodology that we adopted with the APS dataset for different data augmentation techniques in Section IV. In the following paragraph, we elaborate on the selection of data augmentation methods used in the hybrid combinations of this study.

C. PREDICTIVE MODELS WITH APS IMBALANCED DATA

We have developed three different machine learning predictive models to detect APS-related faults (i.e., positive or minority class) or Non-APS-related faults (i.e., negative or majority class). The three models are Logistic Regression (LR) [82], Random Forest (RF) [83] and XGBoost [84]. These models are represented as *ML Classifiers* in Figure 1. We have used these binary classification models to make a fair empirical comparison between the different counterbalancing techniques explained in Section. IV). Given that the original APS dataset is highly imbalanced, it calls for procedures to generate synthetic APS-related failure data samples for the *Ready-to-use* training set, as shown in Figure 1. We have conducted a set of experiments that can be grouped into the following five cases. Each of these cases represents a group of experiments with common pre-processing methods but use different classifiers. The performance metrics for these experiments are reported based on their total cost in addition to their achieved accuracy, precision, recall, and F1-score (discussed in Section. V-C).

- Case I: evaluating the classifier algorithms using the original training data without data augmentation. Principal Component Analysis (PCA) is employed in the pre-processing phase to simplify the complexity of our high-dimensional data.
- Case II: using SMOTE to generate synthetic samples of APS-related faulty samples such that the original training dataset is augmented with new synthetic minority samples. PCA is also applied in the pre-processing phase, and each experiment is repeated for different imbalance ratios (explained in Section. V-A).
- Case III: using GAN and WGAN for data augmentation. PCA is also applied in the pre-processing phase, and each experiment is repeated for different imbalance ratios. Moreover, the quality of generated samples is

evaluated using the performance metric discussed in Section. V-D.

- Case IV: clustering the APS-faulty samples; clusters are reported based on the metric discussed in Section. V-B to measure the separability between clusters. More precisely, we measure the distance between each data sample, the centroid of its assigned cluster and the closest centroid belonging to another cluster. We then use CGAN and WCGAN to generate synthetic samples conditioned on the cluster assigned to minority data samples. For these set of experiments, we also apply PCA, and we calculate the same performance metrics as in Case III for various imbalanced ratios.

It should be noted that the four cases mentioned here aim to minimise the total cost of misclassification. To this end, we tune the hyperparameters of the classification algorithm using 5-fold cross-validation in such a way that the total cost of misclassification of APS-related faulty samples (in Eq. 33) is minimised. In other words, each of the described four cases implements algorithm-based methods to curtail the pitfalls of imbalanced datasets. Case I falls in the category of cost-sensitive approaches (discussed in Section. III-B). Indeed, all three classifier algorithms (LR, RF, and XGBoost) in Case I are trained to minimise the cost matrix for binary misclassification without modifying the original training dataset. On the other hand, all other cases are considered hybrid methods (discussed in Section III-C). In each of these cases, the original training dataset is augmented by generating synthetic APS-related faulty samples (the number of minority class samples is increased to compare to the majority class). At the same time, the three classifiers are trained, and their parameters are tuned for minimising a misclassification cost matrix. Thus, Cases II, III, and IV combine data-driven and algorithmic-based approaches, hence fall under the hybrid category.

D. THRESHOLD FOR IMBALANCED CLASSIFICATION

The RF, LR and XGBoost classifiers output a probability that estimates the likelihood of associating a class label to each of the data samples in the testing dataset. This probability gives some confidence in the label prediction. In principle, the output probability is then converted to a discrete class label in the post-processing phase (see Figure 1). In a binary classifier, this is achieved by using a threshold that is referred to as *a decision threshold*, *a discrimination threshold* or a *cut-off*. The default threshold value is often set to 50% or 0.5 when the dataset is balanced, as shown in Figure 4. However, a decision threshold of 0.5 may not provide an optimal interpretation of predicted probabilities in the case of imbalanced datasets. To this end, the decision threshold is moved along the x-axis in Figure 4 to compensate for the data imbalance and improve the prediction results. This is referred to as *threshold-moving method* in the post-processing phase of classifiers [85].

The decision threshold is selected with the aim of minimising the probabilities of FP and FN, as shown in Figure 4.

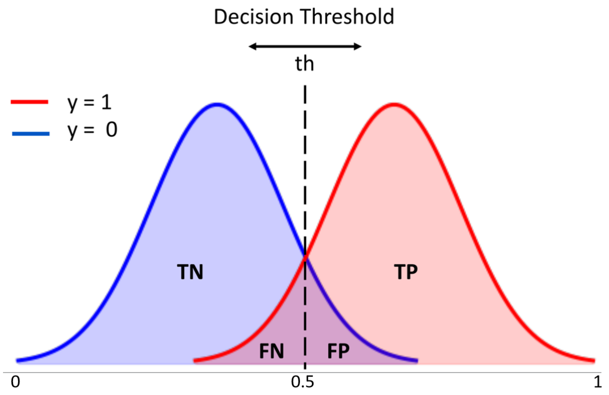


FIGURE 4. Decision boundary to find the optimal threshold *th*
y = 1 for APS faulty samples; y = 0 for Non-APS samples
TP: True Positive, FP: False Positive (i.e., false alarm), TN: True Negative, and FN: False Negative.

Referring to our problem formulation in Section II, label $y = 1$ represent the minority class whereas $y = 0$ refers to the majority class. Then for a sample x, when a classifier outputs a probability $p(y = 1|x) >= th$, it is classified as positive sample, otherwise as a negative sample. The threshold-moving method aims to adjust the value of the decision threshold in order to improve the predictions. Given a threshold centered at the value $th = 0.5$, suppose that a data sample x_1 has a probability $p(y = 1|x_1) = 0.6$ such as $p(y = 1|x_1) > p(y = 0|x_1)$; it is thus classified as belonging to the minority class. Another data sample x_2 with $p(y = 0|x_2) = 0.8$ and $p(y = 0|x_2) > p(y = 1|x_2)$ is classified as belonging to them majority class. If, however, the threshold is moved to $th = 0.85$ or 85%, then x_2 will be assigned to the minority class instead of majority because $p(y = 0|x_2) < th$. On the other hand, With $th = 0.85$, x_1 would remained in the minority class since $p(y = 1|x_1) < th$.

According to the threshold moving method, a sample is only classified as belonging to the majority class if the classifier's confidence in this classification is higher than the set threshold. In our dataset, this implies that a sample is considered to be APS-relate fault unless the classifier is highly confident of the fault not relating to APS. As discussed in Section V-C, the optimum value to the decision threshold for imbalanced dataset is derived using the *precision-recall-curve* (PCR).

In [13], the moving threshold approach is associated with the confidence level of a given sample belonging to the majority class. The best results of predicting APS faulty samples were obtained based on using a decision threshold of $th = 95\%$. In this moving threshold application, the selected threshold determines whether an instance belongs to a majority class (i.e., non-APS-related faulty samples or negative samples). In principle, a data sample is only classified as a non-APS fault if the related confidence level exceeds the particular threshold (i.e., $\geq 95\%$); otherwise, it is classified as an APS related fault.

TABLE 2. Case I: performance of classifiers on actual data (PCs = 11) without augmenting synthetic faulty data samples.

Classifier	Accuracy	th	Precision	Recall	F1-score	No. of FN	No. of FP	Total cost
LR	94%	65.5%	0.3	0.97	0.4	13	1005	€16,550
RF	93%	70%	0.2	0.99	0.4	5	1125	€13,750
XGBoost	95%	98.2%	0.3	0.97	**0.5**	9	773	**€12,230**

Following the same work on APS dataset in [13], [80], [86], we incorporate a threshold moving method as a post-processing phase of classifiers in our conducted experiments. In such a case, the precision-recall curve (PCR explained previously in Section. V-C) is utilised to obtain the list of the potential threshold values. The value of this threshold is optimised whilst minimising the total misclassification cost. In such a case, the best threshold is selected that achieves the minimum cost.

E. PARAMETER SETTINGS AND REPRODUCIBILITY

We have developed and evaluated the classification algorithms and the experiments explained above in Python. The scikit-learn (sklearn) library provides the main implementations of these algorithms [87]. In sklearn, we are able to incorporate class-specific weights in the loss function of LR and RF algorithms. Similar to [13], the weight of each class is automatically adjusted to be inversely proportional to class frequencies. SMOTE is also implemented and available in imbalanced-learn API.[5] In each experiment, we only mention the parameter settings if they are different from the default values in sklearn and imbalanced-learn API. On the other hand, our GAN-based approaches are adapted from the available open-source GAN-Sandbox.[6] To ensure the reproducibility of our results, we have made the code and dataset of our implementations available and have also provided details of a configurable experimental set-up at: https://github.com/YasminFathy/HandleImbalancedDatasets

VII. RESULTS AND DISCUSSION

In this section, we discuss and analyse the results of our four sets of experiments discussed in Section. VI-C.

A. CASE I

Table 2 shows the results of Case I where LR, RF and XGBoost are trained on the original data and tested on the testing set. As mentioned previously, we apply PCA in the pre-processing phase and the moving threshold method during the post-processing step as detailed in Section VI-D. The reported results in Table 2 are derived after finding the optimal decision threshold empirically such that a minimum total cost is achieved. The threshold values for LR, RF and XGBoost are 65.5%, 70% and 98.2%, respectively. Each threshold value was obtained based on optimising the minimum cost while aiming to achieve a trade-off between precision and recall through using the precision-recall curve (PRC) (as discussed in Section. V-C). In Case I experiments, XGBoost achieves the best results with a total cost of €12,230 as highlighted in Table. 2. Our results are aligned with the work in [49] where a boosting-based method achieves less total cost than RF.

The results obtained in Case I show that the accuracy is not an informative metric to measure the overall performance of an imbalanced binary classification. As shown in Table 2, LR achieves 94% accuracy which is equivalent to only 1% deterioration compared to XGboost results of 95%. Furthermore, the results of both classifiers with respect to *Recall* and *Precision* are exactly the same. However, examining the total cost achieved by each classifier reveals that XGBoost indeed outperforms LR by a significant margin of 26%. On the other hand, in comparison with RF, LR has better accuracy by 1% and exactly the same F1-score. However, the cost reduction achieved by RF in comparison with LR is also a significant margin of 17%. These results further demonstrate the discussion around context-aware performance measurement presented in Section. V. In fact, using a single criterion such as precision, recall, F1-score or accuracy metrics fails to discern the critical performance of a classifier in the presence of imbalanced datasets.

B. CASE II

Table 3 shows the results of Case II where LR, RF and XGBoost are trained on the *Ready-to-use* training data to which we have applied PCA and data augmentation (for the minority class only) using SMOTE. The original training dataset has an imbalance ratio of $IR = 0.1$ (Eq. 25) which corresponds to 1000 minority samples over 59000 majority samples. Using SMOTE for data augmentation, we have a total number of 2000, 5000, $10,000$ minority samples for IR equal to 0.2, 0.4, and 0.6, respectively. We do not alter the number of majority samples and do not alter the testing dataset, as shown in Figure 1.

The results shown in Table 3 include those obtained in Case I (reported in Table 2 and are aligned in terms of classifier ranking. The best performance of each classifier for different IR settings is highlighted in bold. As can be seen, XGBoost outperforms RF, which is followed by LR. However, it is worth noting that despite the three data augmentation levels (IR={0.2, 0.4, 0.6}), the performance gain of the classifiers remains limited. LR and RF achieve the minimal cost reduction of 1.6% and 2.8%, respectively, whereas XGBoost does not benefit from data augmentation. XGBoost fails to reduce the number of false positives (i.e., false alarms) as more synthetic faulty data samples are

[5] https://imbalanced-learn.readthedocs.io/en/stable/api.html
[6] https://github.com/mjdietzx/GAN-Sandbox

TABLE 3. Case II: performance of classifiers with augmenting artificial faulty data samples obtained using SMOTE to original data (with PCs=11).

Classifier	Metric	IR = 0.1 (original data)	IR = 0.2	IR = 0.4	IR = 0.6
LR	Accuracy	94%	93.4%	93%	94%
	Precision	0.3	0.3	0.2	0.3
	Recall	0.97	0.97	0.97	0.96
	F1-score	0.4	0.4	0.4	0.4
	No. of FN	13	12	11	15
	No. of FP	1005	1029	1167	899
	Total cost	€16,550	**€16,290**	€17,170	€16,490
	th	65.5%	67.5%	71%	62.9%
RF	Accuracy	93%	94%	94%	94%
	Precision	0.2	0.3	0.3	0.3
	Recall	0.99	0.98	0.98	0.98
	F1-score	0.4	0.4	0.4	0.4
	No. of FN	5	8	8	9
	No. of FP	1125	962	997	887
	Total cost	€13,750	€13,620	€13,970	**€13,370**
	th	70%	60%	59.6%	54.4%
XGBoost	Accuracy	95%	94%	95%	95%
	Precision	0.3	0.3	0.3	0.3
	Recall	0.97	0.98	0.97	0.97
	F1-score	**0.5**	0.4	0.5	0.5
	No. of FN	9	8	12	10
	No. of FP	773	914	779	793
	Total cost	**€12,230**	€13,140	€13,790	€12,930
	th	98.2%	97.4%	92.3%	88.3%

generated by SMOTE. Unlike, LR and RF that incorporate class-specific weights in the loss function, XGBoost equally weights misclassified samples, and its performance often becomes subtle with imbalanced data, and it has to be combined with other ensembling methods to improve imbalanced classification [88].

C. CASE III

Table 4 shows the results of Case III where LR, RF and XGBoost are trained on the *Ready-to-use* training data to which we have applied PCA and data augmentation (for the minority class only) using GAN. The same approach to data augmentation described in Section VII-B with respect to IR={0.2, 0.4, 0.6} and the testing data is adopted, except that GAN is used instead of SMOTE. The real and generated distributions tend to be similar; this is measured using KS-test (discussed in Section. V-D). The same ranking between the three classifiers is maintained as in Cases I and II. Moreover, the same trend seen in Case II, whereby LR performs best with $IR = 0.2$ and RF performs best for $IR = 0.6$. Although XGBoost still does not benefit from the data augmentation, RF and LR achieve better cost reduction than SMOTE with 8.9% and 8.7% improvement compared Case I. Similar experiments have been conducted by WGAN for data augmentation.

GAN is a generative model that produces new content based on the presented training data; however, generated data might be noisy [31]. For example, in [31], GAN improved the classification when compared with the original imbalanced dataset; however, it did not perform better than SMOTE. Our experiments show different behaviour when comparing the results obtained in Case I, Case II, and Case III with both LR and RF classifiers. In both classifiers, SMOTE presents gain compared to Case I and GAN brings a larger gain compared to SMOTE. Indeed, we argue that the benefit of data augmentation and the superiority of GANs over SMOTE essentially depend on the underlying characteristic of the minority class and the data complexity.

Table 5 shows the results of Case III using WGAN with IR={0.2, 0.4, 0.6}. The same ranking between the three classifiers is maintained as in all previous cases. However, we see a degradation in performance compared to the results achieved by GAN (Table 4). With WGAN, the cost reduction achieved by LR and RF shows a limited improvement of 4.0% and 5.1% compared Case I.

Analogous to GAN, WGAN tends to converge faster and being stable during the training. However, WGAN has no substantial effect on reducing the total classification cost in our experiments.

D. CASE IV

Table 6 and Table 7 show the results obtained by CGAN and WCGAN, respectively for each of the classifiers and IR ratios. It is clear that generating synthetic data conditionally on the class label did not improve the result in our experiment. This outcome is also related to the underlying characteristics of the dataset. When calculating the Silouhette coefficient (SC) (discussed in Section. V-B) which measured the quality of clusters in the APS dataset, we find $SC = 0.4$. This is obtained by clustering the APS faulty data samples

TABLE 4. Case III: performance of classifiers with augmenting artificial faulty data samples obtained using GAN to original data (with PCs=11).

Classifier	Metric	IR = 0.1 (original data)	IR = 0.2	IR = 0.4	IR = 0.6
LR	Accuracy	94%	94%	93%	93%
	Precision	0.3	0.3	0.3	0.3
	Recall	0.97	0.97	0.97	0.98
	F1-score	0.4	0.4	0.4	0.4
	No. of FN	13	10	10	9
	No. of FP	1005	1011	1040	1086
	Total cost	€16,550	**€15,110**	€15,400	€15,360
	th	65.5%	44.6%	36.6%	37%
RF	Accuracy	93%	94%	95%	94%
	Precision	0.2	0.3	0.3	0.3
	Recall	0.99	0.98	0.98	0.98
	F1-score	0.4	0.4	0.5	0.4
	No. of FN	5	8	9	6
	No. of FP	1125	990	819	952
	Total cost	€13,750	€13,900	€12,690	**€12,520**
	th	70%	51%	80.6%	89.8%
XGBoost	Accuracy	95%	94%	95%	92%
	Precision	0.3	0.3	0.3	0.3
	Recall	0.97	0.98	0.98	0.98
	F1-score	0.5	0.4	0.5	0.4
	No. of FN	9	8	9	5
	No. of FP	773	940	860	1000
	Total cost	**€12,230**	€13,400	€13,100	€12,500
	th	98.2%	98.2%	97.6%	82.9%

TABLE 5. Case III: performance of classifiers with augmenting artificial faulty data samples obtained using WGAN to original data (with PCs=11).

Classifier	Metric	IR = 0.1 (original data)	IR = 0.2	IR = 0.4	IR = 0.6
LR	Accuracy	94%	93%	92%	94%
	Precision	0.3	0.3	0.2	0.3
	Recall	0.97	0.97	0.97	0.94
	F1-score	0.4	0.4	0.4	0.4
	No. of FN	13	10	10	23
	No. of FP	1005	1088	1294	1002
	Total cost	€16,550	**€15,880**	€17,940	€21,520
	th	65.5%	47.2%	46.2%	33.1%
RF	Accuracy	93%	93%	94%	93%
	Precision	0.2	0.2	0.3	0.3
	Recall	0.99	0.98	0.98	0.99
	F1-score	0.4	0.4	0.4	0.4
	No. of FN	5	8	8	1084
	No. of FP	1125	1123	891	5
	Total cost	€13,750	€15,230	**€12,910**	€13,340
	th	70%	54.5%	81.5%	90.5%
XGBoost	Accuracy	95%	94%	93%	93%
	Precision	0.3	0.3	0.3	0.2
	Recall	0.97	0.98	0.98	0.99
	F1-score	0.5	0.4	0.4	0.4
	No. of FN	9	8	6	5
	No. of FP	773	926	1075	1112
	Total cost	**€12,230**	€13,260	€13,750	€13,620
	th	98.2%	98.1%	98.8%	98.9%

using hierarchical clustering (i.e., agglomerative clustering). Having $SC = 0.4$ means that it is quite hard to separate APS-related fault samples into clusters. For that reason, conditioning GAN on the cluster label does not bring any advantage. This outcome further proves our claim in Section V that there is no one-solution-fits-all when it comes to binary classification. Moreover, it is essential to examine the dataset at hand and understand its context to allow a pertinent choice of classification methods that would prove effective.

TABLE 6. Case IV: performance of classifiers with augmenting artificial faulty data samples obtained using CGAN to original data (with PCs=11).

Classifier	Metric	IR = 0.1 (original data)	IR = 0.2	IR = 0.4	IR = 0.6
LR	Accuracy	94%	95%	93%	92%
	Precision	0.3	0.3	0.2	0.2
	Recall	0.97	0.97	0.98	0.98
	F1-score	0.4	0.5	0.4	0.4
	No. of FN	13	14	9	6
	No. of FP	1005	843	1095	1262
	Total cost	€16,550	**€15,430**	€15,450	€15,620
	th	65.5%	37.3%	42.5%	48.5%
RF	Accuracy	93%	93%	94%	93%
	Precision	0.2	0.3	0.3	0.3
	Recall	0.99	0.98	0.98	0.97
	F1-score	0.4	0.4	0.5	0.4
	No. of FN	5	8	7	10
	No. of FP	1125	1046	932	1048
	Total cost	€13,750	€14,460	**€12,820**	€15,480
	th	70%	54.8%	85.8%	81.2%
XGBoost	Accuracy	95%	94%	91%	94%
	Precision	0.3	0.3	0.2	0.3
	Recall	0.97	0.98	0.97	0.98
	F1-score	0.5	0.4	0.3	0.4
	No. of FN	9	8	10	9
	No. of FP	773	905	1479	900
	Total cost	**€12,230**	€13,050	€19,790	€13,500
	th	98.2%	97.9%	68.9%	97.8%

TABLE 7. Case IV: performance of classifiers with augmenting artificial faulty data samples obtained using WCGAN to original data (with PCs=11).

Classifier	Metric	IR = 0.1 (original data)	IR = 0.2	IR = 0.4	IR = 0.6
LR	Accuracy	94%	0.94%	93%	94%
	Precision	0.3	0.3	0.2	0.3
	Recall	0.97	0.97	0.97	0.96
	F1-score	0.4	0.4	0.4	0.4
	No. of FN	13	13	10	16
	No. of FP	1005	897	1140	910
	Total cost	€16,550	**€15,470**	€16,400	€17,100
	th	65.5%	38.5%	41.8%	27.9%
RF	Accuracy	93%	93%	94%	93%
	Precision	0.2	0.3	0.3	0.3
	Recall	0.99	0.98	0.98	0.98
	F1-score	0.4	0.4	0.4	0.4
	No. of FN	5	8	7	8
	No. of FP	1125	1034	932	1092
	Total cost	€13,750	€14,340	**€12,820**	€14,920
	th	70%	52.9%	82.4%	83.3%
XGBoost	Accuracy	95%	94%	95%	94%
	Precision	0.3	0.3	0.3	0.3
	Recall	0.97	0.99	0.97	0.98
	F1-score	0.5	0.4	0.5	0.5
	No. of FN	9	5	10	9
	No. of FP	773	1035	809	883
	Total cost	**€12,230**	€12,850	€13,090	€13,330
	th	98.2%	98.5%	97%	97.5%

Although WCGAN (in Table 7) shows a slight improvement to WGAN (in Table. 5) by LR and RF achieving 6.5% and 6.8% gain compared to Case I, the benefit of conditional training remains limited due to the low Silouhette factor as explained in Section VII-C.

This is another demonstration of the necessity to understand the intrinsic characteristics of the data before selecting the data-augmentation method. With the APS dataset, using the Wasserstein distance between two probability distributions in the training process instead of the Kullback–Leibler divergence does not improve the quality of the generated synthetic samples. This is clear from the comparison of the results in Table 4 and those in Table 5 and relates to the Silouhette Coefficient of the dataset. Further investigation is left for future work.

VIII. LESSONS LEARNT AND CONCLUSIVE REMARKS

Industry 4.0 offers manufacturing the potential of leveraging IoT data with machine learning to reap the benefits of automation and excellence in quality control. A major obstacle currently facing the progress in this field is the nature of data generated by manufacturing processes. Indeed, manufacturing data is overwhelmed with instances of good performance with few examples of malfunctioning, referred to as imbalanced data. Since machine learning is a data-driven learning process, imbalanced data results in biased learning. In the context of manufacturing, this learning bias causes most manufacturing faults to go unnoticed and compromising the quality of the end product. The results of our study are expected to improve the decision-making in smart manufacturing by detecting unexpected faults that affect products' quality.

In this work, we have presented the first comprehensive comparative analysis of various methods in the literature that aim to curtail the curse of data imbalance in the ML process. We present an evaluation framework that considers all steps of the process, including data preparation and pre-processing, classifier design, and post-processing. More importantly, we set-up a set of key performance indicators that jointly reveal the effectiveness of each method in a context-aware fashion.

We have applied our framework on an industry-based dataset which enabled us to conduct the comparative analysis and draw key insights with particular application to smart manufacturing. We summarise key lessons learnt from this study that we hope will be a useful guide to future research in this field:

- All the experiments conducted in each of the cases in this study have been assessed using the proposed evaluation framework. The overarching insight that can be drawn from these results is that none of the key metrics, such as accuracy or F1-score, can be misleading when examined in isolation. As such, it is essential to inspect the full spectrum of performance metrics to have a complete evaluation of the ML techniques used in manufacturing.
- In binary classification, there is no one-solution-fits-all as the optimum solution depends on the intrinsic characteristics of the dataset and the context in which data is collected, and classification results are employed. Here are a few insights into the role of data and context in the effectiveness of ML methods:
    - Our experiments demonstrate that XGBoost, empowered with a cost-sensitive function and context-aware moving threshold, outperforms Logistic Regression and Random Forest classifiers in every setting, even without data augmentation. This shows that a domain expert's input into the interpretation of the data is critical in the success of ML algorithms.
    - The similarity between real and generated distributions is quantified using KS-test to measure the quality of generated samples. The number of generated synthetic samples does not have a linear relation with the performance of the ML algorithm. In fact, some classifiers perform better with a larger synthetic dataset, and others do the opposite as there is a risk of unsuitable augmentation method generating noise instead of useful data samples. This is an effect of the interplay between the underlying data features, the data augmentation method, and the intrinsic method implemented in the classifier.
    - It is generally believed that conditional GANs (e.g., CGAN and WCGAN) tend to improve the classification performance in comparison with ordinary GANs (e.g., GAN and WGAN). However, our experiments have demonstrated that this is only true if applied to a compatible dataset. The APS dataset used in this work has a low Silhouette coefficient (i.e., a poor distinction between both classes); hence, conditioning the training process on the class label adds little benefit to the end results. In other words, the APS dataset is not compatible with conditional GAN derivatives when conditioned on the class label.
    - Augmenting artificial faulty data samples obtained by GAN achieves a larger gain compared to SMOTE (using logistic regression). More precisely, GAN achieves up to 9% cost reduction compared to the original data (with no augmentation) in Case III and 7% compared to SMOTE in Case II. On the other hand, SMOTE improves the predictive analytics (using random forest) and offers higher gain comparing with the original data (with no augmentation) with a cost reduction up to 3% in Case II
- Overall, it is not always practical to compare our results with existing similar studies. Most of the studies including, [49], [86] do not report the parameter settings of each chosen classifier which makes it a challenging task to reproduce their experiments. For instance, authors in [49] achieve a total cost of €11,090 using RF as a binary classifier on the original data; however, they do not report parameter settings, imputation technique, whether a decision threshold is applied and its value or other evaluation metrics including precision and recall.

APPENDIX

PRINCIPAL COMPONENTS

PCA finds a projection of high dimensional data into a lower-dimensional subspace such that the maximum variance of the data is retained and the least-square reconstruction error is

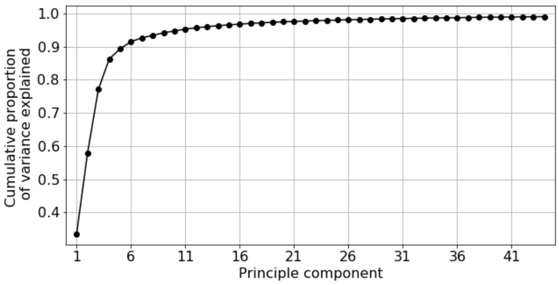


FIGURE 5. **The cumulative variance explained by different number of principal components.**

minimised. Figure. 5 shows the cumulative variance retained by a different number of principal components. We found that 11 principal components are sufficient to capture 95% of the variance in the dataset where a little variance of data is gained by retaining additional principal components (i.e., 11 components seem to be a good indication to hit the point of diminishing returns).

REFERENCES

[1] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 157–169, Jul. 2018.

[2] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using IoT sensor data," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Nov. 2017, pp. 87–90.

[3] M. Syafrudin, G. Alfian, N. Fitriyani, and J. Rhee, "Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing," *Sensors*, vol. 18, no. 9, p. 2946, Sep. 2018, doi: 10.3390/s18092946.

[4] J.-H. Oh, J. Y. Hong, and J.-G. Baek, "Oversampling method using outlier detectable generative adversarial network," *Expert Syst. Appl.*, vol. 133, pp. 1–8, Nov. 2019.

[5] B. Song, X. Zhou, H. Shi, and Y. Tao, "Performance-indicator-oriented concurrent subspace process monitoring method," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5535–5545, Jul. 2019.

[6] B. Song, H. Yan, H. Shi, and S. Tan, "Multisubspace elastic network for multimode quality-related process monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5874–5883, Sep. 2020.

[7] B. Song, H. Shi, S. Tan, and Y. Tao, "Multi-subspace orthogonal canonical correlation analysis for quality related plant wide process monitoring," *IEEE Trans. Ind. Informat.*, early access, Aug. 7, 2020, doi: 10.1109/TII.2020.3015034.

[8] Y. Fathy, P. Barnaghi, and R. Tafazolli, "Large-scale indexing, discovery, and ranking for the Internet of Things (IoT)," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1–53, Jun. 2018.

[9] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[10] G. Wang, A. Ledwoch, R. M. Hasani, R. Grosu, and A. Brintrup, "A generative neural network model for the quality prediction of work in progress products," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105683.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[12] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[13] C. F. Costa and M. A. Nascimento, "Ida 2016 industrial challenge: Using machine learning for predicting failures," in *Proc. Int. Symp. Intell. Data Anal.* Cham, Switzerland: Springer, 2016, pp. 381–386.

[14] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, Jul. 2018.

[15] G. M. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 7–19, Jun. 2004.

[16] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proc. 24th Int. Conf. Mach. Learn. ICML*, 2007, pp. 935–942.

[17] C. K. Aridas, S. Karlos, V. G. Kanas, N. Fazakis, and S. B. Kotsiantis, "Uncertainty based under-sampling for learning naive bayes classifiers under imbalanced data sets," *IEEE Access*, vol. 8, pp. 2122–2133, 2020.

[18] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, p. 27, Dec. 2019.

[19] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci.*, vols. 409–410, pp. 17–26, Oct. 2017.

[20] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. Workshop Learn. Imbalanced Datasets*, vol. 126, Aug. 2003, pp. 1–7.

[21] Y. Hou, B. Li, L. Li, and J. Liu, "A density-based under-sampling algorithm for imbalance classification," *J. Phys., Conf. Ser.*, vol. 1302, Aug. 2019, Art. no. 022064.

[22] F. Kamalov, "Kernel density estimation based sampling for imbalanced class distribution," *Inf. Sci.*, vol. 512, pp. 1192–1201, Feb. 2020.

[23] A. Fernández, S. del Río, N. V. Chawla, and F. Herrera, "An insight into imbalanced big data classification: Outcomes and challenges," *Complex Intell. Syst.*, vol. 3, no. 2, pp. 105–120, Jun. 2017.

[24] D.-H. Lee, J.-K. Yang, C.-H. Lee, and K.-J. Kim, "A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data," *J. Manuf. Syst.*, vol. 52, pp. 146–156, Jul. 2019.

[25] K. Yang, Z. Yu, X. Wen, W. Cao, C. L. Philip Chen, H.-S. Wong, and J. You, "Hybrid classifier ensemble for imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1387–1400, Apr. 2020.

[26] Y. O. Lee, J. Jo, and J. Hwang, "Application of deep neural network and generative adversarial network to industrial maintenance: A case study of induction motor fault detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 3248–3253.

[27] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, 2009, pp. 475–482.

[28] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2005, pp. 878–887.

[29] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[31] F. H. K. dos Santos Tanaka and C. Aranha, "Data augmentation using GANs," 2019, *arXiv:1904.09135*. [Online]. Available: http://arxiv.org/abs/1904.09135

[32] G. D. Ranasinghe and A. Kumar Parlikad, "Generating real-valued failure data for prognostics under the conditions of limited data availability," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2019, pp. 1–8.

[33] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 17, no. 1, 2001, pp. 973–978.

[34] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *Int. J. Advance Soft Compu. Appl*, vol. 7, no. 3, pp. 176–204, 2015.

[35] Y. Hu, C. Guo, E. W. T. Ngai, M. Liu, and S. Chen, "A scalable intelligent non-content-based spam-filtering framework," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8557–8565, Dec. 2010.

[36] C. X. Ling and V. S. Sheng, "Cost-sensitive learning and the class imbalance problem," *Encyclopedia Mach. Learn.*, vol. 2011, pp. 231–235, Jan. 2008.

[37] B. Krawczyk, M. Woźniak, and F. Herrera, "On the usefulness of one-class classifier ensembles for decomposition of multi-class problems," *Pattern Recognit.*, vol. 48, no. 12, pp. 3969–3982, Dec. 2015.

[38] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, 2010.

[39] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[40] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[41] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, pp. 1189–1232, Oct. 2001.

[42] L. I. Kuncheva, *Combining Pattern Classifiers: Methods Algorithms*. Hoboken, NJ, USA: Wiley, 2014.

[43] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Inf. Fusion*, vol. 16, pp. 3–17, Mar. 2014.

[44] Q. Cao and S. Wang, "Applying over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning," in *Proc. Int. Conf. Inf. Manage., Innov. Manage. Ind. Eng.*, Nov. 2011, pp. 543–548.

[45] S. Wang, Z. Li, W. Chao, and Q. Cao, "Applying adaptive over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–8.

[46] C. Drummond and R. C. Holte, "C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *Proc. ICML Workshop Learn. Imbalanced Datasets*, vol. 11, 2003, pp. 1–8.

[47] F. Li, X. Zhang, X. Zhang, C. Du, Y. Xu, and Y.-C. Tian, "Cost-sensitive and hybrid-attribute measure multi-decision tree over imbalanced data sets," *Inf. Sci.*, vol. 422, pp. 242–256, Jan. 2018.

[48] N.-N. Zhang, S.-Z. Ye, and T.-Y. Chien, "Imbalanced data classification based on hybrid methods," in *Proc. 2nd Int. Conf. Big Data Res. ICBDR*, 2018, pp. 16–20.

[49] G. D. Ranasinghe, T. Lindgren, M. Girolami, and A. K. Parlikad, "A methodology for prognostics under the conditions of limited failure data availability," *IEEE Access*, vol. 7, pp. 183996–184007, 2019.

[50] K. Cheng, C. Zhang, H. Yu, X. Yang, H. Zou, and S. Gao, "Grouped SMOTE with noise filtering mechanism for classifying imbalanced data," *IEEE Access*, vol. 7, pp. 170668–170681, 2019.

[51] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Apr. 2018.

[52] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2745–2754.

[53] L. Weng, "From GAN to WGAN," 2019, *arXiv:1904.08994*. [Online]. Available: http://arxiv.org/abs/1904.08994

[54] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" 2015, *arXiv:1511.05101*. [Online]. Available: http://arxiv.org/abs/1511.05101

[55] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*. [Online]. Available: http://arxiv.org/abs/1701.00160

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[57] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[58] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[59] S. Bhatia and R. Dahyot, "Using WGAN for improving imbalanced classification performance," in *Proc. AICS*, 2019, pp. 365–375.

[60] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the improved training of wasserstein GANs: A consistency term and its dual effect," in *Proc. Int. Conf. Learn. Represent.*, Feb. 2018, pp. 1–17.

[61] C. Villani, *Optimal Transport: Old New*, vol. 338. Berlin, Germany: Springer, 2008.

[62] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[63] S. Qin and T. Jiang, "Improved wasserstein conditional generative adversarial network speech enhancement," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 181, Dec. 2018.

[64] M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, and Z. Ma, "Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification," *Inf. Sci.*, vol. 512, pp. 1009–1023, Feb. 2020.

[65] Y. Yu, B. Tang, R. Lin, S. Han, T. Tang, and M. Chen, "CWGAN: Conditional wasserstein generative adversarial nets for fault data generation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 2713–2718.

[66] Y. Liu, Z. Qin, T. Wan, and Z. Luo, "Auto-painter: Cartoon image generation from sketch by using conditional wasserstein generative adversarial networks," *Neurocomputing*, vol. 311, pp. 78–87, Oct. 2018.

[67] Y. Liu, M. Xiao, Y. Zhou, D. Zhang, J. Zhang, H. Gacanin, and J. Pan, "An access control mechanism based on risk prediction for the IoV," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.

[68] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.

[69] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[70] Y. Fathy, P. Barnaghi, S. Enshaeifar, and R. Tafazolli, "A distributed in-network indexing mechanism for the Internet of Things," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 585–590.

[71] J. Stefanowski, "Dealing with data difficulty factors while learning from imbalanced data," in *Challenges in Computational Statistics and Data Mining*. Cham, Switzerland: Springer, 2016, pp. 333–363.

[72] T. Chavdarova and F. Fleuret, "SGAN: An alternative training of generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9407–9415.

[73] W. Li, W. Ding, R. Sadasivam, X. Cui, and P. Chen, "His-GAN: A histogram-based GAN model to improve data generation quality," *Neural Netw.*, vol. 119, pp. 31–45, Nov. 2019.

[74] A. Ramdas, N. Trillos, and M. Cuturi, "On wasserstein two-sample testing and related families of nonparametric tests," *Entropy*, vol. 19, no. 2, p. 47, Jan. 2017.

[75] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *J. Amer. Stat. Assoc.*, vol. 62, no. 318, pp. 399–402, Jun. 1967.

[76] J. L. Hodges, "The significance probability of the smirnov two-sample test," *Arkiv För Matematik*, vol. 3, no. 5, pp. 469–486, Jan. 1958.

[77] F. Luo and S. Mehrotra, "Distributionally robust optimization with decision dependent ambiguity sets," *Optim. Lett.*, vol. 14, pp. 1–30, Jan. 2020.

[78] C. Saez, M. Robles, and J. M. Garcia-Gomez, "Comparative study of probability distribution distances to define a metric for the stability of multi-source biomedical research data," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 3226–3229.

[79] P. Vermeesch, "Dissimilarity measures in detrital geochronology," *Earth-Sci. Rev.*, vol. 178, pp. 310–321, Mar. 2018.

[80] J. Biteus and T. Lindgren, "Planning flexible maintenance for heavy trucks using machine learning models, constraint programming, and route optimization," *SAE Int. J. Mater. Manuf.*, vol. 10, no. 3, pp. 306–315, Mar. 2017.

[81] J. Lever, M. Krzywinski, and N. Altman, "Principal component analysis," *Nature Methods*, vol. 14, pp. 641–642, 2017. [Online]. Available: https://www.nature.com/articles/nmeth.4346#citeas, doi: 10.1038/nmeth.4346.

[82] R. E. Wright, "Logistic regression," in *Reading and Understanding Multivariate Statistics*, L. G. Grimm and P. R. Yarnold, Eds. Washington, DC, USA: American Psychological Association, 1995, pp. 217–244. [Online]. Available: https://psycnet.apa.org/record/1995-97110-007

[83] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

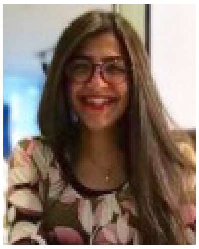
[84] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[85] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2013.

[86] C. Gondek, D. Hafner, and O. R. Sampson, "Prediction of failures in the air pressure system of scania trucks using a random forest and feature engineering," in *Proc. Int. Symp. Intell. Data Anal.* Cham, Switzerland: Springer, 2016, pp. 398–402.

[87] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[88] C. Wang, C. Deng, and S. Wang, "Imbalance-XGBoost: Leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost," *Pattern Recognit. Lett.*, vol. 136, pp. 190–197, Aug. 2020.



YASMIN FATHY received the M.Sc. degree in artificial intelligence (AI) from the AI Laboratory, Vrije Universiteit Brussel (VUB), Belgium, and the Ph.D. degree from the Institute of Communication Systems (ICS), University of Surrey. She was a Research Associate with the Computer Science Department, University College London (UCL). She is currently a Research Associate with the Department of Engineering, University of Cambridge, and a Fellow of the Higher Education Academy. Her research interests include the Internet of Things, machine learning, and data analytics.

MONA JABER (Member, IEEE) received the B.E. degree in computer and communications engineering and the M.E. degree in electrical and computer engineering from the American University of Beirut, Beirut, Lebanon, in 1996 and 2014, respectively, and the Ph.D. degree from the 5G Innovation Centre, University of Surrey, in 2017. Her Ph.D. research was on 5G backhaul innovations. She was a Telecommunication Consultant in various international firms with a focus on the radio design of cellular networks, including GSM, GPRS, UMTS, and HSPA. She was leading the IoT Research Group, Fujitsu Laboratories on Europe, from 2017 to 2019, where she focused in particular on automotive applications. She is currently a Lecturer in Internet of Things with the School of Electronic Engineering and Computer Science, Queen Mary University of London. Her research interests include cyber-physical systems, data-driven digital twins, and AI/ML applications in the automotive industry.

ALEXANDRA BRINTRUP received the Ph.D. degree from Cranfield University, Cranfield, U.K. She is currently a Lecturer in digital manufacturing with the University of Cambridge, Cambridge, U.K. She develops intelligent systems to help organizations navigate through complexity. Her main work in this area includes system development for digitized product lifecycle management. She uses artificial intelligence paradigms, particularly for data analytics and automated decision making. She held postdoctoral and fellowship appointments with the University of Cambridge and the University of Oxford. She teaches operations management and decision engineering. Her research interest includes the modeling, analysis, and control of dynamical and functional properties of emergent manufacturing networks.