Neurocomputing 561 (2023) 126891

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Interpretable machine learning assessment

Henry Han [a,*], Yi Wu [b], Jiacun Wang [c], Ashley Han [d]

[a] The Laboratory of Data Science and Artificial Intelligence Innovation, Department of Computer Science, School of Engineering and Computer Science, Baylor University, Waco, TX 76798, USA
[b] Music and Audio Research Laboratory, New York University, New York, NY 10003, USA
[c] Department of Computer Science and Software Engineering, Monmouth University, West Long Branch NJ 07764, USA
[d] Skyline High School, Ann Arbor MI 48103, USA

ARTICLE INFO

Keywords:
D-index
Interpretability
Breakeven
Imbalanced point
Learning singularity problems

ABSTRACT

With the surge of machine learning in AI and data science, there remains an urgent need to not only compare the performance of different methods across diverse datasets but also to analyze machine learning behaviors with sensitivity using an explainable approach. In this study, we introduce a uniquely designed diagnostic index: d-index to tackle this challenge. This tool integrates classification effectiveness from multiple dimensions, delivering a transparent and comprehensive assessment that transcends the limitations of traditional evaluation methods in classification. We propose two innovative concepts: breakeven states and imbalanced points in this study. Integrated with the d-index, these concepts afford a more profound understanding of the learning behaviors across different machine learning models compared to the existing classification metrics. Significantly, the d-index excels as a powerful tool, identifying learning singularity problems (LSPs) that remain elusive to most current machine learning models and imbalanced learning techniques. Furthermore, leveraging the d-index, we unravel the mechanisms behind imbalanced point generation in binary and multiclass classification. We also put forth a novel technique: identifying a priori informative kernels to optimize support vector machine learning, ensuring outstanding d-index values with the fewest necessary support vectors. Moreover, we address a seldom-discussed state of overfitting in deep learning, where overfitting occurs despite the training and testing loss curves exhibiting favorable trends throughout the epochs. To the best of our knowledge, this work represents a pioneering stride in the realm of explainable machine learning assessments and will inspire further studies in this area.

1. Introduction

Machine learning (ML) has achieved remarkable success in revolutionizing data-driven problem solving in the fields of AI and data science. Its impact is far-reaching, inspiring novel applications in areas such as business, engineering, medicine, and science, and contributing to breakthroughs in AI theory [1–2]. ML has been employed to diagnose COVID-19 and other complex diseases, facilitate language translation, enable high-frequency trading, and even beat human top-players in the game of Go with elegance [3–5]. For instance, deep reinforcement learning is used in high-frequency trading (HFT) to increase automatic trading efficiency [6], while different ML models and techniques are employed in peer-to-peer (P2P) lending to predict customer credit risk and achieve efficient hedging [7]. Additionally, a variety of deep learning methods are being used to detect emotions in speech, predict odor quality for previously uncharacterized odorants, and discover latent molecular phenotypes from histopathological images [8,9]. To a significant extent, ML is transforming human society and impacting lives in unprecedented ways, playing a vital role in advancing modern data science and artificial intelligence.

Despite its remarkable success and a plethora of research initiatives aimed at its challenges, machine learning (ML) confronts enduring interpretability issues [10]. For instance, unsupervised ML techniques, including manifold learning, often find it challenging to meaningfully evaluate their dimension reduction quality in an explainable and comparative manner. Conversely, many supervised ML methods, despite their commendable outcomes, struggle to provide clear insight into their decision-making mechanisms. Take deep learning models as an example: they house millions, if not billions, of intricate parameters. This complexity, while enabling them to attain state-of-the-art results

* Corresponding author.
E-mail address: Henry_Han@baylor.edu (H. Han).

https://doi.org/10.1016/j.neucom.2023.126891
Received 21 May 2023; Received in revised form 22 September 2023; Accepted 3 October 2023
Available online 5 October 2023

across various tasks [2,11], often obscures their operational intricacies, casting them as 'black boxes'. Such obscured methodologies, despite their prowess, may not suffice in high-stakes arenas like healthcare or finance. In these sectors, comprehending the rationale behind decisions is paramount, especially when mere accuracy can, at times, obfuscate the complete narrative of ML or be biased or misleading in situations with data imbalances [11,12].

1.1. The challenge of interpretable ML assessment

Moreover, an important issue in ML explainability, one that remains underexplored in both ML and explainable AI, concerns the ability to transparently and accurately compare the performance of distinct ML models, specifically in terms of interpretable ML assessment [10]. This becomes paramount in high-stakes sectors like finance, healthcare, and disease diagnosis. To illustrate, a seemingly minor performance disparity between two ML techniques might lead to vast differences in algorithmic trading returns, potentially amounting to millions. Similarly, in healthcare, a doctor might favor an AI diagnostic tool with even a marginally reduced false positive rate over another system, despite both offering comparable diagnostic accuracy.

Nonetheless, comparing the learning performance of different ML models using existing classification evaluation measures in an accurate and explainable manner can be challenging. This is mainly because these measures, such as accuracy, recall (sensitivity), precision, and F1-score, each evaluate only one perspective of learning. For example, one ML model may have a slightly higher accuracy but a lower recall than another under the same learning task, making it almost impossible to determine which model will be more effective. Although the F1-score can be helpful in some cases, it still cannot provide a comprehensive perspective on learning evaluation and carries a risk of biased assessment, as it does not take true negatives into consideration.

Moreover, using multiple classification measures simultaneously or at least a few measures together may further complicate the interpretation of learning results as it can be challenging to consistently compare their combinations. For instance, it would be hard to determine if ML model A outperforms B or vice versa, if A achieved 87 % accuracy, 92 % sensitivity, and 70 % specificity, while B achieved 85 % accuracy, 82 % sensitivity, and 89 % specificity on the same dataset. In this case, there is not enough numerical evidence to support which will be better. Therefore, the current classification metrics pose a challenge in selecting the most efficient model from a set of candidates by assessing the learning results in an accurate and interpretable way.

Furthermore, the existing classification measures can be misleading when applied to imbalanced learning datasets. Imbalanced learning, which refers to datasets with imbalanced or extremely imbalanced label distributions, has become increasingly important in AI and data science. This is partly due to the fact that some real-world datasets are inherently imbalanced, such as P2P lending, credit risk, malware, and omics data. In such datasets, most observations belong to one or a few majority classes or sources. For instance, in credit risk or P2P lending data, only a small fraction of customers have a 'bad' credit record, while the majority have a 'good' one [12]. Similarly, in cybersecurity, only a small percentage of software is classified as malware. Many omics datasets for disease diagnosis are also imbalanced because some disease subtypes have a much lower prevalence than others in reality or in the data acquisition process.

Since imbalanced data itself introduces bias in the label distribution, the classic classification measures can produce misleading or biased evaluation results in imbalanced learning. This is because these measures can only capture one learning perspective well, which might be sufficient for balanced data but fails to provide a comprehensive learning assessment and explanation that accounts for the impact of imbalanced data.

The accuracy measure, though commonly used, can be misleading in imbalanced learning contexts. When faced with skewed data, the majority class can dominate the training process, leading to models that disproportionately recognize the majority class while neglecting the minority. As a consequence, the reported accuracy might closely approximate, or even equal, the ratio of the majority class—resulting in deceptively high values, especially when the majority ratio is large. Such a strong bias towards the majority class can drastically skew recall and specificity, diminishing the model's effectiveness in detecting minority observations. Essentially, the ML model becomes nearly 'overfitted' to the majority, largely overlooking the minority. A parallel issue can be observed with the F1-score. If the majority class is deemed positive, the F1-score can give an illusion of perfection in imbalanced learning contexts. Technically, the F1-score, whether micro or macro, assumes that precision and recall are equally important. This might not hold true for all data science and AI applications, especially for imbalanced data.

The biased accuracy measure in imbalanced learning can have adverse effects on the ML model parameter tuning process. Most parameter tuning methods, such as grid search, rely on the accuracy metric to seek the best parameters for the model. However, when working with imbalanced datasets, the accuracy measure can be misleading, and the resulting parameters may not reflect the true performance of the model. This can lead to false parameters and inaccurate learning results, even when a satisfactory accuracy cutoff is reached. Therefore, the built-in weakness in the existing classification metrics may prevent from providing explainable and accurate learning performance assessment, especially under imbalanced learning.

1.2. Related work

Numerous studies have addressed the weaknesses of traditional classification metrics and related issues in ML assessment. However, the literature lacks interpretable classification measures. For instance, Chicco and Juman demonstrated the advantages of Matthews correlation coefficient (MCC) over F1-score and accuracy in binary classification evaluation [13], while Tharwat provided a detailed review of various classification assessment measures and their influence on balanced and imbalanced data [14]. Sokolova and Lapalme conducted a systematic analysis of performance measures in classification from a measure invariance standpoint [15], and Hand and Christen highlighted the bias of F1-score by comparing it with MCC [16]. Powers noted the weakness of F1-score for imbalanced data [17]. Optitz and Burst found that the arithmetic mean of class-wise F-scores exhibited an advantage over the class-wise precision and recall means in multi-class classification [18], while Yang *et al.* introduced a generalized F1-score in multi-class classification [19]. Grandini *et al.* reviewed classification measures for multi-class classification [20], and Jurman *et al.* compared MCC and CEN (confusion entropy) error measures for multi-class classification [21]. Ballabio *et al.* examined classification performance measures using a multivariate analysis approach [22], and Boughorbel *et al.* discussed the use of MCC in optimal classifiers for imbalanced data [23].

The previous works have significantly advanced the understanding of ML performance assessment, but they may have some limitations. Almost all of them focus on evaluating and comparing the existing classification measures, rather than proposing new metrics. While Yang *et al.* introduced a generalized F1-score for multi-class classification, it is only applied to specific data [19]. Additionally, the previous works did not address the problem from an explainable AI perspective, and it remains unknown about the interpretability of the metrics. As such, some recommended measures cannot be used in explainable ML performance assessment because of the lack of interpretability. For instance, MCC suffers from its non-explainable formula and inconsistent ranges in binary and multiclass classifications, limiting its impact on AI and data science application domains [23–26]. In addition, confusion entropy (CEN) lacks interpretability even when compared to traditional metrics. Unlike accuracy, which ranges intuitively from 0 to 1, CEN does not have such a straightforward scale, making its values harder to interpret

without context [20,21].

1.3. The standards of interpretable ML assessment

It is desirable to have a novel, explainable measure to assess ML performance informatively to overcome the weakness of the existing metrics. The measure itself should own good interpretability and be easily understood by users and serve as a good discriminator to compare and select ML models. To achieve these goals, the ML assessment measure should satisfy the following standards.

It should provide a comprehensive evaluation of ML performance, assessing both binary and multiclass ML accurately and detecting subtle differences between two or more ML models. Additionally, it should not focus solely on one learning perspective (e.g., true positive ratio).
It should be sensitive to imbalanced learning by avoiding bias from traditional metrics (e.g., accuracy) and distinguishing different ML models while detecting anomalous behaviors such as underfitting or overfitting.
The calculation of the measure should be easy to conduct and self-explanatory. Avoiding non-interpretable, complicated formulas is essential. Complex formulas can increase computational costs for large datasets, present difficulties in interpretability, and limit widespread adoption.

In this study, we propose a novel classification assessment measure called the d-index, or diagnostic index, which satisfies the three standards previously mentioned. Defined as $d = \log_2(1+a) + \log_2\left(1+\frac{s+p}{2}\right)$, it leverages the learning accuracy (a), sensitivity (s), and specificity (p) to evaluate binary classification, with an extension for multiclass classification. It can detect subtle differences in performance between models, which is essential for accurate model selection. Unlike recall or precision, which only consider one aspect of classification effectiveness, the d-index synthesizes multiple perspectives to provide a more comprehensive evaluation of ML performance. More importantly, the d-index can monitor learning behaviors sensitively, especially for detecting anomaly learning statuses such as imbalanced points. It also demonstrates a high sensitivity to imbalanced ML performance and has a smooth extension to multiclass classification without changing the value range. Additionally, the d-index can detect underfitting, overfitting, and other special ML behaviors, such as learning singularity problems (LSPs), which can cause most ML models and imbalanced handling techniques to fail.

1.4. Comparing d-index with peer measures from other studies

Compared to common metrics like MCC, Cohen's Kappa, CEN, micro, macro, and weighted F1 scores, the d-index offers good interpretability and broader appeal.

The MCC ranges from -1 to 1, indicating prediction quality from discord to perfection, while the d-index's 0 to 2 range is often more intuitive. The MCC can encounter mathematically undefined scenarios, especially when a class isn't predicted or when there's complete agreement between predictions and true values. This is common in multiclass imbalanced datasets. In contrast, the d-index doesn't face such ambiguities. Further insights on the MCC and d-index comparison are in section 2.4.

Cohen's Kappa is calculated as $\kappa = \frac{P_0 - P_e}{(1-P_e)}$, where P_0 is the relative observed agreement and P_e is the expected agreement by chance. Adjusting for chance agreement, it offers a better measure ranging from -1 (complete disagreement) to 1 (perfect agreement) compared to percentage agreement. However, interpreting Kappa is challenging, especially with no agreed standard for a "good" value. While a high Kappa doesn't always reflect strong minority class performance, the d-index consistently does.

CEN's vulnerability to data noise raises interpretability concerns. Even slight dataset variations can trigger notable entropy shifts that don't always align with classifier performance. Additionally, GEN's intricate formula adds to these interpretability challenges. More details on GEN, see section 2.4.

The micro F1 score, aligning with the standard F1 score in binary classification, is derived from the harmonic mean of micro-averaged precision and recall: $F1_{micro} = 2 \times \frac{precision_{micro} \times recall_{micro}}{precision_{micro} + recall_{micro}}$. In imbalanced datasets, the micro F1 may lean towards majority class performance, whereas the d-index effectively addresses this bias.

The macro F1 score is the arithmetic average of the F1 scores of all classes: $F1_{macro} = \sum_{i=1}^{m} F1_i$, where $F1_{ii}$ is the F1 score for the i^{th} class and m is the number of classes. It can be misleading in imbalanced datasets due to its equal class weighting. The d-index reliably reflects performance across all classes, especially in imbalanced scenarios.

The weighted F1 score is an average of the F1 scores of each class, weighted by the number of true instances for each label: $F1_{imacro} = \sum_{i=1}^{m} w_i F1_i$, w_i is the weight for the F1 score for the i^{th} class. This score inherently favors larger classes. If a classifier falters on a crucial minority class, the weighted F1 score can still appear high due to good majority class performance, potentially hiding poor results on smaller classes, and its "weighting" concept can confuse unfamiliar stakeholders. In contrast, the d-index provides a balanced evaluation regardless of minority class size, offering wider appeal.

1.5. This study's contributions

In our study, we have demonstrated the superiority of the d-index in evaluating machine learning performance on benchmark datasets from high-stakes application domains, including credit risk prediction, natural language processing (NLP), and complex disease diagnosis in biomedical data science. Our results show that the d-index not only meets the urgent demand for interpretability in machine learning result assessment, but also has the potential to positively impact AI by monitoring and detecting anomalous machine learning behaviors or states. To the best of our knowledge, this is the first work on interpretable machine learning assessment and is expected to inspire future research in this field. We summarize our contribution in this study briefly as follows.

We propose the d-index, an innovative and explainable metric for both binary and multiclass classification. This measure surpasses traditional metrics, offering a more in-depth and interpretable evaluation of ML performance. Through our theoretical exploration of the d-index, we further establish its credibility as an interpretable tool for efficient model selection and detecting anomaly learning statuses across NLP, Fintech, business, and medicine datasets.
We introduce three novel ML concepts: breakeven states, imbalanced points, and learning singularity problems (LSPs). Breakeven states signal underfitting thresholds, and imbalanced points signify when the ML model is fully overfitted to the predominant class in learning. LSPs are ML challenges with confirmable 'learnability' but lead most models to produce imbalanced points. Leveraging these concepts, the d-index provides enhanced insights into ML behaviors, enriching existing theory and application. Additionally, the introduction of LSPs opens a new avenue in ML research. We also unveil the imbalanced point generation mechanisms for both binary and multiclass classification models, besides introducing a novel d-index-based LSP detection algorithm.
We've demonstrated through theoretical proof that even if data isn't imbalanced, established ML models like SVM can still falter, producing imbalanced points. We've introduced innovative techniques to pinpoint effective kernels in SVM, ensuring optimal d-index values with minimal support vectors before the real learning process begins.

In our deep learning analysis, we discovered that models, specifically transformers, can generate imbalanced points even with linearly separable data when paired with unsuitable loss functions. We also uncover a unique overfitting scenario in deep learning: instances where both training and testing loss curves consistently show positive trends across epochs, yet overfitting is still present.

The paper is organized as follows. In Section 2, we introduce the d-index and extend it to multiclass classification. We propose the concepts of the breakeven state and imbalanced point to model different ML behaviors for more interpretable ML assessments. We also provide a rigorous theoretical analysis to illustrate the special characteristics of d-index in underfitting detection and imbalanced point generation. Section 3 demonstrates the applications of d-index in binary, multiclass, and imbalanced data classification by comparing it with peer measures under various benchmark datasets in different ML applications. We further validate the superiorities of d-index in robust model selection, sensitive imbalanced learning monitoring, and learning singularity problem detection. Additionally, we present a method for distinguishing learning performance under the same d-index values for SVM, along with priori kernel selection. Section 4 discusses more applications of d-index and its limitations, as well as possible enhancements. Finally, in Section 5, we conclude this study and discuss future directions for research in interpretable ML assessment.

2. Diagnostic index (d-index)

The d-index is an explainable classification metric offering a nuanced evaluation of classification efficacy, particularly in imbalanced learning. Originally devised by the first author for RNA-seq dataset comparisons, we've expanded its applicability from binary to multiclass classifications [24].

2.1. D-index

Given an implicit prediction function $\widehat{f}(x): x \rightarrow \{-1, 1\}$ constructed from training data $X_r = \{x_i, y_i\}_{i=1}^{m}$ under an ML model Θ, where each sample $x_i \in R^k$ and its label $y_i \in \{-1, 1\}, i = 1, 2, \cdots m$, d-index evaluates the effectiveness of $\widehat{f}(x)$ to predict the labels of test data $X_s = \{x_j', y_j'\}_j^l$, where x_j' is a test sample and its label $y_j' \in \{-1, 1\}$. The d-index is defined as:

$$d = \log_2(1 + a) + \log_2\left(1 + \frac{s + p}{2}\right) \tag{1}$$

where a, s, and p represent the corresponding accuracy, sensitivity, and specificity in diagnosing test data X_s respectively. The d-index is in (0, 2]. The larger the d-index value, the better the predictability of $\widehat{f}(x)$, i.e., the better learning performance achieved by the ML model Θ. d-index logarithmically depicts the trend of the accuracy a and balanced accuracy: $\frac{s+p}{2}$, which is the average of the true positive and negative ratios, in a log mode because of $2^d = (1 + a)\left(1 + \frac{s+p}{2}\right)$.

The accuracy $a = \frac{TP+TN}{TP+FN+TN+FP}$ is the ratio between the number of correctly predicted positive (+1) and negative (-1) samples and the total number of samples in query. TP and TN represent the number of correctly predicted positive and negative samples, respectively: $TP = \left|\left\{x_j' : \widehat{f}(x_j') = 1 \wedge y_j' = 1\right\}\right|$; $TN = \left|\left\{x_j' : \widehat{f}(x_j') = -1 \wedge y_j' = -1\right\}\right|$. In contrast, FN and FP represent the number of incorrectly predicted positive and negative samples, respectively: $FN = \left|\left\{x_j' : \widehat{f}(x_j') = -1 \wedge y_j' = 1\right\}\right|$; $FP = \left|\left\{x_j' : \widehat{f}(x_j') = 1 \wedge y_j' = -1\right\}\right|$.

While accuracy provides a measure of overall classification performance, it does not consider the prediction function $\widehat{f}(x)$'s performance on different subgroups. As a result, accuracy may not be an appropriate metric for evaluating classification performance in imbalanced datasets or when the cost of misclassification differs across subgroups.

The sensitivity (recall) $s = \frac{TP}{TP+FN}$ is the ratio of correctly predicted positive samples to the total number of true positive samples. It measures the ability of the model to identify all positive samples, i.e., the true positive rate (TPR). On the other hand, the specificity $p = \frac{TN}{TN+FP}$ is the ratio of correctly predicted negative samples to the total number of true negative samples. It measures the ability of the model to identify all negative samples, i.e., the true negative rate (TNR). Ideally, the prediction function $\widehat{f}(x)$ should be equally likely to predict + 1 and −1 samples. However, in practice, when the training data is imbalanced, $\widehat{f}(x)$ may have a bias towards predicting the majority type sample. This can result in sensitivity and specificity values that demonstrate extreme values when evaluating classification performance in imbalanced datasets.

If we assume there exist $N = N_p + N_n$ samples in query consisting of $N_p = TP + FN$ positive samples and $N_n = TN + FP$ negative samples, we can rewrite the d-index definition as follows,

$$d = \log_2\left(\frac{N + TN + TP}{N}\right) + \log_2\left(\frac{2N_pN_n + \mathrm{TP}N_n + \mathrm{TN}N_p}{2N_pN_n}\right) \tag{2}$$

This rewritten formula for the d-index provides a more comprehensive explanation of learning performance compared to classic metrics, as it includes all elements involved in classification. Additionally, the weights of TP and TN in $\frac{2N_pN_n+\mathrm{TP}N_n+\mathrm{TN}N_p}{2N_pN_n}$ help to prevent possible biased impacts from imbalanced data, such as when N_p is much greater than N_n, on the classification process.

In the following section, we introduce new ML concepts: "breakeven", "imbalanced point," and "learning singularity" to exploit the potentials of the d-index for the sake of interpretable ML result assessment.

2.2. Breakeven states and underfitting

The d-index exhibits unique characteristics in the breakeven state and can effectively detect various forms of underfitting rigorously. More importantly, it eliminates the ambiguity and bias associated with using conventional metrics.

2.2.1. Breakeven states

A **breakeven** state in binary classification for an ML model Θ is a state in which the model classifies a sample as positive or negative with an equal likelihood. for a sample x with a label $y \in \{-1, 1\}$, the prediction function $\widehat{f}(x)$ of the ML model Θ maintains $Pr\{\widehat{f}(x) = 1|\Theta\} = Pr\{\widehat{f}(x) = -1|\Theta\} = 50\%$ in prediction.

The breakeven state is a critical indicator for an ML model represented by the symbol Θ, as it denotes the state where the model performs no better than a random classifier and fails to provide any significant insights during the learning process. This point serves as a measure to evaluate the relevance of using ML. If the performance of the model Θ drops below this point, it leads to underfitting, where the model performs worse than a random classifier. In such a scenario, ML loses its purpose, and the model's performance degrades to that of a random coin-flipping process. Moreover, if the performance of the ML model continues to deteriorate below the breakeven point, it can encounter severe underfitting, which can significantly impact its predictive accuracy.

The breakeven state under binary classification has certain outcomes that can be analyzed to gain a holistic understanding of the performance of the machine learning model using the d-index.

Lemma 1. *The d-index is* $2\log_2\left(\frac{3}{2}\right)$ *if an ML model is in the break-even state under binary classification.*

Proof. Under the breakeven state, the ML model Θ is a random

classifier with a 50 % probability to conduct correct prediction, i.e., TP = FN = $N_p/2$and TN = FP=$N_n/2$. Let $\frac{N_n}{N_p} = \eta$, we have,

$$a = \frac{TP + TN}{TP + FN + TN + FP} = \frac{TP}{N_p + N_n} + \frac{TN}{N_p + N_n} = \frac{1}{2 + 2\eta} + \frac{\eta}{2 + 2\eta} = \frac{1}{2} \quad (3)$$

Similarly, $s = \frac{TP}{TP+FN} = \frac{1}{2}$, $p = \frac{TN}{TN+FP} = \frac{1}{2}$. Thus $d = \log_2(1 + 1/2) + \log_2\left(1 + \frac{1/2+1/2}{2}\right) = 2\log_2\left(\frac{3}{2}\right)$ under the break-even state.

Lemma 2. *If an ML model is in the breakeven state under binary classification, then AUC (area under the curve) AUC* $= \frac{1}{2}(s+p) = \frac{1}{2}$, $F1 = \frac{2}{3+\eta}$, *where* $\eta = \frac{N_n}{N_p}$ *is the ratio between the number of negative samples over that of the negative ones.*

Proof. According to the result from Lemma 1, it is easy to have $AUC = \frac{1}{2}(s+p) = \frac{1}{2}$ because sensitivity and specificity both are ½ at the breakeven state. Similarly, we have.

$$F1 = \frac{TP}{TP + (FN + FP)/2} = \frac{N_p/2}{N_p/2 + \left(N_p + N_n\right)/2} = \frac{2}{3 + N_n/N_p} = \frac{2}{3 + \eta} \quad (4)$$

Theorem 1. *The F1 score of an ML model in the breakeven state falls in* $\left[\frac{1}{2}, \frac{2}{3}\right)$.

Proof. Based on the findings from Lemma 2, the F1 score can be calculated using the formula $F1 = \frac{2}{3+\eta}$, where η represents the ratio of negative samples to positive samples in the training data. When the number of positive and negative samples is balanced, F1 = 0.5. However, when η → 0, meaning the training data is entirely dominated by negative samples, the F1 score increases and approaches a value of 2/3.

2.2.2. Underfitting detection

The d-index is a useful metric to detect underfitting in machine learning models, as it provides a clear indicator of underfitting when the value is less than $2\log_2(3/2)$, the d-index of the breakeven state. This makes it a more reliable method to identify underfitting compared to traditional approaches such as observing accuracy or AUC. These traditional methods may not be robust enough, as accuracy and AUC values can be misleading when the data is imbalanced, and they are not necessarily definitive indicators of underfitting.

For example, a binary machine learning classifier can face underfitting when its accuracy is lower than 50 %. This low accuracy suggests that the model is not capturing the underlying patterns in the data. However, it could also be biased due to imbalanced data. For instance, consider a training dataset containing 100 samples, where 25 are true positives (TP), 20 are true negatives (TN), 55 are false negatives (FN), and 5 are false positives (FP). In this case, an accuracy score of 45 % may not necessarily signify underfitting since the model can still accurately classify 80 % of the negative samples, resulting in a high specificity of 80 %. However, it's worth noting that the model's d-index in this context will be 1.1838, which is above the d-index of the breakeven state ($2\log_2\left(\frac{3}{2}\right) = 1.1699$) suggesting that there is no underfitting.

Similarly, an AUC value of 0.49 does not necessarily indicate that the machine learning model is encountering underfitting. This is because an AUC value of 0.49 could also indicate that the model is overfitted to the positive samples, resulting in a high sensitivity and low specificity. For example, a sensitivity of 98 % and a specificity of 0 % would result in an AUC value of 0.49, i.e., the area under the ROC curve is equal to the area of a diagonal line, which represents the performance of a model that makes less than random predictions.

Theorem 2 states d-index in general binary classification falls in $\left(2\log_2\left(\frac{3}{2}\right), 2\right]$ when there is no underfitting.

Theorem 2. *The range of d-index is between* $2\log_2\left(\frac{3}{2}\right)$ *and 2 if we assume no underfitting in binary classification. When d-index* $< 2\log_2\left(\frac{3}{2}\right)$, *which is the d-index of the breakeven state, the ML model encounters underfitting.*

Proof. The upper bound of d-index is 2. It indicates an ideal learning performance, in which$a = s = p = 100$ %, i.e., all samples are perfectly classified . On the other hand, the lower bound of d-index comes from the breakeven state, i.e. $d \geq 2\log_2\left(\frac{3}{2}\right) = 1.1699$. When d-index is less than that of the breakeven state, it indicates that the ML model performs worse than a random classifier and encounters underfitting.

2.3. The imbalanced point detection

In addition to detecting underfitting, the d-index is a superior metric for model selection because it demonstrates good sensitivity in assessing imbalanced learning performance by sensitively detecting anomalous learning states. This section proposes a new imbalanced point concept to model the exceptional learning state in imbalanced learning for the sake of explainable and sensitive imbalanced learning assessment. In addition, it compares d-index with existing AUC and MCC measures in term of interpretability.

2.3.1. Majority ratio

The majority ratio serves as the foundation for modeling imbalanced learning. In binary classification, it is defined as the count of the majority label divided by the total count of labels. Given training data $X_r = \{x_i, y_i\}_{i=1}^m, y_i \in \{-1, 1\}$ in binary classification, the majority ratio under the binary classification is calculated as:

$$\gamma = \frac{\max(|\{x_i : y_i = 1\}|, |\{x_i : y_i = -1\}|)}{|\{x_i : y_i = 1\}| + |\{x_i : y_i = -1\}|} \quad (5)$$

The majority ratio can theoretically be greater than 50 % in binary classification. However, in practice, if the input data is imbalanced, the majority ratio may range from 75 % to even 99 %+. This is because it is possible that almost all observations belong to the majority class, while the minority class only contains a very small percentage of observations (e.g., less than 5 %).

Majority ratio in multiclass classification is defined as the ratio of the maximum class count over the total counts in the training data $X_r = \{x_i, y_i\}_{i=1}^m, y_i \in \{1, 2, \cdots k\}$, where the number of classes $k > 2$,

$$\gamma = \frac{\max(|\{x_i : y_i = 1\}|, |\{x_i : y_i = 2\}|, \cdots |\{x_i : y_i = k\}|)}{\sum_{i=1}^k |\{x_j : y_j = i\}|} \quad (6)$$

The majority ratio can take a wider range of values in multiclass classification, and its value is influenced by the class imbalance present in the data. Generally, the larger the majority ratio, the more negative impacts it can have on machine learning. This is because the ML model is more likely to lose its learning capabilities by classifying almost all minority samples as the majority class.

It is important to note that the majority ratio of the training data may not necessarily be the same as that of the test or validation data in machine learning, especially when the data has a limited number of minority observations. In such cases, the majority ratio of the training data should be replaced by that of the test or validation data when calculating classification metrics to ensure accurate performance evaluation of the model.

2.3.2. Imbalanced point

An imbalanced point refers to a learning state in which the ML model Θ loses its learning capability by predicting all minority samples as majority ones in imbalanced or even general learning. This is technically an overfitting state where the model is overfitted to the majority type data. Without loss of generality, we can describe this phenomenon under binary classification as follows.

Imbalanced point. Given training data $X_r = \{x_i, y_i\}_{i=1}^m, y_i \in \{-1, 1\}$ in binary classification with the majority ratio γ, in which the majority

type is the positive type: '+1' under an ML model Θ, $\widehat{f}(x|\Theta, X_r)$ is the prediction function built under the model Θ using the training data. The ML model is said to reach an imbalanced point, provided $\widehat{f}(x|\Theta, X_r) = +1$ for $\forall x$ whose label is unknown.

In other words, at the imbalanced point, the ML model will classify all majority samples correctly but all minority samples incorrectly. The following lemma states that the special values of classic metrics at the imbalanced point.

Lemma 3. *The classic metrics have the following special values at the imbalanced point in binary classification with majority ratio γ, assuming the majority type is positive. These values are accuracy $\alpha = \gamma$, sensitivity $s = 100$ %, specificity $p = 0$ %, and F1 score* $F1 = \frac{2\gamma}{\gamma+1}$.

Proof. We assume there are N samples in query under the ML model Θ at the imbalanced point, then we have $TP = N \times \gamma$, $FN = 0$, $TN = 0$, and $FP = N \times (1-\gamma)$. This is because all majority samples, which are assumed positive, are correctly predicted: $TP = N \times \gamma$ and $TN = 0$. Similarly, all minority samples are falsely predicted: $FP = N \times (1-\gamma)$ and $TN = 0$. Therefore, the learning accuracy $\alpha = \frac{TP+TN}{TP+FN+TN+FP} = \frac{N\times\gamma}{N\times\gamma+N\times(1-\gamma)} = \gamma$. Similarly, *precision* $= \frac{TP}{TP+FP} = \frac{N\times\gamma}{N\times\gamma+N\times(1-\gamma)} = \gamma$, sensitivity $s = \frac{TP}{TP+FN} = \frac{N\times\gamma}{N\times\gamma} = 100\%$, and specificity $p = \frac{TN}{TN+FP} = 0\%$. Moreover,F1 $= \frac{TP}{TP+(FN+FP)/2} = \frac{N\gamma}{N\gamma+(N\times(1-\gamma))/2} = \frac{2\gamma}{\gamma+1}$.

The Lemma 3 states that classic metrics such as accuracy, precision, and F1 score become biased and lose their interpretability when assessing machine learning results on imbalanced datasets. This is especially true when the majority class ratio (γ) is very high. In such cases, these metrics may appear to be good (e.g., 90 % accuracy, 94.73 % F1 score, 100 % sensitivity, and 90 % precision), but they can be deceptive as they do not reflect the true learning status. Essentially, the model is only making a majority class prediction, regardless of the input.

On the other hand, the following theorem shows that d-index can provide more interpretable and transparent learning assessment at the imbalanced point. The d-index overcomes the bias of the classic metric by reporting the true learning status. By doing so, it provides a clear indication of whether the model has learned to differentiate between the minority and majority classes or is merely predicting the majority class.

Theorem 3. ***Binary imbalanced point theorem.*** *Given an implicit prediction function $\widehat{f}(x) : x \to \{-1, 1\}$ constructed from training data $X_r = \{x_i, y_i\}_{i=1}^m$ with the majority ratio γ, under the ML model Θ, then at an imbalanced point, the ML model has the d-index $d = \log_2\left(\frac{3(1+\gamma)}{2}\right)$.*

Proof. Without loss of generality, we assume the majority type is positive at the imbalanced point. We have accuracy $\alpha = \gamma$ sensitivity $s = 100\%$, and specificity $p = 0\%$ according to the Lemma 3, then the d-index value:

$$d = \log_2(1+\gamma) + \log_2\frac{3}{2} = \log_2\left(\frac{3(1+\gamma)}{2}\right) \tag{7}$$

Imbalanced point detection using d-index. The d-index is a more appropriate measure than traditional classification metrics for detecting the imbalanced point in imbalanced learning. This is because the d-index is calculated as a function of the majority ratio γ, making it a more accurate measure in such scenarios. For instance, when γ is 90 %, an F1-score of 94.74 % and accuracy of 90 % may suggest that the model has good learning performance. However, a low d-index value of 1.5110 indicates that the model's performance is actually poor.

2.3.3. Imbalanced point generation

It's worth noting that not all imbalanced learning scenarios will lead to the occurrence of the imbalanced point. However, its appearance is a clear indication that the machine learning model has failed to handle the imbalanced data. In general, the higher the majority ratio, the more likely the imbalanced point will occur. Once the imbalanced point is generated, the learning process fails and becomes trapped in a special overfitting state. This overfitting state can result in the machine learning model becoming "too rigid" to recognize any minority samples, leading to poor classification performance on these samples.

We employ a k-NN model to illustrate how the imbalanced point is generated in imbalanced learning. For an incoming test sample, all its nearest neighbors in k-NN will have more majority samples than the minority ones because of imbalanced data. Thus, an incoming sample will be classified as the majority type inevitably or at least with a very high likelihood no matter what kinds of voting schemes employed. Therefore, all test samples will be classified as the majority type. Finally, the k-NN learning accuracy will be the majority ratio of the test dataset, which will be the majority ratio γ of the training dataset or approximate it.

Approximately imbalanced points (AIPs). In practice, the imbalanced point may appear as an approximately imbalanced points (AIP) under an ML model, i.e., accuracy will be approximately the majority ratio γ and d-index will be close to $\log_2\left(\frac{3(1+\gamma)}{2}\right)$ This is because the ML model may classify few majority samples as the minority type or vice versa in learning.

For example, if the training dataset has the majority ratio $\gamma = 0.92$, then its imbalanced point will be reached d-index $\log_2\left(\frac{3(1+\gamma)}{2}\right)_{|\gamma=0.92} = 1.52$ with accuracy 0.92 under an ML model according to the Theorem 3. However, when some ML model achieves d-index 1.51 with an accuracy 0.89, such an AIP is still an imbalanced point practically. The following theorem estimates the range of d-index under imbalanced data classification, where the d-index touches its lower bound at the imbalanced point or AIP.

Theorem 4. *The d-index has the following range when the training dataset has a large enough majority ratio (e.g., $\gamma \geq 75\%$) under an ML model Θ,*

$$\log_2\left(\frac{3(1+\gamma)}{2}\right) + \varepsilon \leq d \leq 2 \tag{8}$$

where $|\varepsilon| > 0$ is a small ratio related to the model Θ. The better the model's learning capability, the more likely that d-index move right with respect to the imbalanced point's d-index.

Proof. The worst situation will be that the prediction function $\widehat{f}(x)$ built from the training dataset would misclassify all the minority samples incorrectly but all the majority samples correctly, i.e., accuracy $a = \gamma$, $d = \log_2\frac{3(1+\gamma)}{2}$, and the ML model attains the imbalanced point.

However, there exists a likelihood that few majority samples might be misclassified due to unpredictable nonlinearity or artifacts during the ML process. At the same time, some minority samples can be also correctly learned in the procedure. If the former had more contributions to the learning results, then d-index would be $< \log_2\left(\frac{3(1+\gamma)}{2}\right)$slightly, because the decrease of the accuracy and sensitivity will be more than the increase of the specificity, i.e., $\varepsilon < 0$. Otherwise, d-index would be $> \log_2\left(\frac{3(1+\gamma)}{2}\right)$ because $\varepsilon > 0$. Besides the characteristics of the imbalanced dataset, the better the ML model, the higher likelihood its d-index moves to right.

According to the previous results, we have the following d-index estimations of the AIP and the imbalanced point under the extremely imbalanced binary classification.

Corollary 1. *Given training data $X_r = \{x_i, y_i\}_{i=1}^m, x_i \in R^k$, $y_i \in \{-1, 1\}$ with the majority ratio $\gamma > 50\%$ in binary classification under the ML model Θ, if there exists an AIP, then its accuracy and d-index will be close to the majority ratio γ and $\log_2\left(\frac{3(1+\gamma)}{2}\right)$ respectively.*

Extremely imbalanced cases in binary classification. Since d-index is the function of the majority ratio γ at the imbalanced point:

$d(\gamma) = \log_2\left(\frac{3(1+\gamma)}{2}\right)$, it will approach $\log_2 3(1.5850)$: $\lim_{\gamma\to 1}\log_2\left(\frac{3(\gamma+1)}{2}\right) = 1.5850$ at the imbalanced point in the extremely imbalanced binary classification, in which the majority ratio $\gamma\to 1$.

2.3.4. Imbalanced point generation under non-imbalanced data

It is worth noting that an imbalanced point can even be generated under non-imbalanced data. The non-imbalanced data refers to those data with a majority ratio close to 50 % (e.g., 55 %). While technically considered balanced data, it is possible for certain machine learning models to generate an imbalanced point in these cases. To illustrate this point, we will use support vector machines (SVMs) as an example, given their importance in machine learning.

The following theorem highlights that SVM can generate an imbalanced point even when the input data is non-imbalanced data, provided that the kernel matrix is an identity or approximately an identity matrix. This anomalous learning state can be difficult to detect using traditional metrics like accuracy, but d-index offers a straightforward way to identify it.

Theorem 5. ***Imbalanced point generation under SVM.*** *Given training data* $X_r = \{x_i, y_i\}_{i=1}^m, x_i \in R^q, y_i \in \{-1, 1\}$ *under binary SVM classification with a kernel* $k(x,y)$, *let* $\widehat{f}(x) : x \to \{-1, 1\}$ *be the prediction function constructed in training. If the SVM kernel matrix K is an identity or approximately identity matrix, i.e.,* $\forall x_i, x_j \in X_r, i \neq j, K_{ij} = k(x_i, x_j) \approx 0$, *and* $K_{ii} = k(x_i, x_i) = 1$, *then there exists an imbalanced point in SVM learning, i.e., for* $\forall x$ *with an unknown type,* $\widehat{f}(x) = +1$, *if we assume the majority type is the positive '+1'.*

Proof. Given training data $X_r = \{x_i, y_i\}_{i=1}^m, x_i \in R^q$, $y_i \in \{-1, 1\}$, the SVM model seeks the optimal hyperplane $w^T\varphi(x) + b$ by finding the normal vector $w \in R^q$ and offset $b \in R^1$ by solving the quadratic programming problem:

$$\begin{aligned} &min_w \frac{1}{2}w^T w + C\sum_{i=1}^m \xi_i, w \in R^q, \xi_i \in R, b \in R \\ &s.t. y_i\left(w^T\varphi(x_i) + b\right) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2\cdots m, \end{aligned} \tag{9}$$

where $\xi_i, i = 1, 2\cdots m$ are slack variables, $C > 0$ is the penalty term, and $\varphi(x)$ is the function mapping input data into the high-dimensional Hilbert space, where $k(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$. Thus, the SVM prediction function is $\widehat{f}(x) = sign\left(\sum_{i=1}^m \alpha_i y_i k(x_i, x) + b\right)$, where $\alpha_i \geq 0$ are the solutions of the dual problem of the original quadratic programming problem,

$$\begin{aligned} &max_\alpha -\frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m y_i y_j k(x_i, x_j)\alpha_i\alpha_j + \sum_{i=1}^m \alpha_i \\ &s.t. \sum_{i=1}^m \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2\cdots m \end{aligned} \tag{10}$$

Since non-diagonal kernel matrix terms are zero or approximately zero, i.e., $k(x_i, x) \approx 0$, the classification result will only depend on the offset term b, i.e., $\widehat{f}(x) = sign(b)$. The offset b can be determined by the normal vector $w = \sum_{i=1}^m \alpha_i\varphi(x_i)y_i$, i.e., $b = -\frac{1}{2}(w^T\varphi(x_+) + w^T\varphi(x_-)$, where x_+ and x_- are two support vectors with $+1$ and -1 labels respectively, i.e.,

$$b = -\frac{1}{2}\left(\sum_{j=1}^m \alpha_j y_j k(x_j, x_+) + \sum_{j=1}^m \alpha_j y_j k(x_j, x_-)\right) \tag{11}$$

Since the kernel matrix is an identity or approximately identity matrix: $k(x_j, x_+) \approx 0$, $k(x_j, x_-) \approx 0$, $k(x_+, x_+) = k(x_-, x_-) = 1$, we have $b = -\frac{1}{2}(\alpha_+ - \alpha_-)$, where α_+ and α_- are the alpha values corresponding to the two support vectors. Moreover, we have the following trivial problem by applying $k(x_i, x_j) \approx 0$, for $i \neq j$ and $k(x_i, x_i) = 1$ to the original dual:

$$\begin{aligned} &max_\alpha -\frac{1}{2}\sum_{i=1}^m \alpha_i\alpha_i + \sum_{i=1}^m \alpha_i \\ &s.t. \sum_{i=1}^m \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2\cdots m \end{aligned} \tag{12}$$

The trivial dual exists solutions: $\alpha_+ = \frac{m_-}{m}$, $\alpha_- = \frac{m_+}{m}$, where $m_+ = |\{x_i : y_i = +1\}|$, $m_- = |\{x_i : y_i = -1\}|$. Thus, we have the final offset $b = \frac{m_+ - m_-}{2m}$ and prediction function $\widehat{f}(x) = sign\left(\frac{m_+ - m_-}{2m}\right) = sign(m_+ - m_-)$.

The prediction function of imbalanced learning implies that a query sample's class type is determined by the majority class in the training data. If the majority class is positive, then every query sample will be classified as positive (+1), i.e., for $\forall x$ with an unknown type, $\widehat{f}(x) = +1$, regardless of its actual class. This generates an imbalanced point, where all samples are classified as the majority class.

This type of imbalanced point generation under SVM occurs frequently in high-dimensional omics data when the Gaussian kernel is not properly set. For example, when the parameter η is set improperly small (e.g., $\eta = 0.5$) in the Gaussian kernel: $k(x,y) = e^{-\eta\|x-y\|^2}$, the corresponding SVM kernel matrix will become the identity matrix or approximately one because of the large pairwise distance between omics samples caused by the molecular signal amplification mechanism [27]. As a result, even non-imbalanced data can generate an imbalanced point because all samples are classified as the majority type.

As an example, we applied SVM to a breast cancer omics dataset [27], consisting of 97 patient samples across 24,188 genes, with 46 patients exhibiting 5-year metastasis and 51 patients without. Although the dataset has a majority ratio of only 52.58 %, we observed the generation of an imbalanced point under SVM with the Gaussian kernel $k(x, y) = e^{-\|x-y\|^2/2}$ under the 5-fold cross validation. Specifically, all minority samples were classified as the majority type, resulting in a d-index of $d = \log_2\left(\frac{3(1+\gamma)}{2}\right)_{\gamma=51/97} = 1.1945$, accuracy $\alpha = \gamma = 0.5258$, sensitivity $s = 1.0$, and specificity $p = 0.0$. This highlights that even non-imbalanced data can lead to an imbalanced point under SVM.

2.3.5. The difference between the imbalanced point and breakeven

Both the breakeven state and imbalanced point describe anomalous states in classification where an ML model loses its learning capabilities. However, they are caused by different reasons and occur in different datasets. The breakeven state is mainly caused by the ML model being unsuitable for the input data, whether balanced or not. On the other hand, the imbalanced point is primarily caused by the high majority ratio in imbalanced data, along with improper parameter settings in the ML model (such as in SVM). The breakeven state can be considered the "inflexion point" at which underfitting occurs, while the imbalanced point is the state at which overfitting occurs and the ML model can only recognize the majority type. In this case, the learning process is "hijacked" by the majority samples, and the ML model can become "too overfitted" to recognize the minority samples.

Traditional classification metrics lack enough sensitivity and good interpretability to distinguish the two learning states well because they generally only reflect a single learning perspective. However, d-index can model and interpret them accurately and detect the anomalous states sensitively because it explains and models learning behaviors from more comprehensive perspectives. To some degree, with the help of d-index, the two new concepts would contribute to interpreting ML results more accurately and rigorously.

2.4. D-index is more representative and explainable

The proposed d-index also demonstrates its superiority to widely used non-accuracy measures such as AUC and MCC in monitoring imbalanced learning by providing interpretable assessment. For example, let us consider an ML model Θ that produces TP = 90, FN = 0, TN = 0, FP = 10 results under a majority ratio: $\gamma = 0.9$ for 100 samples

in query. The accuracy (90 %), sensitivity (100 %), F1 score (0.9474), and precision (0.9) values indicate good performance, while the specificity (0 %) and AUC (0.5) values suggest the opposite. However, the d-index value of 1.5110 indicates that it is a poor performance case.

Compared to d-index, AUC is less informative because it cannot distinguish between the breakeven and imbalanced point states, both of which have AUC values of 0.5. However, d-index can differentiate between these two states, with values of 1.1699 and 1.5110 + ε, respectively, as per Theorem 4. Thus, d-index provides a more comprehensive and informative evaluation of the two learning states compared to the AUC metric. The d-index not only includes information from AUC, which is the average of sensitivity and specificity, but also considers the overall classification performance.

In addition, compared to MCC that takes values in [-1,1], d-index that falls in (0,2] is more intuitive and explainable. It is built upon the three widely used measures accuracy, sensitivity, and specificity. On the other hand, MCC can be viewed as a special discretized version of the Pearson correlation for binary variables [28]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}} \tag{13}$$

Despite MCC's utility, its formula's complexity often renders it less intuitive to interpret. Furthermore, MCC cannot differentiate between the breakeven and imbalanced point, returning zero values for both scenarios. This inability makes MCC less effective in discerning these unique learning states. Conversely, as highlighted before, the d-index distinctly values these learning states, proving it more representative than both AUC and MCC.

Furthermore, the d-index is easy to understand and explain compared to the less utilized metric GEN, defined by the formula: $CEN = \sum_{i=1}^{c}\sum_{j=1}^{c} p_{ij}\log\left(\frac{cp_{ij}}{p_{i,+}\,p_{+j}}\right)$, where c is the number of classes, p_{ij} is the probability of class i being predicted as class j, $p_{i,+}$ is the marginal probability of the true class being i, and p_{+j} is the marginal probability of the predicted class being j. Besides its complicated calculation, the range of the CEN depends on the number of classes, meaning that the CEN values are not directly comparable across datasets with a different number of classes. Therefore, it is almost impossible to use it to detect the breakeven and imbalanced points.

2.5. Multiclass d-index

We extend d-index to the multiclass by averaging the local d-index values for each class in the multiclass. The extension consists of the following three steps. The first step redefines the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values for each class.

Given a class i in total k classes: $\Lambda = \{1,2,\cdots k\}$, we define the samples belonging to the class i as positive and the samples belonging to other classes ($\neg i$) as negative respectively. Suppose $\widehat{f}(x)$ is the prediction function built under an ML model Θ using training data $X_r = \{x_i, y_i\}_{i=1}^{m}$, $y_i \in \Lambda$. We have the following definitions of TP, TN, FP, and FN for each class $i = 1, 2, \cdots k$. Given test data $X_t = \{x_j', y_j'\}_{j=1}^{l}$, $y_j' \in \Lambda$, we have

- tp_i: TP of the class i: the number of samples belonging to class i correctly classified as class i, i.e.,$i \to i$,

$$tp_i = \left|\left\{x_j' : \widehat{f}(x_j') = i \wedge y_j' = i\right\}\right| \tag{14}$$

- fp_i: FP of the class i: the number of *non-i* class samples falsely classified as the class i, i.e.,$\neg i \to i$,

$$fp_i = \left|\left\{x_j' : \widehat{f}(x_j') = i \wedge y_j' \neq i\right\}\right| \tag{15}$$

- tn_i : TN of i: the number of *non-i* class samples correctly classified as *non-i class,* i.e.,$\neg i \to \neg i$

$$tn_i = \left|\left\{x_j' : \widehat{f}(x_j') \neq i \wedge y_j' \neq i\right\}\right| \tag{16}$$

- fn_i: FN of i: the number of class i samples falsely classified as the *non-i* class, i.e.,$i \to \neg i$

$$fn_i = \left|\left\{x_j' : \widehat{f}(x_j') \neq i \wedge y_j' = i\right\}\right| \tag{17}$$

The second step calculates the local accuracy $\alpha_i = \frac{tp_i+tn_i}{tn_i+tp_i+fp_i+fn_i}$, sensitivity $s_i = \frac{tp_i}{fn_i+tp_i}$, specificity $p_i = \frac{tn_i}{tn_i+fp_i}$, and d-index $d_i = \log_2(1+\alpha_i) + \log_2\left(1+\frac{s_i+p_i}{2}\right)$ for each class $i \in \Lambda$:

The third step calculates the final d-index as the expected value of the local d-index values for all classes as

$$d = \frac{1}{k}\sum_{i=1}^{k}\left(\log_2(1+\alpha_i) + \log_2\left(1+\frac{s_i+p_i}{2}\right)\right) \tag{18}$$

2.6. Breakeven state in multiclass classification

Breakeven state in multiclass classification for a multiclass ML model Θ is a state in which the model classifies a sample as one of labels in $\Delta = \{1, 2, \cdots k\}$ with an equal likelihood. For a sample x with a label $y \in \Delta$, the prediction function $\widehat{f}(x)$ of the ML model Θ maintains $Pr\{\widehat{f}(x) = 1|\Theta\} = \cdots Pr\{\widehat{f}(x) = k|\Theta\} = \frac{1}{k}$ in prediction. The d-index of an ML model under the breakeven in multiclass classification is $2\log_2\left(\frac{k+1}{k}\right)$ by extending the previous result in binary classification.

Lemma 4. *The d-index is $2\log_2\left(\frac{k+1}{k}\right)$, where k is the number of classes: $\Delta = \{1, 2, \cdots k\}$, if an ML model is in the break-even state under multiclass classification.*

Proof. According to the definitions of multiclass d-index and breakeven, we have local accuracy, sensitivity, and specificity for each class $i \in \Lambda$ $\alpha_i = s_i = p_i = \frac{1}{k}$. Then the d-index:$d = \frac{1}{k}\sum_{i=1}^{k}\left(\log_2\left(1+\frac{1}{k}\right) + \log_2\left(1+\frac{1}{k}\right)\right) = 2\log_2\left(\frac{k+1}{k}\right)$.

Theorem 6. *The range of d-index in multiclass classification is $\left(2\log_2\left(\frac{k+1}{k}\right), 2\right]$, where k is the number of classes: $\Delta = \{1, 2, \cdots k\}$. If we assume no underfitting in learning. When d-index $< 2\log_2\left(\frac{k+1}{k}\right)$, which is the d-index of the breakeven state, the ML model encounters underfitting.*

Corollary 2. *An ML model Θ is more likely to encounter underfitting in multiclass classification with an increase in the number of labels.*

Theorem 6 states that the range of multiclass d-index values falls between $(\log_2\frac{k+1}{k}, 2]$, and its proof is omitted for simplicity. This suggests that multiclass classification in ML models is more likely to encounter underfitting due to the lower d-index cutoff at the breakeven state. For instance, when $k = 3$, the breakeven state is characterized by d-index $d = 2\log_2\left(\frac{4}{3}\right)$ 0.8301. If the d-index is less than $d = 2\log_2\left(\frac{4}{3}\right)$, the model will encounter underfitting for 3-class classification. Similarly, for 4-class, 5-class, and 6-class classification, the breakeven d-indices will be 0.6439, 0.5261, and 0.4448, respectively. As the breakeven d-index cutoff decreases with an increase in the number of classes, it suggests that an ML model is more likely to encounter underfitting in multiclass classification with an increase in the number of labels.

2.7. Multiclass imbalanced point d-index estimation

We have the following d-index estimation at the imbalanced point in multiclass classification.

Theorem 7. ***Multiclass imbalanced point theorem.*** *Given an implicit prediction function* $\widehat{f}(x): x \to \Delta = \{1, 2 \cdots k\}$ *constructed from training data* $X_r = \{x_i, y_i\}_{i=1}^m, y_i \in \Delta$, *with the majority ratio* γ(*e.g.*, 50%), *under the ML model* Θ, *then at an imbalanced point, the ML model has the following d-index:*

$$d = \frac{1}{k}\left[\log_2\left[\prod_{i=1}^{k-1}(2-\gamma_i)\left(\frac{3}{2}\right)\right] + \log_2\left(\frac{3}{2}(1+\gamma)\right)\right] \tag{19}$$

Proof. Without loss of generality, we define the label ratio $\gamma_j = \frac{|\{x_i : y_i = j\}|}{\sum_{i=1}^{k}|\{x_j : y_j = i\}|}$ for class $j = 1, 2 \cdots k$. We also assume the label ratios of the total k classes follow the relationship: $0 < \gamma_1 \le \gamma_2 \le \cdots < \gamma_k < 1$, where the majority count ratio $\gamma = \gamma_k$.

At the imbalanced point, all the first *k-1* classes will be recognized as the majority type, denoted as $i \to k$ where i ranges from *1* to *k-1*. If we assume there are m total samples in the classification, then the True Positive (TP) and False Positive (FP) values for class i will be 0, represented as $tp_i = fp_i = 0$.

The False Negative (FP) value for class i ($i \neq k$) can be calculated as $fn_i = m\gamma_i$, because all samples belonging to class i, which is $m\gamma_i$, are falsely classified as the majority type k. Similarly, the True Negative (TN) for class i can be calculated as $tn_i = m - m\gamma_i$, because all samples not belonging to class i, which is $m - m\gamma_i$, are correctly classified as $\neg i$, which is the majority type k. Therefore, we have the confusion matrix C_i for class $i, i \neq k$,

$$C_i = \begin{pmatrix} tp_i & fp_i \\ tn_i & fn_i \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ m(1-\gamma_i) & m\gamma_i \end{pmatrix} \tag{20}$$

Then, we calculate the local accuracy, local sensitivity, and local specificity for class i as follows. Since there are no True Positives for that class, and all its samples are falsely classified as the majority class k. The local accuracy for class i only depends on the False Negative rate, which is the proportion of samples from class i that are classified as k. Thus local accuracy $\alpha_i = \frac{m(1-\gamma_i)+0}{m} = 1-\gamma_i$; local sensitivity: $s_i = \frac{0}{0+m(1-\gamma_i)} = 0$; and local specificity $p_i = \frac{m(1-\gamma_i)}{m(1-\gamma_i)+0} = 1$. Finally, we calculate the local d-index for class i as

$$d_i = \log_2(2-\gamma_i) + \log_2\left(\frac{3}{2}\right) \tag{21}$$

Similarly, when $i = k$, we have the confusion matrix C_k for class k :

$$C_k = \begin{pmatrix} tp_k & fp_k \\ tn_k & fn_k \end{pmatrix} = \begin{pmatrix} m\gamma_k & m(1-\gamma_k) \\ 0 & 0 \end{pmatrix} \tag{22}$$

We then calculate the local accuracy, local sensitivity, local specificity, and local d-index for class k as $\alpha_k = \frac{m\gamma_k+0}{m} = \gamma_k$, $s_k = \frac{m\gamma_k}{m\gamma_k} = 1$, $p_k = \frac{0}{0+m(1-\gamma_k)} = 0$, and $d_k = \log_2(1+\gamma_k) + \log_2\left(\frac{3}{2}\right)$ correspondingly.

Finally, we calculate the expected value of the local d-index values for all classes to get the final d-index that assesses the overall quality of the ML model's learning performance across all classes.

$$d = \frac{1}{k}\left[\sum_{i=1}^{k-1}\left(\log_2(2-\gamma_i) + \log_2\left(\frac{3}{2}\right)\right) + \log_2(1+\gamma_k) + \log_2\left(\frac{3}{2}\right)\right] \tag{23}$$

After simplification, we have $d = \left[\log_2\left[\prod_{i=1}^{k-1}(2-\gamma_i)\left(\frac{3}{2}\right)\right] + \log_2\left(\frac{3}{2}(1+\gamma)\right)\right]$, where the majority ratio $\gamma = \gamma_k$.

The theorem suggests that the d-index in multiclass classification, at the point of imbalance, will increase as the majority ratio γ increases. For instance, in a 3-class classification, the d-index at the imbalanced point is 1.3182 when $\gamma_1 = \gamma_2 = 0.25$, and $\gamma_3 = 0.5$. However, it reaches 1.48 when $\gamma_1 = \gamma_2 = 0.1$, and $\gamma_3 = 0.8$. It means the latter has a higher degree of overfitting at the imbalanced point.

It is worth noting that the majority ratio γ does not have to be very large (e.g., >50 %) to generate an imbalanced point in multiclass classification, but it should large enough compared to $\frac{1}{k}$ for a k-class classification. For example, in a 3-class classification problem, the imbalanced point can be generated under a majority ratio $\gamma = 0.49$ and other two minority ratios are about 0.24 and 0.27. Additionally, it is possible to generate more than one imbalanced point in a multiclass classification problem if some classes have a considerably larger label ratio than others.

The following corollary highlights the value of the d-index in the extremely imbalanced case of multiclass classification, where the majority ratio γ approaches 1, and the minority ratios $\gamma_i \to 0$ for $i = 0, 1, 2 \cdots k$. The corollary shows that the d-index value in such scenarios is consistent with the previous binary case.

Corollary 3. ***D-index in extremely imbalanced cases in multiclass classification.*** *The d-index in the extremely imbalanced cases in multiclass classification, where the majority ratio* $\gamma \to 1$ *and other minority ratios* $\gamma_i \to 0$, *for* $i = 0, 1, 2 \cdots k$, *approaches* $\log_2 3 = 1.5850$.

3. Results

3.1. Data

In this section, we showcase the superiority of the d-index over traditional classification metrics in providing explainable assessments for machine learning (ML). To demonstrate this, we utilize four imbalanced datasets from various domains, including natural language processing (NLP), FinTech, business, and medicine. The datasets, available at https://github.com/hank08819/DINDEX, were collected by the first author and have not been explored in any previous works. Table 1 illustrates the details of the four datasets, where the majority ratio refers to the ratio of the entries of the majority class with the most counts relative to the total number of entries in the dataset. The parameters n and p denote the number of observations and features, respectively. Specifically, we highlight the advantages of the d-index in robust model selection, sensitive monitoring of imbalanced learning, and detection of learning singularity. Furthermore, we illustrate how to differentiate learning performance under the same d-index values for support vector machines (SVM) for the simulated credit risk data.

3.2. Robust model selection using d-index

We showcase how the d-index can effectively enhance model selection by providing a comprehensive and explainable assessment of learning performance. To demonstrate this, we employ the imbalanced NLP dataset IB-EMODB, derived from the benchmark German emotional dataset EMODB used in speech emotion recognition [9]. The original EMODB dataset contains 535 sentences (audio files) spanning seven distinct emotion categories. The subset we utilize, IB-EMODB, includes 300 spoken sentences that are grouped into four emotional categories: anger (A), boredom (B), disgust (D), and fear (F). Of these, the anger (A) category is the most prevalent, constituting 42.33 % of the total

Table 1
Four datasets.

Dataset	(n,p)	Majority ratio	Classes	Field
IB-EMODB	(300,54)	42.33 %	4	NLP
Credit risk	(150,000,11)	93.05 %	2	Fintech
Simulated credit risk	(1670,6)	92.22 %	2	Business
Ovarian	(266,20531)	98.50 %	2	Medicine

sentences. Audio files file is characterized by 54 features obtained through Mel Frequency Cepstral Coefficients (MFCC) and other spectral feature extraction techniques. More feature extraction details can be found in the supplemental materials. Table 2 presents the distribution of the sentences across the various emotions: anger (A) at 42.33 %, boredom (B) at 27 %, disgust (D) at 7.67 %, and fear (F) at 23 %.

Fig. 1 illustrates the t-SNE (t-distributed Stochastic Neighbor Embedding) visualization of this multiclass dataset, revealing that the various emotions are relatively well separated [29]. Nearly every class forms its own distinct, well-bounded local clusters in the t-SNE embedding space, despite some scattering observed among these clusters. Notably, only a few samples from different classes are intertwined. This visualization showcases the good separability of the dataset, suggesting that ML models hold the potential to deliver reasonable, if not excellent, performance on it.

To demonstrate the superiority of the d-index in model selection, we compare the learning performance of six widely used ML models on this dataset. For this purpose, we partition the dataset into 80 % training and 20 % testing data for each model. The ML models used in this comparison include an SVM with a Gaussian kernel, random forests (RF) with 500 *'gini'*-based trees capped at a depth of 20; extremely randomized trees (ET) with 500 non-bootstrapped trees also limited to depth 20; deep neural networks (DNN) with L_2 regularization ($\alpha = 0.0001$), having hidden layers of 100, 50, and 25 neurons; linear discriminant analysis (LDA) employing *'svd'* solver with no shrinkage; and Gaussian-distributed NB [30–35]. To assess the learning performance of these models, we calculate the d-index as well as classic measures such as accuracy, sensitivity, specificity, precision, and negative prediction ratio (NPR) for this multiclass dataset.

The d-index provides a more comprehensive and explainable assessment of learning performance than traditional measures, leading to more accurate model evaluation. Table 3 presents a comparison of the performance of the six ML models based on both the d-index and traditional classification metrics. Using traditional measures, it can be challenging to evaluate the models' performance in an interpretable manner. For example, it is unclear whether SVM outperforms DNN and LDA or vice versa, as SVM has higher accuracy and precision, while DNN and LDA have better sensitivities and nearly equivalent specificities compared to SVM. However, the d-index comparison resolves this issue by demonstrating that DNN would slightly outperform SVM and LDA, based on their d-index values: 1.9078 (DNN) > 1.9050 (SVM) > 1.9015 (LDA). Similarly, the d-index of NB shows it outperforms ET and RF, despite ET having better accuracy than RF and NB. Furthermore, based on the d-index, it is evident that DNN is the best model for this NLP dataset. Therefore, the d-index offers a more straightforward and comprehensive evaluation in model selection, owing to its good interpretability.

Fig. 2 provides a visual representation of the comparison between the d-index and traditional classification measures for multiclass classification. The left plot compares the performance of the ML models using the 5 traditional classification measures. However, it can be challenging to evaluate the performance of the models using individual measures like accuracy or all possible measures. This is because individual measures do not fully reflect all aspects of learning, and combining them can lead to inconsistent evaluation of the models.

On the other hand, the right plot of Fig. 2 evaluates the performance of the models using their d-index values. It presents the performance of the ML models more clearly based on their d-index values in model selection, where DNN > SVM > LDA > NB > ET > RF. It avoids the possible bias from the accuracy measure and confusion that may arise when using all the traditional classification measures.

Table 2
The IB-EMODB dataset information.

Emotion type	The number of observations
Anger (A)	127
Boredom (B)	81
Disgust (D)	23
Fear (F)	69

3.3. Monitoring imbalanced learning using d-index

We use two imbalanced credit risk datasets to demonstrate how the d-index can be employed to monitor the behavior of ML models in imbalanced learning scenarios. The first dataset is a large-scale credit risk dataset, privately collected from small businesses. The second dataset, referred to as "simulated credit risk data," is a small credit risk dataset derived from a classic simulated credit risk dataset [36].

The first imbalanced credit risk dataset comprises of $n = 150{,}000$ credit records across $p = 11$ variables obtained from small businesses with no more than 20 employees. To obtain a clean dataset, we remove missing data which results in $n = 120{,}269$ observations. Out of these, 111,912 observations correspond to non-delinquency ('good credit') and 8,357 to delinquency ('bad credit') samples. The majority type ratio for this dataset is $\gamma = 93.05$ %. Table 4 presents all the variables including 10 general variables and one dependent variable 'delinquency' indicating the credit risk status.

The t-SNE visualization in Fig. 3(a) reveals the imbalanced nature of the delinquency and non-delinquency data, highlighting the risk of the majority data potentially dominating the learning process. Additionally, the variable visualization demonstrates that the two groups of data follow different probability distributions. Specifically, Fig. 3(b) presents a violin plot of the 10 variables in relation to delinquency and non-delinquency, with data being subjected to a log transformation. The plot indicates that each variable exhibits distinct distributions in terms of delinquency and non-delinquency. For example, the delinquency samples have a higher median revolving credit percentage than the non-delinquency ones, although the latter have more large outliers in revolving credit percentages. Similarly, the non-delinquency samples have a greater number of outliers in capital reservations and monthly income. Additionally, the non-delinquency type has a substantially smaller number of entries with late payments of <=60/90 days or longer.

We employed 4 ML models: *k*-NN, random forests (RF), gradient boosting (GB), extremely randomized tress (ET), and 4 deep learning models: deep neural networks (DNN), convolution neural networks (CNN), long short-term memory (LSTM), and transformer to handle this large imbalanced dataset [37–41]. The k-NN employs 5 neighbors with uniform weights, using the Euclidean distance and an auto-selected search algorithm. The GB operates with 100 trees, a depth of 3, a 0.1 learning rate, and employs the Friedman mean squared error for decisions. The CNN features two convolution layers of 128 and 64 nodes, followed by a pooling layer, flatten layer, and a dense layer with 20 % dropout, using *'relu'* and *'softmax'* activations. The LSTM has 5 sets of paired LSTM and dense layers with 64 and 128 nodes, accompanied by a flatten layer and 35 % dropout. The transformer integrates two embedding layers, two transformer blocks with multi-head attentions, two flatten layers, and two feedforward layers. All deep models adopt cross-entropy as their loss function. The RF, ET, and DNN maintain earlier settings. To evaluate their performance, we partitioned the dataset into 80 % for training and 20 % for testing purposes. Table 5 compares d-index and classic classification metrics of different learning models on the dataset.

The d-index values of the learning models demonstrate superior modeling capabilities compared to the classic metrics for imbalanced learning. While accuracy, specificity, and NPR metrics indicate good classification performance for all models, precision and sensitivity suggest mediocre or poor performance. However, the d-index values suggest that all models, except for LSTM, generate AIPs, as they are close to the imbalance point d-index $\log_2\left(\frac{3(1+0.9305)}{2}\right) = 1.5339$, as stated in Theo-

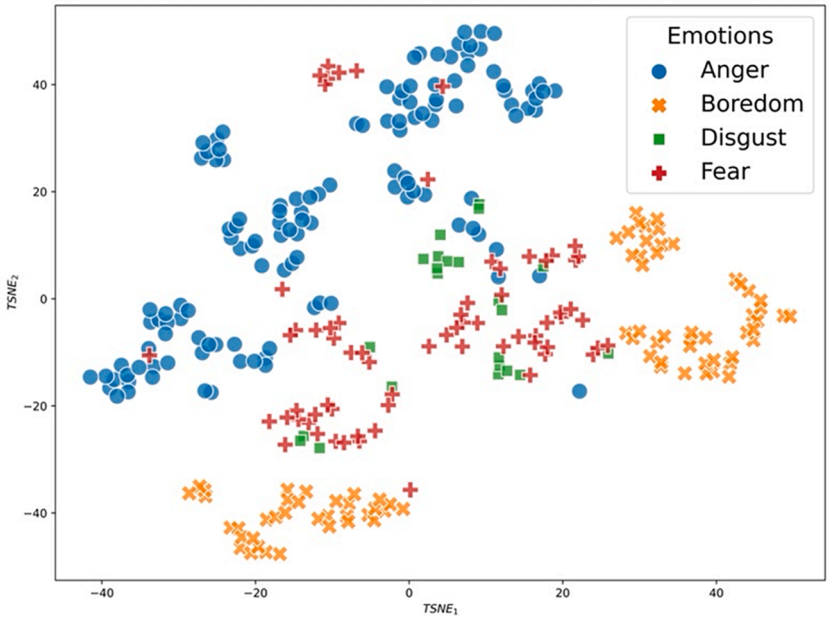


Fig. 1. t-SNE visualization of the IB-EMODB dataset depicting four types of emotions" 'Anger', 'Boredom', 'Disgust', and 'Fear'. The samples corresponding to the 'Anger' emotion display a relatively more concentrated distribution compared to the other three emotions.

Table 3
The d-index values and traditional measures on the multiclass dataset.

Measures\Models	SVM	LDA	DNN	RF	ET	NB
D-Index	1.9050	1.9015	1.9078	1.8033	1.8447	1.8467
Accuracy	0.95	0.9167	0.9333	0.8833	0.9	0.8833
Sensitivity	0.8636	0.9242	0.9034	0.7576	0.8201	0.8598
Specificity	0.9830	0.9748	0.9792	0.9594	0.9646	0.9602
Precision	0.9534	0.8576	0.9375	0.9	0.9131	0.8490
NPR	0.9849	0.9711	0.9767	0.9630	0.9672	0.9590

rem 2. This reveals that the models generally fail to learn because only a very small portion of minority samples are correctly predicted, as indicated by their low sensitivities. Table 4 compares the d-index and classic classification metrics of different learning models on the dataset, further highlighting the superior performance of d-index values.

For instance, the sensitivities of *k*-NN and ET are extremely low, at 0.0473 and 0.0119 respectively, indicating that they only correctly classify a mere 4.73 % and 1.19 % of minority samples, while misclassifying the remaining 95.27 % and 98.81 % of minority samples as the majority type. In contrast, both k-NN and ET correctly predict 99.72 % and 99.96 % of majority samples. This suggests that these models have a high tendency to recognize only the majority type, making them unsuitable for imbalanced learning as they generate AIPs. Similarly, while RF, GB, DNN, CNN, and transformer have slightly better d-index values than *k*-NN and ET, they too fail at imbalanced learning. It is not surprising to see that the self-attention mechanism in transformers cannot contribute to performance enhancement. While self-attention can enhance the model's ability to identify complex patterns and relationships in the data, it doesn't inherently contribute much to addressing the data imbalance issue, because attention may not be synonymous with data representation.

On the other hand, the LSTM's d-index of 1.7544 suggests that it is significantly different from the d-index of the imbalanced point of 1.5339, indicating a somewhat acceptable performance in imbalanced learning even though it correctly classifies only 55.45 % of the minority samples. However, it would be nearly impossible to distinguish the models' different behaviors using only their accuracy values, which are very close to the majority ratio: 93.05 %. Similarly, the weighted F1 score of 95 % obtained from this dataset falsely suggests that the

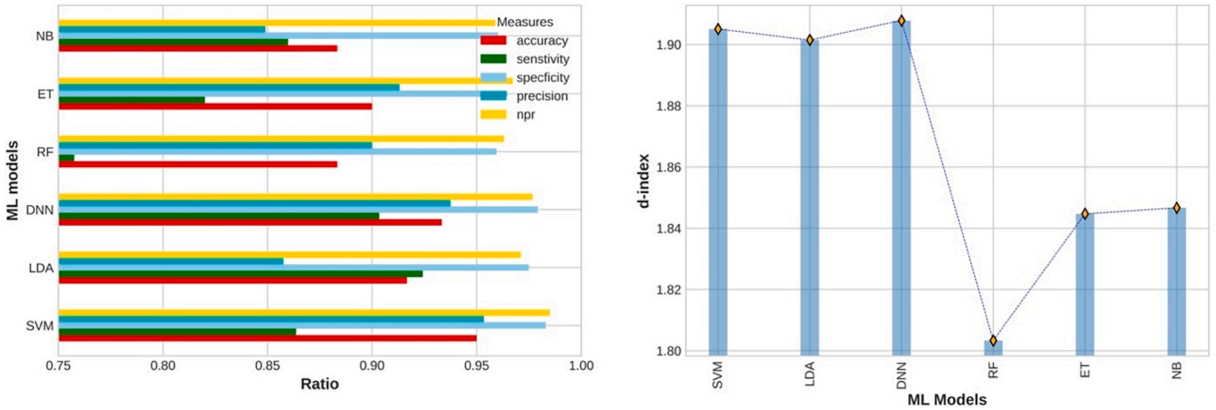


Fig. 2. A comparative analysis of six ML models' performance on the NLP multiclass dataset, utilizing both traditional classification measures (left plot) and their associated d-index values (right plot). The d-index offers a clearer and more interpretable assessment for model selection. As depicted, the DNN model surpasses others in both evaluation metrics, with the model hierarchy as follows: DNN > SVM > LDA > NB > ET > RF.

Table 4
Credit risk dataset variables.

Variable	Descriptions
Revolving Credit Percentage	The percentage of revolving credit over the total credit limits.
Capital Reserves	Money reserved in account to pay contingencies (e.g., mortgage)
Num Late 60:	The number of late payments within 60 days
Debt Ratio:	Borrower's debt to asset ratio.
Monthly Income ($):	The monthly income of borrower
Num Credit Lines ($1000):	The total amount of credit lines
Num Late Past 90:	The number of late payments above 90 days
Num Real Estate:	The number of real estates owned by borrower
Num Late 90:	The number of late payments within 90 days
Num Employees:	The number of employees of borrower
Delinquency	1 means bad credit standing (delinquency) and 0 good credit standing (non-delinquency)

learning performance is excellent. Furthermore, the MCC value obtained from this dataset is 0.25; while this indicates a performance that is better than random chance, it falls significantly short of excellence, demonstrating somewhat limited expressive interpretative power compared to the d-index. Thus, the proposed d-index shows good capability in monitoring learning models' behavior in imbalanced learning, offering a more accurate interpretation of imbalanced learning compared to classic classification metrics.

Fig. 4 further illustrates the classification report of the LSTM and the accuracy and loss plots for both the training and test data over the initial 20 epochs. The data presented in the two subplots suggest that an LSTM learning performance benchmarked at a d-index of 1.7544 is acceptable, albeit the accuracy remains proximate to the majority ratio. Subplot (a) delineates the classification results, while subplot (b) exhibits the evolution of accuracy and loss metrics for the training and test datasets throughout the first 20 epochs. One potential reason for the superior results on this imbalanced dataset is the LSTM's capability to capture temporal dependencies in the data, facilitating a deeper understanding of underlying patterns not immediately apparent when analyzing individual data points in isolation.

3.4. Detecting different imbalanced learning behaviors for linearly separable data

While imbalanced points or AIPs are common in many imbalanced learning problems, their presence is not guaranteed in all cases. The presence of AIPs or imbalanced points depends on various factors, such as the ML models being used, the parameters set, and the data itself. Interestingly, some imbalanced learning problems may be linearly separable with one ML model but not with another, due to the presence of AIPs, even if data is theoretically linearly sparable. Traditional classification metrics may not always be effective at detecting imbalanced points, AIPs or other imbalanced learning behaviors for such data. However, the d-index is a metric that can detect these behaviors with sensitivity.

To illustrate this, we turn to the simulated credit risk dataset, characterized by linearly separable data, utilizing the SVM as delineated in [28]. Our preference for SVM over other deep learning models is grounded in its deterministic and transparent nature, which guarantees reproducible learning results — a crucial asset in analyzing imbalanced learning behaviors. Although there is a minor risk of non-deterministic results with SVM in the rare instances of ties, primarily when data

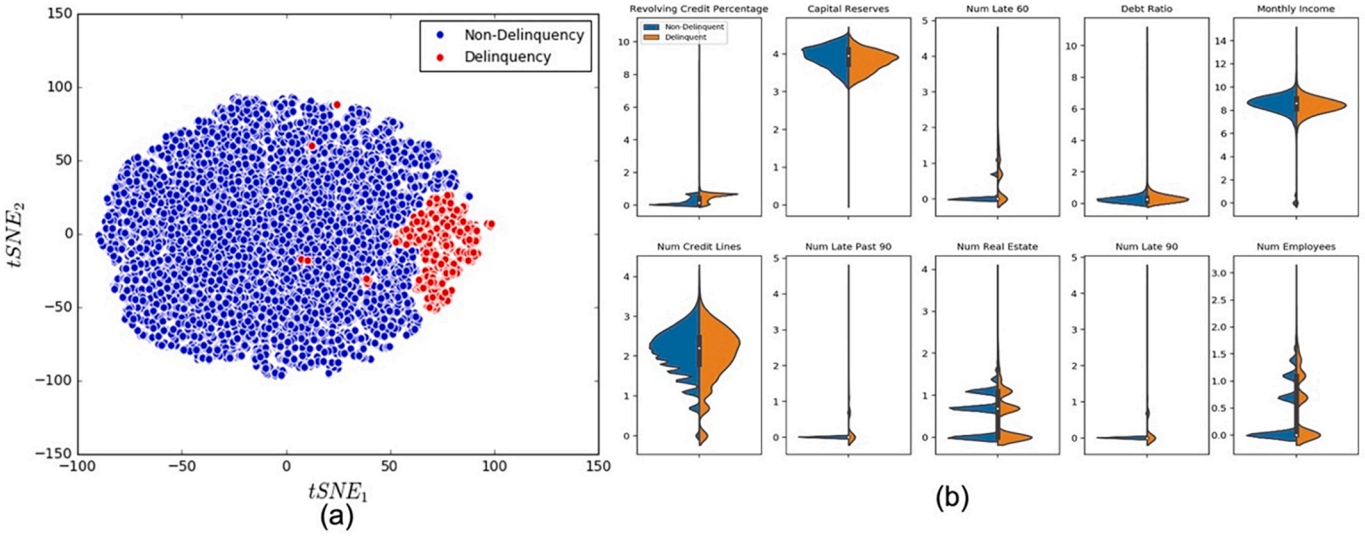


Fig. 3. The visualizations of the credit risk data and variables. Fig. 3(a) illustrates the t-SNE visualization of the imbalanced credit risk dataset, where label information is particularly used in t-SNE manifold learning process for the sake of imbalance visualization. Fig. 3(b) shows the violin plots of 10 independent variables after log transformation with respect to the delinquency and non-delinquency types. All variables demonstrate different probability distributions with respect to the delinquency and non-delinquency.

Table 5
The d-index and classic measures of the credit risk dataset under different ML models.

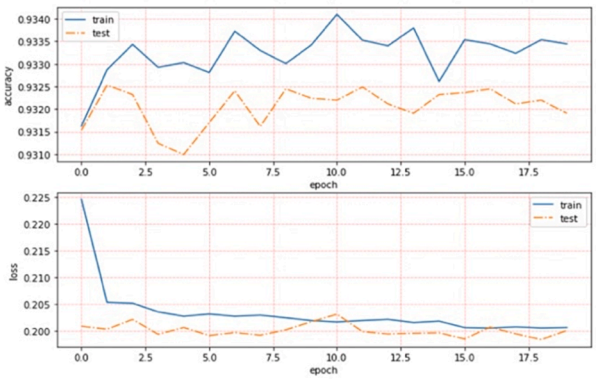
Measures\Models	*k*-NN	RF	GB	ET	DNN	CNN	LSTM	Transformer
D-index	1.5555	1.6108	1.6026	1.5395	1.6134	1.6049	1.7554	1.6149
Accuracy	0.9309	0.9319	0.9336	0.9306	0.9324	0.9309	0.9319	0.9318
Sensitivity	0.0473	0.1732	0.1485	0.0119	0.1779	0.1620	0.5545	0.1817
Specificity	0.9972	0.9888	0.9926	0.9996	0.9890	0.9886	0.9407	0.9892
Precision	0.5587	0.5369	0.5994	0.6667	0.5587	0.5158	0.1796	0.5634
NPR	0.9331	0.9410	0.9395	0.9310	0.9413	0.9402	0.9890	0.9404

```
Accuracy is 0.931903
Sensitivity(Recall) is 0.554545
Specificity is 0.940733
Precision(PPR) is 0.179623
NPR(negative prediction ratio) is 0.989041
Diagnostic index is 1.755430
              precision    recall  f1-score   support

           0       0.99      0.94      0.96     23504
           1       0.18      0.55      0.27       550

    accuracy                           0.93     24054
   macro avg       0.58      0.75      0.62     24054
weighted avg       0.97      0.93      0.95     24054
```

(a)



(b)

Fig. 4. (a) illustrates the LSTM classification on the imbalanced credit risk dataset, where 55.45% minority samples and 98.90% majority samples are correctly predicted. Fig. 4 (b) shows the accuracy and loss plots of the training and test data of the LSTM model during the first 20 epochs.

points align perfectly with the decision boundary, the likelihood remains low, particularly with linearly separable datasets. In contrast, deep learning models frequently encounter reproducibility issues, largely stemming from the widespread utilization of Stochastic Gradient Descent (SGD) for loss optimization, introducing an inherent randomness during weight updates, coupled with non-deterministic processes inherent in GPU operations, and other factors.

3.4.1. *Detect imbalanced points using d-index*

The second credit risk dataset is a simulated dataset to analyze the credit risk rankings of 1670 businesses from 12 industries. There are 1540 and 130 businesses across 6 variables ranked as good and bad credits respectively. The majority ratio of this dataset is $\gamma = 92.22\%$. The six variables include Working capital / Total Assets (WC/TA), Retained Earnings / Total Assets (RE/TA), Earnings Before Interests and Taxes / Total Assets (EBIT/TA), Market Value of Equity / Book Value of Total Debt (MVE/BVTD), Sales / Total Assets (S/TA), and Industry sector labels from 1 to 12 (Industry).

Fig. 5 presents the t-SNE visualization of the dataset and the correlation matrix visualization of all the variables [29]. The t-SNE plot shows that the dataset is linearly separable through orthogonal separation. Moreover, the strong correlations between variables suggest that this dataset could achieve good learning performance even though it is imbalanced with a 92 % majority ratio. However, we have also discovered that SVM can completely lose its learning capabilities under certain special kernels, such as the Sigmoid kernel, due to the creation of imbalanced points or AIPs. This indicates that imbalanced learning can exhibit either linear separability or the generation of imbalanced points under different parameter settings for an ML model like SVM.

In our implementation, we use support vector machines (SVMs) with PCA dimension reduction to predict credit statuses and achieve good separations. This approach not only provides accurate predictions but also enables effective visualization of the learning process through the d-index. To train and test our model, we partition the data into 70 % for training and 30 % for testing. We utilize four different kernels in our SVM implementation: *'linear'* $k(x,y) = x^T y$, *'Gaussian'* $k(x,y) = e^{-\eta\|x-y\|^2}$, *'polynomial'* $k(x,y) = (\eta x^T y + 1)^3$, and *'Sigmoid'* $k(x,y) = \tanh(\eta x^T y + 1)$. The parameter η is set as $1/q$ where q is the number of features of the dataset [30].

Table 6 presents a comparison of the four kernels used in SVM with respect to the d-index and classic measures. Notably, the Sigmoid kernel exhibits a d-index of 1.5064, which suggests the existence of an AIP or imbalanced point. This is due to its d-index being close to the value at the imbalanced point: $\log_2\left(\frac{3(1+0.9222)}{2}\right) = 1.5227$. Furthermore, the results indicate that the F1 score is biased because it achieves a high score of 0.9464 despite all the minority samples being wrongly classified as the majority, as both the specificity and NPR values are 0 %.

Interestingly, it is almost impossible to detect when SVM loses its learning capability under the *'Sigmoid'* kernel by relying solely on classic measures like accuracy (89.82 %), F1 score (0.9464), precision (0.9036), and sensitivity (0.9933). However, the d-index can easily

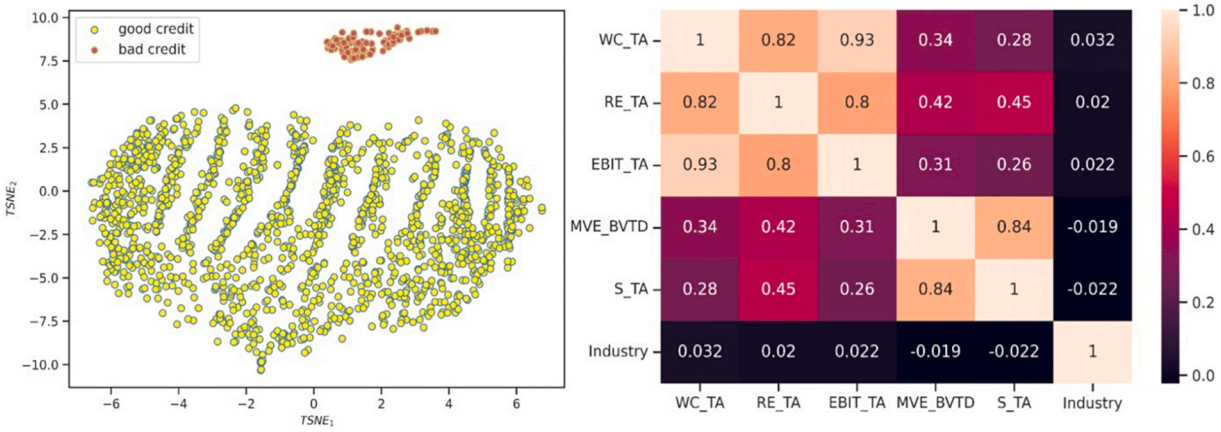


Fig. 5. The left plot presents the t-SNE visualization of the small credit risk dataset, wherein the two groups are clearly delineated as independent clusters. The right plot illustrates the correlation matrix for all the variables within the dataset, revealing substantial correlations between most of them.

Table 6
The d-index and classic measures of the small credit risk dataset under SVM.

Measures\kernels	*Linear*	*Gaussian*	*Polynomial*	*Sigmoid*
D-index	1.9978	1.9978	1.9955	1.5064
Accuracy	0.9980	0.9980	0.9960	0.8982
Sensitivity	0.9978	0.9978	0.9956	0.9933
Specificity	1.0	1.0	1.0	0.0
Precision	1.0	1.0	1.0	0.9036
NPR	0.9796	0.9796	0.9960	0.0
F1	0.9989	0.9989	0.9978	0.9464

signal the anomalous learning status. In contrast, the other three kernels achieve nearly perfect prediction ratios, as their d-index values are close to 2. This observation suggests that imbalanced learning does not always result in the imbalanced point or AIP. Instead, it may exhibit different learning behaviors under varying parameter settings of the given ML model.

3.4.2. *Distinguish learning performance under the same d-index*

While it is possible for two machine learning (ML) learning results to exhibit the same d-index value, this occurrence is often unlikely, especially when different parameter settings are used under the same ML model, such as SVM. However, if such a situation arises, how can we further evaluate ML performance under the same d-index values? To address this question, let us continue using the example of small credit risk prediction with SVM.

Table 6 demonstrates that the *'linear'*, *'Gaussian'*, and *'polynomial'* kernels demonstrate almost the same d-index values. Notably, the first two kernels have identical performance across all measures, including the d-index. In cases where the d-index values are the same under SVM, we can evaluate model performance by considering the number of support vectors. Support vectors refer to observations on the boundary of the optimal hyperplane built by the training data. A smaller number of support vectors suggests better scalability and generalization of the SVM model.

Fig. 6 visualizes the support vectors of the four kernels under SVM learning. The *'linear'* and *'polynomial'* kernels have fewer support vectors compared to the *'Gaussian'* and *'Sigmoid'* kernels. It is evident from the visualization that the small number of support vectors under the *'linear'* and *'polynomial'* kernels can almost perfectly separate the two groups of samples, but the former having a slightly larger d-index than the latter. Although the *'Gaussian'* kernel achieves the same level of learning performance, its large number of support vectors suggest that it requires more effort to achieve similar results compared to the *'linear'* and *'polynomial'* kernels. Such a high number of support vectors may lead to poor scalability and generalization in learning. Therefore, the *'linear'* kernel performs the best due to its d-index and smaller number of support vectors.

On the other hand, the southeastern plot of Fig. 6 illustrates the impact of the imbalanced point on the *'Sigmoid'* kernel, where almost all minority samples in the training data are incorrectly identified as support vectors. Consequently, SVM loses its ability to learn by misclassifying all minority samples in queries as the majority. Similar results occur when replacing PCA with t-SNE.

3.4.3. *Priori kernel selection*

Knowing which kernel will achieve a good d-index and the least number of support vectors before applying it to real SVM classification remains an unsolved problem in SVM learning [31–32]. Although we do not intend to provide a systematic answer to this question, we offer a case study solution from an interpretable assessment perspective. Since the learning capability of an SVM model relies on the representativeness of its kernel matrix $K \in \mathscr{R}^{n\times n}$, we believe that a kernel matrix with sparser eigenvalues would be more representative and able to generate

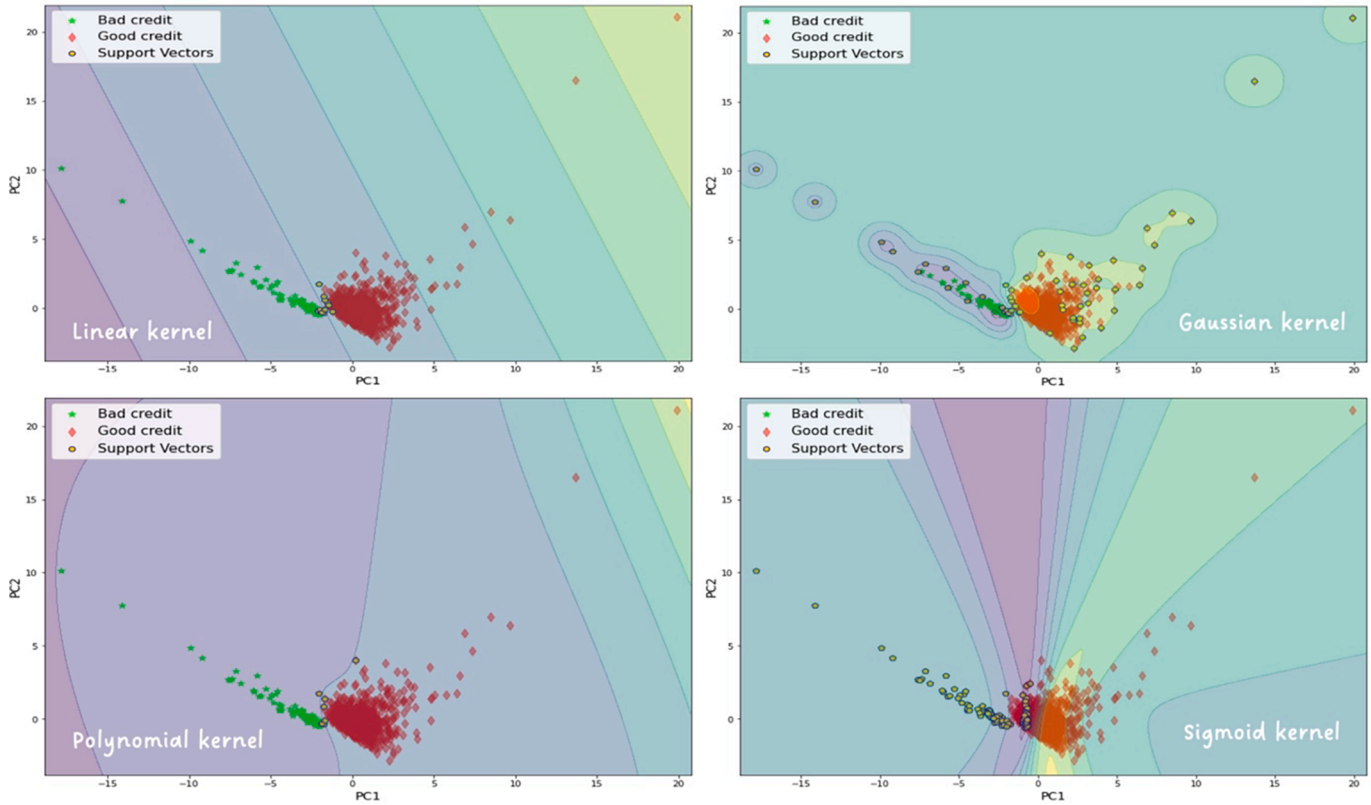


Fig. 6. Visualization of support vectors in SVM classification under *'linear'*, *'Gaussian'*, *'polynomial'*, and *'Sigmoid'* kernels. The *'linear'* and *'polynomial'* kernels have fewer support vectors compared to the *'Gaussian'* and *'Sigmoid'* kernels. Notably, the *'Sigmoid'* kernel generated the imbalanced point in classification, causing SVM to lose learning capabilities by incorrectly identifying almost all minority samples in the training as support vectors.

fewer support vectors while achieving good d-index values. If the eigenvalues of the kernel matrix are sparse, it means that the data points in the feature space can be represented using only a small number of principal components. This implies that the data is low-dimensional, and the decision boundary can be defined using a relatively small number of support vectors. As a result, the SVM model can achieve good performance with fewer support vectors, reducing the computational complexity and memory requirements of the model.

The sparsity of the kernel matrix eigenvalues. To determine the sparsity of the eigenvalues in the kernel matrix of an SVM, we sort the top k (e.g., $k \geq 100$) eigenvalues in descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ and evaluate their sparsity using the parameter ε, which is typically set to a very small value such as $\varepsilon = 10^{-12}$. We define the sparsity of the eigenvalues of the kernel matrix K as $\rho(\varepsilon)$, given by the equation:

$$\rho(\varepsilon) = \left|\left\{\lambda : \lambda \leq \varepsilon, \lambda \in \cup_{i=1}^{k}\lambda_i, \forall x \in \mathscr{R}^n, Kx = \lambda x, \right\}\right|/k \tag{24}$$

Here, $\rho(\varepsilon)$ measures the proportion of eigenvalues that are less than or equal to ε, relative to the total number of selected top eigenvalues (e.g., $k = 100$). The higher the value of $\rho(\varepsilon)$, the sparser the eigenvalues in the kernel matrix, indicating that fewer support vectors are required to define the decision boundary of the SVM. The sparsity of the eigenvalues of the kernel matrix K is a good indicator of the quality of SVM learning. Therefore, we have the following proposition.

Proposition 1. *If the kernel matrix K of an SVM learning machine with sparser top eigenvalues than those of the other kernel matrices, then the SVM can produce better learning results with fewer support vectors.*

The sparsity analysis of eigenvalues for kernel matrices is closely related to the performance of SVM kernels. As depicted in Fig. 7, we compare the top 100 eigenvalues of the *'linear'*, *'Gaussian'*, *'polynomial'*, and *'Sigmoid'* kernels, where their sparsity values for $\varepsilon = 10^{-12}$ are 0.98, 0.0, 0.59, and 0.0, respectively. Among the four kernels, the *'linear'* kernel has the sparest eigenvalues, achieving the best d-index with the least number of support vectors. The *'polynomial'* kernel has the second sparsest eigenvalues and the second lowest number of support vectors, with a d-index value that is only slightly lower than that of the 'linear' kernel. However, the *'Gaussian'* and *'Sigmoid'* kernels have denser and larger eigenvalues compared to the other two, indicating that they require more support vectors to define the decision boundary of the SVM.

3.5. Detect learning singularity problems

The d-index not only facilitates effective model selection, monitors ML behaviors, and detects imbalanced points, but it also has the potential to identify learning singularity problems. Despite the lack of research on this important topic, identifying and addressing learning singularity problems is critical to both ML theory and practice as they widely exist in all AI and data science domains such as AI disease diagnosis in medicine. Successfully solving these problems has the potential to bring unprecedented impacts on ML theory, AI techniques and various data science applications. However, due to the limitations of existing ML theory, these problems are generally viewed as individual 'hard' nonlinear problems, rather than recognized as a systematic category of ML problems with distinct characteristics. This is mainly because they are not easily detectable. To address this issue, we propose a definition for learning singularity problems that takes into account their unique characteristics.

3.5.1. The learning singularity problem

A learning singularity problem can be also called a 'Non-Deterministic' Imbalanced learning (NDI) problem that cannot be solved by existing ML models or imbalanced learning handling methods (such as resampling) due to the generation of an imbalanced point or AIP. It remains unknown which ML methods can find a meaningful solution by avoiding the generation of the imbalanced point or AIP. A learning singularity must satisfy the following two conditions.

1. The imbalanced learning problem must fail almost all existing ML models as well as imbalanced learning handling methods (e.g., resampling) by unavoidably generating an imbalanced point or AIP.
2. It can achieve an acceptable or even a good result when the knowledge to be learned, such as the labels of the test samples, is fused in training.

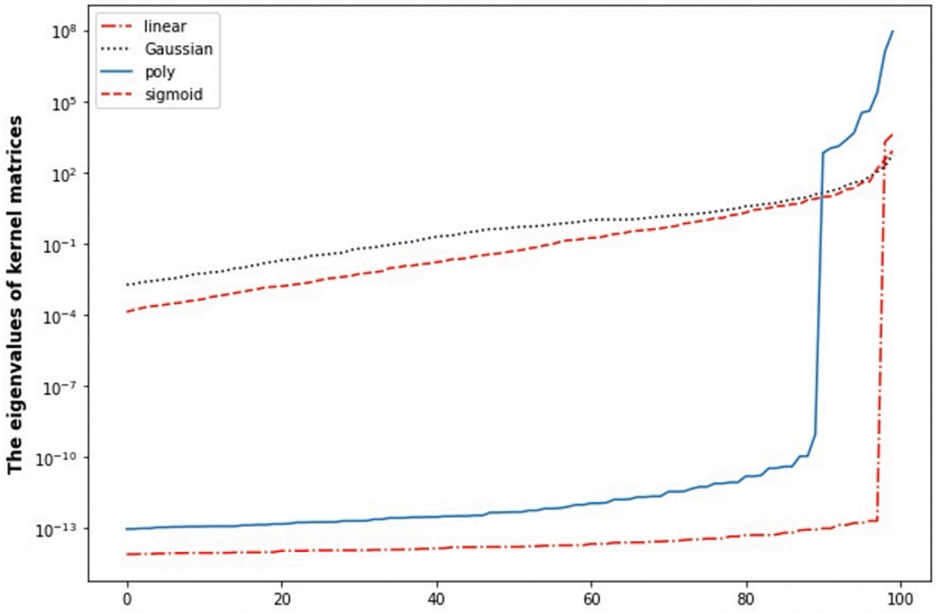


Fig. 7. Comparison of the top 100 eigenvalues of kernel matrices for the *'linear'*, *'Gaussian'*, *'polynomial'*, and *'Sigmoid'* kernels. The *'linear'* kernel exhibits the highest level of sparsity, with most entries close to zero. The *'polynomial'* kernel shows the second highest level of sparsity. On the other hand, the *'Gaussian'* and *'Sigmoid'* kernels have much denser and larger eigenvalues compared to the other two kernels.

It is important to note that there are various techniques available for fusing knowledge into the training process. One common and relatively simple approach, often misused by beginners in the ML field, is to conduct resampling (e.g., SMOTE) on the entire dataset rather than solely on the training data [42,43]. In other words, while there is a 'learning path' to verify the 'learnability' of a learning singularity problem, it is still unknown whether a meaningful solution exists since the problem typically fails existing methods by generating an imbalanced point or AIP. We provide the following formal description about the learning singular problem.

Learning singularity problem. Given training data $X_r = \{x_i, y_i\}_{i=1}^m$, $y_i \in \Delta = \{1, 2\cdots k\}$ in classification with the majority ratio γ, without loss of generality, we define the label ratio $\gamma_j = \frac{|\{x_i : y_i = j\}|}{\sum_{i=1}^{k} |\{x_j : y_j = i\}|}$ for class $j = 1, 2\cdots k$. We also assume the label ratios of the total k classes follow the relationship: $0 < \gamma_1 \leq \gamma_2 \leq \cdots < \gamma_k < 1$, where the majority ratio is from class k, i.e., $\gamma = \gamma_k > 1/k$. Then an imbalanced learning problem is called a learning singularity problem if and only if it satisfies the following conditions:

1) The prediction function $\widehat{f}(x|\Theta, X_r)$ constructed from any ML model Θ will have $\widehat{f}(x|\Theta, X_r) = k$ for each sample x in query with by generating an imbalanced point or AIP, i.e., $d \approx \log_2\left(\frac{3(1+\gamma)}{2}\right)$ if $k = 2$ or $d \approx \left[\log_2\left[\prod_{i=1}^{k-1}(2-\gamma_i)\left(\frac{3}{2}\right)\right] + \log_2\left(\frac{3}{2}(1+\gamma)\right)\right]$, if $k > 2$.
2) If the knowledge to be learned, which are the label information of the test data $X_t = \{x_k, y_k\}_{k=1}^n$, is fused into the training process, then $\exists$ an ML model Θ', and $\eta > 0$, such that the d-index of the model Θ' have

$$2\log_2\left(\frac{k+1}{k}\right) + \eta < d \leq 2 \tag{25}$$

Detecting a learning singularity problem can be challenging with traditional classification metrics. This difficulty arises because such metrics often lose interpretability in favor of providing a misleading assessment of machine learning performance. Consequently, they may not be able to accurately identify whether an ML model has reached an imbalanced point or an AIP state, which is an essential step addressing the learning singularity problem in both binary and multiclass classification problems.

Moreover, validating the imbalanced point or AIP generation for an input imbalanced data by trying all possible ML models and relevant imbalanced handling techniques can be computationally prohibitive [34]. It remains unknown which approach should be used to integrate the knowledge to be learned in the training process.

We tackle the problem of detecting learning singularities by using the d-index due to its good interpretability in detecting imbalanced points or AIPs. To avoid potential computing overhead, we select a set of representative machine learning (ML) models and imbalanced handling techniques, rather than including all of them. For example, the representative ML models include basic shallow learning methods (e.g., k-NN), kernel-based learning (SVM), ensemble learning (e.g., RF), and deep learning (e.g., CNN). Similarly, we employ classic resampling methods (e.g., ROS) to address data imbalance. We perform ROS resampling for both training and test data to fuse knowledge during the training process. Algorithm 1 presents our learning singularity problem detection approach using the d-index. Without loss of generality, we assume that the training and test data have the same majority ratio. If the majority ratios are different, we use the majority ratio of the test data in the d-index calculation.

Algorithm 1: Learning singularity problem detection

Input: Training data: $X_r = \{x_i, y_i\}_{i=1}^n, y_i \in \Delta = \{1, 2\cdots k\}$
Test data: $X_t = \{x_j', y_j'\}_{j=1}^m, y_j' \in \Delta = \{1, 2\cdots k\}$
The majority ratio is $\gamma > 1/k$ and the majority class is k
The label ratio γ_j for class $j = 1, 2, \cdots k-1$
Representative ML models: $\theta_1, \theta_2 \cdots \theta_N$
Imbalanced handling techniques $g_1, g_2, \cdots g_l$
Tolerance ε (default 0.20)
Offset η (default 0.50)
Output: Learning singularity problem status: LSP_{status}

$LSP_{status} \leftarrow T$
// representative ML models
for *each* e *in* $\theta_1, \theta_2 \cdots \theta_N$
$d_{list} \leftarrow ComputeLearningDIndex(X_r, e, X_t)$
// representative imbalanced handling techniques
for *each* g *in* $g_1, g_2, \cdots g_l$
$d^e_{list} \leftarrow ComputeLearningDIndex(X_r, g, X_t, e)$
for d *in* $d_{list} \bigcup d^e_{list}$
if $k == 2 \wedge |d - \log_2\left(\frac{3(1+\gamma)}{2}\right)| > \varepsilon$
$LSP_{status} \leftarrow F$
Return LSP_{status}
if $k > 2 \wedge \left|d - \left[\log_2\left[\prod_{i=1}^{k-1}(2-\gamma_i)\left(\frac{3}{2}\right)\right] + \log_2\left(\frac{3}{2}(1+\gamma)\right)\right]\right| < \varepsilon$
$LSP_{status} \leftarrow F$
Return LSP_{status}
for *each* e *in* $\theta_1, \theta_2 \cdots \theta_N$
$d_{fuse} \leftarrow ComputeDIndexUnderFuseknowledgeInTraining(X_r, e, X_t)$
for d *in* d_{fuse}
if $2\log_2\left(\frac{k+1}{k}\right) + \eta < d \leq 2$
Return LSP_{status}
$LSP_{status} \leftarrow F$
Return LSP_{status}

3.5.2. Hdi-data-based disease diagnosis

We use HDI-data-based disease diagnosis as an example to apply algorithm 1 to demonstrate learning singularity problem detection.

HDI (high-dimensional and imbalanced) data is a unique type of imbalanced data that frequently arises in the field of biomedical data science. Unlike traditional imbalanced data, HDI data is not only high-dimensional, but also extremely imbalanced due to limited data resources (such as rare disease subtypes) and acquisition limitations. A typical HDI dataset, for instance, may contain 100 positive samples and only 10 negative samples across a selection of 5,000 genes.

Ovarian data. The ovarian dataset comprises of RNA-seq data collected from TCGA by the first author, consisting of 4 solid ovarian tumors and 262 recurrent ovarian tumors across 20,531 genes. The majority ratio of the dataset is 98.50 % (262/266), indicating that it is an extreme HDI dataset with only 4 minority samples. In machine learning, resampling techniques such as SMOTE are commonly used to handle imbalanced data, especially to increase the quantity of minority samples.

Fig. 8 illustrates the PCA visualization of the Ovarian dataset before and after applying the SMOTE resampling procedure. The original dataset contains 4 solid ovarian tumors and 262 recurrent ovarian tumors, comprising a highly imbalanced dataset with only 4 minority samples. As shown in the figure, the minority samples are almost indistinguishable among the 262 majority samples. After applying SMOTE, the minority samples are oversampled based on certain rules, resulting in a significantly increased quantity of the minority samples. However, the minority samples' distributions become quite different from the majority samples due to the specific resampling process of SMOTE.

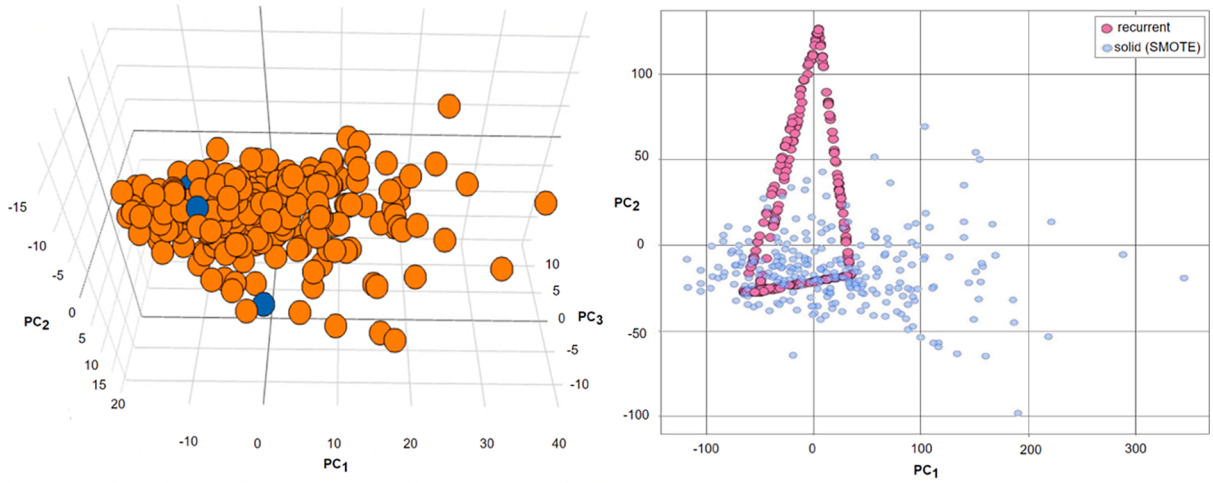


Fig 8. The left plot shows the PCA visualization of the Ovarian dataset, where the four minority samples are indistinguishable from the 262 majority samples. The right plot illustrates the effect of applying SMOTE resampling to the Ovarian dataset, resulting in a significant increase in the quantity of the minority samples (e.g., solid (SMOTE)). However, the distribution of the minority samples is noticeably different from that of the majority samples due to the specific resampling process of SMOTE.

To apply Algorithm 1 and determine whether the ovarian data classification is a learning singularity problem, we have chosen seven representative machine learning models from various categories, including shallow learning (e.g., *k*-NN), kernel-based learning (e.g., SVM with the *'linear'* kernel), ensemble learning (e.g., RF, GB, and ET), and deep learning (e.g., DNN and CNN) [35]. We partitioned the dataset into 70 % training data with 186 observations, including 2 minority samples, and 30 % test data with 80 samples, including the other 2 minority samples.

Table 7 presents the learning results of the seven selected ML models, evaluated using classic metrics and d-index. As expected, all the models produce the imbalanced point and achieve a d-index value of $\mathrm{d} = \log_2\left(\frac{3(1+\gamma)}{2}\right)_{\gamma=78/80} = 1.5668$. The accuracy, sensitivity, and specificity are 97.5 % (the majority ratio), 100 %, and 0 %, respectively. This indicates that all negative (positive) samples are misclassified (classified correctly), resulting in TN = FN = 0. As a consequence, NPR is undefined (nan) because NPR = TN/(TN + FN). Importantly, this finding is consistent across different data partitions.

We have also observed the same imbalanced point with the d-index of 1.5668 when using two commonly used imbalanced data handling techniques, namely SMOTE and random oversampling (ROS), to generate additional minority samples before training the ML models. This finding suggests that while these techniques may increase the number of minority samples, they cannot eliminate the risk of generating an imbalanced point in the classification results.

We further fuse the knowledge to be learned in training by conducting resampling for the whole data before the train-test partition. We find the ML models: GB achieves good performance by attaining d-index 1.9863 under SMOTE, and RF achieves d-index 2 under SMOTE and ROS [33–34]. These findings suggest that the classification of HDI ovarian data is a learning singularity problem according to Algorithm 1.

3.6. Imbalanced points under transformer models

It is worthy to examine the generation of imbalanced points under deep learning models because almost all deep learning models seem to encounter imbalanced points in both credit risk data and ovarian data scenarios. We employ the state-of-the-art transformer model here as a representative due to its efficiency and good scalability. This model is a kind of deep learning architecture initially crafted for natural language processing but has since found successful applications in a diverse array of tasks well beyond sequential data. In our exploration, we discovered that the transformer model encounters imbalanced points or AIPs as readily as other deep learning models when applied to these two datasets. Moreover, we found that the d-index can offer a correction to the potential biased perspectives that classic training and testing loss analyses might provide when imbalanced points are present.

3.6.1. Transformer

Leveraging self-attention mechanisms to parallel process inputs, the transformer model manages to capture complex patterns and relationships in different data types besides language data, showcasing versatility and high performance across various machine learning tasks [41]. We describe the transformer model for classification in brief for the convenience of description.

Given input data $X \in \Re^{n\times p}$, where n is the number of samples and p is the number of features, a transformer model uses a self-attention mechanism to assess the importance of different data components and assign attention weights accordingly. It organizes data with queries (Q), keys (K), and values (V) through linear transformations: $Q = XW_Q$, $K = XW_K$, $V = XW_V$, where matrices W_Q, W_K, W_V are the weight matrices to be learned in training. Q represents different aspects of the data transformed through W_Q, K works with Q to determine the relationships between different components of data, and V signifies the content of the

Table 7
The d-index and classic measures of the ovarian dataset.

Measures\Models	*k*-NN	SVM	RF	GB	ET	DNN	CNN
D-index	1.5668	1.5668	1.5668	1.5668	1.5668	1.5668	1.5668
Accuracy	0.9750	0.9750	0.9750	0.9750	0.9750	0.9750	0.9750
Sensitivity	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Specificity	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Precision	0.9750	0.9750	0.9750	0.9750	0.9750	0.9750	0.9750
NPR	nan	nan	nan	nan	nan	nan	nan

data being focused upon to create new representations based on the attention weights. The self-attention scores are computed using Q and K: $score(Q,K)=\frac{QK^T}{\sqrt{d_k}}$, where d_k is the dimensionality of keys. Then the scores are normalized by the *softmax* function: $f_{softmax}(x_j)=\frac{e^{x_j}}{\sum_{j=1}^{k}e^{x_j}}$ to produce attention weights: $weights_{attention}=f_{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$. Finally, the attention weights apply to V to produce the self-attention output of data, encapsulated mathematically as

$$X_{attention}\leftarrow Attention(Q,K,V)=f_{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{26}$$

Notably, the self-attention mechanism can be implemented using multi-head attention, utilizing multiple sets of Q, K, and V matrices in parallel to focus on different parts of the input.

Subsequently, the output from the self-attention layer is passed through a feedforward neural network (FFNN), represented mathematically as

$$Z_{ffn}=f_{relu}\left(X_{attention}W_{ffn}^{(1)}+b_{ffn}^{(1)}\right)W_{ffn}^{(2)}+b_{ffn}^{(2)} \tag{27}$$

where $W_{ffn}^{(i)}, b_{ffn}^{(i)}$, for $i=1,2$, are the weight matrices and bias vectors in the i^{th} layer of the FFNN, and $f_{relu}=\max(x,0)$ is the ReLu activation function. Finally, the classification probabilities are calculated using the *softmax* function applied to the FFNN output: as $P_c\leftarrow f_{softmax}(Z_{ffn})$.

3.6.2. Imbalanced points or AIPs under transformer

Our transformer model implementation incorporates two embedding layers, two transformer blocks with multi-head attentions and two flatten layers, alongside two dense layers, while partitioning each dataset into 80 % for training and 20 % for testing. The model is trained over $n_e=100$ epochs on each dataset. We implement a cross-entropy loss function as

$$\mathbf{L}_{train}^{(j)}(\theta)=-\frac{1}{n}\sum_{i=1}^{n}[y_i\log(p_{ij}+\epsilon)+(1-y_i)\log(1-(p_{ij}+\epsilon))] \tag{28}$$

where θ represents the hyperparameters of the transformer model to be optimized, $y_i\in[-1,1]$ is the true label of sample x_i during training, p_{ij} denotes the predicted probability of x_i sample at epoch $j\in\{1,2,\cdots n_e\}$, $\epsilon=10^{-10}$ is the tolerance parameter to prevent $\log(0)$, and n is the training dataset size. The model is trained over $n_e=100$ epochs on each dataset. Similarly, we have the loss function for testing data: $\mathbf{L}_{test}^{(j)}(\theta)=-\frac{1}{m}\sum_{i=1}^{m}[y_i\log(p_{ij}+\epsilon)+(1-y_i)\log(1-(p_{ij}+\epsilon))]$ for m testing samples $\mathbf{L}_{train}^{(j)}(\theta)=-\frac{1}{m}\sum_{i=1}^{m}[y_i\log(p_{ij}+\epsilon)+(1-y_i)\log(1-(p_{ij}+\epsilon))]$, and the rest of the parameters hold the same meanings as in the training loss function definition. The loss values $\{\mathbf{L}_{train}^{(j)}(\theta)\}_{j=1}^{n_e}$ and $\{\mathbf{L}_{test}^{(j)}(\theta)\}_{j=1}^{n_e}$ form the training and testing loss curves.

Fig. 9 compares the loss curves, accuracies, and d-index values of training and testing data of the ovarian and credit risk datasets across 100 epochs. The two subfigures on the left reveal a seemingly 'good performance' from both the training and testing loss curves, with notable reductions in loss and exhibiting similar patterns over time. This observation is further corroborated by the high cosine similarity between the curves, calculated as:

$$r_{cosine}=\frac{\sum_{j=1}^{n_e}\mathbf{L}_{train}^{(j)}(\theta)\times\mathbf{L}_{test}^{(j)}(\theta)}{\left(\sum_{j=1}^{n_e}\mathbf{L}_{train}^{(j)}(\theta)^2\right)^{1/2}\left(\sum_{j=1}^{n_e}\mathbf{L}_{test}^{(j)}(\theta)^2\right)^{1/2}} \tag{29}$$

Scoring 0.9282 for the ovarian dataset and an almost perfect 0.9996 for the credit risk dataset, the cosine similarity suggests a similar direction of movement across epochs for both curves. This is generally perceived as a positive indication of the model's ability to effectively generalize from the training data to the unseen testing data, avoiding the pitfall of overfitting.

However, a deeper examination reveals that the transformer model reaches the imbalanced point in the case of the ovarian dataset and an Acceptable Imbalanced Point (AIP) for the credit risk dataset. This phenomenon occurs because the d-index values for the training and testing datasets either equal or closely approximate the theoretical d-index values derived from the equation $\log_2\left(\frac{3(1+\gamma)}{2}\right)$, where γ is the majority ratio of the training or testing data.

For example, in the ovarian dataset, the d-index of the testing data is $\log_2\left(\frac{3(1+\gamma)}{2}\right)_{\gamma=53/54}=1.5715$ because they are 53 counts in the majority class among the total 54 testing samples. Similarly, the d-index of the training data is $\log_2\left(\frac{3(1+\gamma)}{2}\right)_{\gamma=209/212}=1.5747$. The right subfigure in the first row of Fig. 9 displays the d-indices, signaling a prominent overfitting scenario.

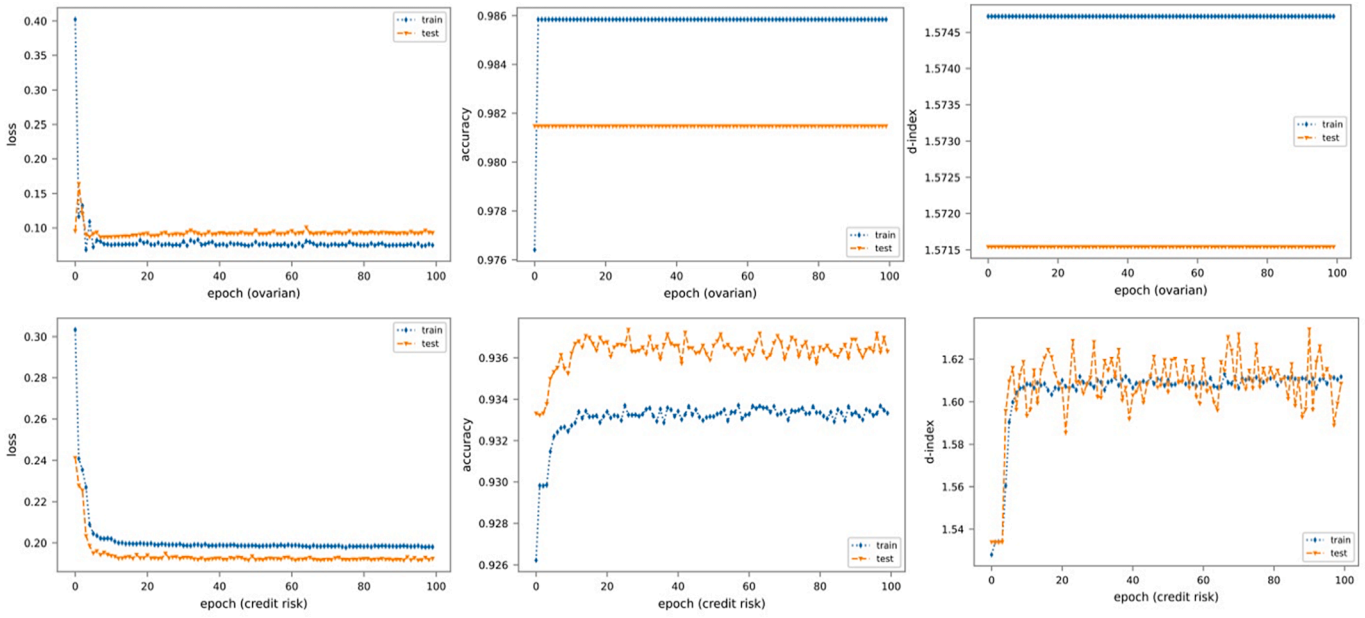


Fig. 9. The comparisons of the loss curves, accuracies, and d-index values of training and testing data of the ovarian and credit risk datasets across 100 epochs.

Besides validating the learning singularity problem from a transformer perspective, this finding overturns the optimistic view given by the training and testing loss curves, which implied a good generalization capability of the transformer model. It underscores the necessity of evaluating d-index values to attain a precise understanding of the model's performance. Additionally, the neighboring subfigure points out that the high accuracies of 98.58 % and 98.15 % for training and testing data, respectively, simply mirror the majority ratios of the datasets. Moreover, while an MCC score of 0 might vaguely hint at random predictions in this context, it doesn't strictly denote "random" predictions. Contrary to this, the d-index emerges as a more potent tool, aptly delineating the subtle dynamics of imbalanced learning behaviors.

Similarly, the right subfigure in the second row of Fig. 9 illustrates the d-indices of the training and testing data concerning the credit risk dataset. After 100 epochs of learning, the testing data attain a d-index of 1.6149, which, along with a 93.18 % accuracy, 18.17 % sensitivity, and 98.92 % specificity, points to the occurrence of an AIP. This inference comes from the fact that the d-index value is relatively close to 1.5399, a situation arising due to the misclassification of 18.47 % of the minority samples and 1.08 % of the majority samples during learning. Simultaneously, the d-indices of the training data oscillate between 1.58 and 1.63, an indication that they are bordering the d-index of the imbalanced point. Consequently, as depicted in the middle subfigure of the second row of Fig. 9, the corresponding accuracies for the training and test data hover near the majority ratio, registering at 93.05 %.

4. Discussion

Although we know how to determine which SVM models will be optimal when they achieve identical d-index values, assessing the performance of two or more ML models that share the same d-index levels can pose a challenge, especially in the case of different deep learning models. In such situations, we propose adhering to a general principle of prioritizing model simplicity for the final selection; in essence, models with fewer layers, learning nodes, or parameters should be favored. This approach not only simplifies interpretation but also potentially enhances the model's interpretability. For instance, between a CNN model with 10 layers and 3×10^7 parameters and an LSTM model with 5 layers and 5×10^6 parameters that achieve the same d-index, the latter should be chosen for its reduced complexity [36–37]. However, it is important to note that a high-complexity deep learning model that secures a superior d-index compared to a low-complexity alternative should be recognized as the preferable option. Additionally, when models share identical d-index levels and complexity, reproducibility should be a decisive factor in the final selection.

As an explainable metric for assessing ML models, the d-index can also be used to identify complicated overfittings in a more straightforward and explainable approach by avoiding possible biases from traditional metrics for imbalanced data. This is achieved by comparing the d-index derived from the training data predicting itself with the standard d-index obtained using the testing data. An occurrence of overfitting is suggested if there is a substantial divergence between the training and testing d-index values, or if both approach the d-index of the imbalanced point or AIP, such as $\log_2\left(\frac{3(1+\gamma)}{2}\right)$ in binary classification. Furthermore, we applied the transformer model to the IB_EMODB dataset and obtained a d-index value of 1.2718, which is close to the imbalanced point d-index 1.3518 estimated by using the following equation estimated by the theorem 7 in this study.

$$d = \frac{1}{k}\left[\log_2\left[\prod_{i=1}^{k-1}(2-\gamma_i)\left(\frac{3}{2}\right)\right] + \log_2\left(\frac{3}{2}(1+\gamma)\right)\right]_{|k=4,\gamma,\gamma_1\gamma_2\gamma_3\ (0.4233,0.0767,0.23,0.27)} \tag{30}$$

It suggests that the transformer encounter overfitting for its d-index, reaffirming that the deep learning models would encounter overfitting under the imbalanced dataset. When compared with scores such as MCC = 0, Cohen's Kappa = 0, and weighted F1 = 0.2451, all of which imply at random or unsatisfactory predictions, the proposed d-index stands as a more transparent and precise metric for evaluating the outcomes of multiclass imbalanced learning.

Furthermore, the d-index enhances parameter tuning techniques such as grid search, providing more precise results unaffected by imbalanced learning bias — a notable advantage over accuracy and other traditional metrics due to its comprehensive modeling of machine learning performance.

Limitations. As an interpretable learning assessment measure, the d-index comes with certain limitations. Computing the d-index can be particularly expensive, especially with larger datasets involving a high number of classes (e.g., >10) in the classification. This can significantly increase the time complexity of the process, potentially resulting in a time-consuming computation. Additionally, there is a risk of encountering numerical stability issues while computing the local d-indexes for each class, particularly when some classes have a very small number of instances. To mitigate this, it is advisable to either utilize data augmentation and resampling to balance the least represented classes or reduce the number of classes to avoid such complexities.

Impacts of loss function selection on imbalanced deep learning. Although d-index is a sensitive metric for detecting the imbalanced points or AIPs in imbalanced learning, it remains challenging to predict when they will occur during various learning scenarios. This is because some imbalanced learning scenarios may not generate the imbalanced points or AIPs even for very imbalanced data due to the nature of data, learning models or even relevant parameter settings. For example, the transformer model achieves a perfect performance for the simulated credit risk dataset, which is linearly separable data, under the cross-entropy loss, but the same model encounters the imbalanced point d-index: $\log_2\left(\frac{3(1+\gamma)}{2}\right)_{\gamma=0.9222} = 1.5272$ under the focal loss: $\mathrm{FL}(p_t) = \alpha_t(1-p_t)^{\beta}\log(p_t)$, where p_t is the true class probability produced by the model, α_t is a weighting factor for the class and can be set to 1 for balanced datasets, β, typically set as 2, is a focusing parameter that controls the strength of down-weighting for well-classified examples. It suggests that imbalanced point generation can be affected by various factors even if the dataset itself is linearly separable. On the other hand, it implies that loss function selection can play an essential role for some deep learning models in imbalanced learning [42,43].

It appears that deep learning models such as transformer are more likely to produce imbalanced points or AIPs than general ML models. This is likely due to the complex composite decision functions of deep learning models, most of which utilize layer-by-layer mapping mechanisms [44]. As a result, even a small degree of information imbalance may be amplified in the decision function, resulting in the production of

imbalanced points or AIPs. As such, how to design deep learning architecture and training algorithms to mitigate such imbalanced amplification mechanism could be an interesting direction for enhancing the explainability of deep learning models [45].

Resolving LSPs. Furthermore, detecting and solving learning singularity problems (LSPs) present a significant challenge in ML, as there may be various types of such problems in different AI and data science domains. Identifying meaningful solutions for these problems can lead to breakthroughs not only in ML theory but also in various AI and data science applications. Therefore, there is an urgent need to develop more systematic research frameworks to address this challenge, such as categorizing different sources of learning singularity problems and utilizing novel AI tools such as quantum machine learning to investigate them [46].

5. Conclusion

This study introduces the d-index, a novel metric for interpretable ML assessment that is well-suited for both binary and multiclass classification tasks. The d-index introduces new concepts in ML, such as breakeven, imbalanced point, AIPs, and learning singularity problems, which extend the existing ML theory and applications. Compared to traditional metrics such as MCC, F1-score, CEN, and Cohen's Kappa, the d-index provides a more comprehensive and sensitive assessment of ML performance while being self-interpreted and avoiding possible evaluation biases, especially for imbalanced learning. The d-index overcomes the limitations of traditional metrics in achieving good interpretability and brings more robust, accurate, and efficient model selection, making it a valuable tool for both researchers and practitioners in the field. Furthermore, the d-index can improve parameter tuning efficiency and fairness by avoiding possible biases caused by traditional metrics. Its ability to enhance model performance assessment can significantly improve the quality of ML models and facilitate their practical use in various AI and data science applications, making it a crucial tool in the field.

In contrast to traditional metrics, the d-index excels in assessing various ML behaviors, especially in situations of imbalanced learning where traditional measures such as accuracy and F1 score may be biased or even misleading. Its significant advantage lies in its ability to sensitively detect the imbalanced point or AIPs, elements often overlooked by classic metrics. Consequently, it offers fresh insights and techniques to the expanding field of imbalanced learning, a sector steadily gaining traction in AI and data science. Utilizing the d-index allows for the capture of subtle dynamics of imbalanced learning behaviors by rectifying potential biases derived from conventional training and testing loss curve analyses, particularly when the curves demonstrate similar directional trends across epochs. In technical terms, the proposed index reveals a seldom discussed state of overfitting: a scenario where overfitting occurs despite the training and testing loss curves showing favorable reductions and correlations throughout the epochs, especially in the context of imbalanced learning.

Additionally, the d-index facilitates more rigorous and sensitive identification of other anomalous ML behaviors such as underfitting, offering a tool for interpretable ML performance assessment. Specifically, it has been proven that a d-index within the range of $\left(2\log_2\left(\frac{k+1}{k}\right), 2\right]$, indicates a normal learning status, with k representing the number of classes involved in learning. Notably, a d-index falling below $2\log_2\left(\frac{k+1}{k}\right)$, signals the onset of underfitting — a phenomenon particularly prevalent in data imbalance scenarios which traditional learning metrics fail to detect effectively.

Moreover, the d-index paves a new pathway in ML theory, identifying learning singularity problems (LSPs) and marking out the unlearnable sets of imbalanced learning problems within the existing ML landscape. However, the cardinality of these unlearnable sets and their equivalent learnable problems remain unexplored. On another note, given the close relationships between overfitting and LSPs in both traditional ML and modern deep learning models, finding solutions to LSPs could foster new techniques to address the special type of overfitting associated with LSPs, enhancing both the current and future landscape of machine learning.

Furthermore, we demonstrate how to distinguish ML performance under the same d-index values for Support vector machines (SVMs) and propose a meaningful priori kernel selection that achieves a good d-index and generalization. Interestingly, we also prove that SVMs can lose their learning capability by generating the imbalanced point, even if the data is not inherently imbalanced. Given the key status of SVM in reproducible machine learning and kernel-based learning. These findings provide new insights into the two fields from an explainable ML assessment perspective [47,48]. Besides, we show the importance of loss function selection plays an essential role in imbalanced learning for deep learning models such as transformer.

Impacts of normalization on the d-index. In our study, we observed that while different normalization methods can affect the d-index, they generally maintain the characteristics of imbalanced points or AIPs, especially in deep learning models. We utilized standard scaler normalization for the datasets with predominantly Gaussian distributed heterogeneous variables, such as IB-EMODB, credit risk, and simulated credit datasets. Conversely, the minmax normalization was applied to the ovarian dataset, which contains largely non-Gaussian distributed features, to scale them between 0 and 1.

It is essential to note that using different normalization techniques can yield varying d-indices. To illustrate, employing the MaxAbs scaler – defined by the transformation $x_i^{'} = \frac{x_i}{\max(|x_i^{'}|)}$, where $x_i^{'}$ is the transformed one of feature x_i, and $\max(|x_i^{'}|)$ is the maximum absolute value of x_i – on the credit risk dataset resulted in a d-index of 1.5909. Meanwhile, the standard and minmax scalers gave d-indices of 1.6149 and 1.5825, respectively. More details can be found in the supplemental materials.

The d-index serves as an interpretable ML assessment metric, enhancing transparency and reliability in evaluating ML performance. It unveils previously hidden deep learning subtle dynamics and ML behaviors under different imbalanced learning scenarios, streamlining efficient model selection. This makes it indispensable for explainable AI, especially in imbalanced learning scenarios. Our ongoing and future work aims to design interpretable deep learning models with dynamic and adjustable learning topologies along with novel knowledge extraction methods to address the learning singularity problems (LSPs) [49,50]. Our pursuit promises to enrich the AI and data science landscapes, pushing for more interpretable and efficient ML models.

CRediT authorship contribution statement

Henry Han: Conceptualization, Methodology, Software, Data curation , Formal analysis , Writing – original draft , Writing – review & editing. **Yi Wu:** Software. **Jiacun Wang:** Investigation, Validation. **Ashley Han:** Software.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work is partially supported by NASA Grant 80NSSC22K1015, NSF 2229138, and McCollum endowed chair startup fund. Authors

sincerely thank the valuable suggestions and comments from the editor and four anonymous reviewers.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.neucom.2023.126891.

References

[1] Jeff Dean (2022) A Golden Decade of Deep Learning: Computing Systems & Applications, https://www.amacad.org/publication/golden-decade-deep-learning-computing-systems-applications.

[2] H. Han, X. Liu, The challenges of explainable AI in biomedical data science, BMC Bioinformatics 22 (2022) 443.

[3] Jason Hickey, Using Machine Learning to 'Nowcast' Precipitation in High Resolution, Google AI Blog, January 13, 2020, https://ai.googleblog.com/2020/01/using-machine -learning-to-nowcast.html.

[4] Maithra Raghu and Eric Schmidt, "A Survey of Deep Learning for Scientific Discovery," arXiv (2020).

[5] Han, et al., (2021) Predict High-Frequency Trading Marker via Manifold Learning, Knowledge-Based System 213 (2021), 106662.

[6] Antonio Briola, Jeremy Turiel, Riccardo Marcaccioli, Tomaso Aste, Deep Reinforcement Learning for Active High Frequency Trading, arXiv:2101.07107 [cs.LG].

[7] Han, H (2021) Hierarchical Learning for Option Implied Volatility Pricing, *Proceedings of the 54th Hawaii International Conference on System Sciences*, 1573-1582.

[8] B.K. Lee, E.J. Mayhew, B. Sanchez-Lengeling, J.N. Wei, W.W. Qian, K.A. Little, M. Andres, B.B. Nguyen, T. Moloy, J. Yasonik, J.K. Parker, R.C. Gerkin, J. D. Mainland, A.B. Wiltschko, A principal odor map unifies diverse tasks in olfactory perception, Science 381 (6661) (2023 Sep) 999–1006, https://doi.org/10.1126/science.ade4401.

[9] H. Zhang, H. Huang, H. Han, A Novel Heterogeneous Parallel Convolution Bi-LSTM for Speech Emotion Recognition, Applied Sciences. 11 (21) (2021 October) 9897.

[10] Han, et al., Enhance Explainability of Manifold Learning, Neurocomputing 500 (877–895) (2022) 2022.

[11] Burkart, N, Huber, M (2020) A Survey on the Explainability of Supervised Machine Learning, arXiv:2011.07876 [cs.LG].

[12] Y. Chen, R. Calabrese, B. Martin-Barragan, Interpretable machine learning for imbalanced credit scoring datasets, European Journal of Operational Research 312 (1) (2024) 357–372.

[13] Chicco, D., Jurman, G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, *BMC Genomics,* 2022 **21,** 6 (2020) https://doi.org/10.1186/s12864-019-6413-7.

[14] Tharwat, A: Classification assessment methods, Applied Computing and Informatics ISSN: 2634-1964.

[15] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information Processing and Management (2009).

[16] Hand and Christen, A note on using the F-measure for evaluating record linkage algorithms, Statistics and Computing 28 (3) (2018) 539–547.

[17] D. Powers, Evaluation: From Precision, Recall and F-Score to ROC, Informedness, Markedness & Correlation, Journal of Machine Learning Technologies (2011).

[18] J. Opitz; S. Burst (2019). "Macro F1 and Macro F1". arXiv:1911.03347.

[19] Yang, et al., A Case Study of Multi-class Classification with Diversified Precision Recall Requirements for Query Disambiguation, SIGIR 2020 (2020) 1633–1636.

[20] Grandini et al (2020) Metrics for Multi-Class Classification: an Overview arXiv: 2008.05756.

[21] Jurman et al (2012) A Comparison of MCC and CEN Error Measures in Multi-Class Prediction PLOS ONE.

[22] Ballabio et al (2017) Multivariate comparison of classification performance measures 175:15 March *Chemometrics and Intelligent Laboratory Systems* 2018, 33-44.

[23] Boughorbel S, Jarray F, El-Anbari M (2017) Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PLoS ONE 12(6): e0177678 pone.0177678.

[24] H. Han, K. Men, How does normalization impact RNA-seq disease diagnosis? Journal of Biomedical Informatics 85 (2018) 80–92.

[25] Wang, et al., (2016) Improving classification of mature microRNA by solving class imbalance problem, Science Reports 16 (6) (2016 May) 25941.

[26] Lin, Q, Chen, J: Class-imbalanced classifiers for high-dimensional data, *Brief Bioinform* Jan;14(1):13-26.doi: 10.1093/bib/bbs006.

[27] H. Han, X. Jiang, Overcome Support Vector Machine Diagnosis Overfitting, Cancer Informatics 13 (1) (2014) 145–158.

[28] B. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochimica Et Biophysica Acta. 405(2):442±451 (1975).

[29] L.V. Der Maaten, G.E. Hinton, Visualizing High-Dimensional Data Using t-SNE, Journal of Machine Learning Research (2008) 2579–2605.

[30] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, Cambridge, 2011.

[31] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[32] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, International Journal of Computer Vision (2007).

[33] M. Abdel-Mottaleb, W. Alhalabi, Discriminant Correlation Analysis: Real-Time Feature Level Fusion for Multimodal Biometric Recognition, IEEE Transactions on Information Forensics and Security. 11 (9) (2016) 1984–1996.

[34] P. Geurts, D. Ernst, L. Wehenkel, Extremely Randomized Trees. Machine Learning 63 (1) (2006) 3–42.

[35] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 28.

[36] R. Merton, On the Pricing of Corporate Debt: The Risk Structure of Interest Rates, Journal of Finance. 29 (2) (1974) 449–470.

[37] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural computation.*1997 Nov 15;9(8):1735-80.

[38] Zhang, et al., Physics-Informed Deep Learning for Musculoskeletal Modeling: Predicting Muscle Forces and Joint Kinematics From Surface EMG, IEEE Transactions on Neural Systems and Rehabilitation Engineering 31 (2023) 484–493, https://doi.org/10.1109/TNSRE.2022.3226860.

[39] Zhang et al., "Boosting Personalized Musculoskeletal Modeling With Physics-Informed Knowledge Transfer," in IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-11, 2023, Art no. 2500811, doi: 10.1109/TIM.2022.3227604.

[40] Nur Ezlin Zamri, Siti Aishah Azhar, Mohd. Asyraf Mansor, Alyaa Alway, Mohd Shareduwan Mohd Kasihmuddin, Weighted Random k Satisfiability for k=1,2 (r2SAT) in Discrete Hopfield Neural Network, *Applied Soft Computing* 126,2022, 109312.

[41] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems* (pp. 5998-6008).

[42] G. Lemaitre, F. Nogueira, C. Aridas, Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning, Journal of Machine Learning Research 18 (2017) 1–5.

[43] F. Han, S. Zhu, Q. Ling, et al., Gene-CWGAN: a data enhancement method for gene expression profile based on improved CWGAN-GP, Neural Computing and Applications 34 (2022) 16325–16339, https://doi.org/10.1007/s00521-022-07417-9.

[44] V. Sampath, I. Maurtua, J.J. Aguilar Martín, A. Gutierrez, A survey on generative adversarial networks for imbalance problems in computer vision tasks, Journal of Big Data 8 (1) (2021) 1–59.

[45] Li *et al.* (2021) Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond, arXiv:2103.10689.

[46] H.Y. Huang, M. Broughton, M. Mohseni, et al., Power of data in quantum machine learning, Nature Communications 12 (2021) 2631.

[47] Han, et al., Forecasting Stock Excess Returns with SEC 8-K Filings, Communications in Computer and Information Science 1725 (2022) 3–18.

[48] Luo *et al.* Parameterized explainer for graph neural network. arXiv preprint arXiv: 2011.04573, 2020.

[49] Chen *at al* .(2023) Learning A Sparse Transformer Network for Effective Image Deraining, CVPR.

[50] K.R. Moon, D. van Dijk, Z. Wang, et al., Visualizing structure and transitions in high-dimensional biological data, Nature Biotechnology 37 (2019) 1482–1492, https://doi.org/10.1038/s41587-019-0336-3.

Dr. Henry Han is McCollum Family Chair in Data Sciences and Professor of Computer Science at Baylor University. He earned his PhD in Applied Mathematics and Computational Science from the University of Iowa. He has published more than 100 papers in leading journals and conferences in data science/AI, Fintech, machine learning, and informatics fields. His research is supported by NSF, NASA, and NIH.

Mr. Yi Wu is currently pursuing his PhD at the Music and Audio Research Laboratory, New York University. He holds a BS in Computer Science from Fordham University and a BE in Electrical Engineering from Columbia University, both awarded in 2016. Additionally, in 2019, he completed his MS in music technology from the Georgia Institute of Technology. His research is primarily focused on augmented reality complemented by interests in machine learning and AI.

Dr. Jiacun Wang is a Professor in the Computer Science and Software Engineering Department at Monmouth University, West Long Branch, New Jersey. He earned his PhD in computer engineering from Nanjing University of Science and Technology. He has published four books and about 100 papers in software engineering, discrete event systems, formal methods, wireless networking, AI, and real-time distributed systems.

Miss Ashley Han is a high school student at Skyline High School in Ann Arbor, Michigan. Her primary research interests lie at the intersection of AI and programming.