Journal of Big Data

RESEARCH Open Access

Predicting startup success using two bias-free machine learning: resolving data imbalance using generative adversarial networks

Jungryeol Park[1], Saesol Choi[1] and Yituo Feng[2*]

*Correspondence:
Yituo Feng
fengyi47455@naver.com
[1]Technology Strategy Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea
[2]Information Management and Information Systems, Yunnan Minzu University, Kunming, China

Abstract
The success of newly established companies holds significant implications for community development and economic growth. However, startups often grapple with heightened vulnerability to market volatility, which can lead to early-stage failures. This study aims to predict startup success by addressing biases in existing predictive models. Previous research has examined external factors such as market dynamics and internal elements like founder characteristics.While such efforts have contributed to understanding success mechanisms, challenges persist, including predictor and learning data biases. This study proposes a novel approach by constructing independent variables using early-stage information, incorporating founder attributes, and mitigating class imbalance through generative adversarial networks (GAN). Our proposed model aims to enhance investment decision-making efficiency and effectiveness, offering a valuable decision support system for various venture capital funds.

Keywords Predicting startup success, Two bias-free machine learning, Imbalanced data, Generative adversarial networks, Crunchbase

Introduction

Newly established companies spread innovative products and services to the market. Their survival and prosperity have the potential to play an important role in community development and the economic growth of the country. However, startups tend to be relatively more vulnerable to uncertain situations such as market volatility than large companies [1, 2], which can cause failures in the early stages of the life cycle [3], and are facing low survival and growth problems [4, 5]. This is due to the inherent lack of resources, capabilities, and experience of newly established companies, and is known as the Liability of Newness [6, 7]. Venture capital firms investing in these startups expect long-term growth and investment returns, but about 80% of companies fail or perform poorly [8, 9]. Therefore, identifying which companies will survive and grow in the market can be seen as one of the important factors that venture capitalists should consider in deciding on investment [10, 11]. From this point of view, a model that predicts startup success can support venture capital and general investors' decision-making and improve investment performance [10].

In recognition of these problems, studies have been made to identify the factors of startup survival and success. In particular, there is a tendency to pay attention to external environments such as the market to which the company belongs and internal factors such as founder and organizational characteristics. For example, a study [12] found that it is important to match technology and experience between entrepreneurs and members in the success of startups. Another study analyzed the effect of founder's personal information on the success of SMEs [13]. In terms of business administration, the influence of management tools and theories on the success of startups was also investigated [14], and the correlation between founders' personal goals and company growth was also studied [15]. Despite these efforts, many studies have not yet sufficiently identified the mechanism for the success of startups, suggesting that it is difficult to identify it through a single factor [3, 16, 17]. In more recent studies, studies have been conducted to explore various factors to be used in the prediction of surviving companies through data from Crunchbase, one of the largest platforms for providing startup information [18]. Liang & Yuan [19] discovered that they could predict future connected companies by drawing the network of currently connected companies and performing link prediction methods based on this. Dellermann [20] introduced a model utilizing machine learning algorithms incorporating concrete data like team size and entrepreneurial background, alongside group-derived judgments. This approach combines insights from experts and non-experts, who draw upon their market knowledge and instincts to forecast startup success. The outcomes are amalgamated for the final classification outcome [20]. Żbikowski & Antosiuk [21] outlined a blueprint for constructing a predictive model for identifying prosperous startups through machine learning techniques.

These studies have contributed a lot to understanding the success mechanism of startups, but most studies have limitations in that they have not solved the following two bias problems. The first is the bias of predictors. Previous studies have included information known after the company was established. For example, information such as funding events, collaboration with other companies, or various support from VCs may be examples. This can be attributed to the fact that crunchbase provides a function to update data. Several studies, including [22–25], use crunchbase data to analyze funding events and venture capital backing. However, some of these studies suffer from the look-ahead bias, as they use data that is only available after a company has achieved success or failure. This bias can affect the validity of the final model and make it useless in real-world scenarios. Some of the works that exhibit this bias include Yuxian & Yuan [18], Krishna et al. [24], Dellermann [20], Bento [22], and Xiang et al. [23]. The potential harm caused by data leakage between different time periods is also present in Sharchilev et al. [26]. Żbikowski & Antosiuk [21] conducted a study to reduce the bias of predictors by constructing independent variables using only information that would have been known in the early stages of the company's operation. However, there is a limitation in these studies that they did not sufficiently reflect important characteristics such as the founder's major and work experience. The founder's major and work experience in related fields have a positive effect on the survival and growth of the company [5, 26, 27]. Therefore, reflecting the characteristics of these founders in the predictive model can be a way to improve the performance of the predictive model and improve explanatory power.

The second is the bias of learning data. Most previous studies mention that the class of dependent variables is unbalanced in the data to be learned in the predictive model [21, 22, 25, 28–30]. For example, when there are 10 companies, it consists of two successful companies and eight failing companies. This suggests that, as we know well, the success of startups is not easy. However, if such unbalanced data is used as learning data, the model is likely to be biased as it learns mainly data belonging to a number of classes [31]. As a result, the accuracy may increase, but the reproduction rate may be very low [32, 33]. Generally, the focus in classification tasks centers on the minority category, as noted by Leevy et al. [34]. Nevertheless, instances of class imbalance present challenges, hampering attainment of desirable classification results. In certain scenarios, this misclassification of minority class samples [35] could lead to adverse consequences. Take the example of a rare medical condition; an erroneous classification as "normal" might deprive a patient of essential treatment opportunities. Similarly, financial fraud detection could suffer from inaccuracies if unauthorized credit card usage goes undetected [36]. Given the paramount importance of correctly classifying minority instances, addressing data imbalance becomes imperative. Therefore, many studies developing predictive models emphasize the need to solve and learn the model if the class of dependent variables to be learned in the model is unbalanced [32, 33, 37]. However, as far as we know, it is not easy to find studies that have solved this problem in studies that have attempted to predict the success of startups using crunchbase data.

This study proposes the following two methods to overcome the limitations of these previous studies. First, as proposed by Żbikowski & Antosiuk [21], independent variables are constructed using only information that would have been known at the beginning of the company's operation. Variables such as the area where the company is located, city, holding technology, and founder's education level are used as variables for measuring and predicting corporate performance [38–43]. In addition, in this study, related variables were added based on previous studies that claimed that the founder's major and work experience were important variables in predicting startup success. It proposes a method to reduce the bias of predictors by developing a predictive model based on these variables. Second, in this work, the bias of learning data is reduced by solving the class imbalance problem of dependent variables through data oversampling using generative adversarial networks (GAN). Generative adversarial neural networks are the underlying techniques of generative AI and show high performance in the field of synthetic data generation [44–46]. Therefore, in this study, the research field was expanded by using GAN for modeling success prediction of startups. As a result, the predictive model that eliminates these two biases is a challenge worth taking to increase the efficiency and effectiveness of decision making at the investment stage. Our model can be applied directly as a decision support system for different types of venture capital funds. Through this study, we would like to find answers to the following two research questions.

Research question 1 What is the accuracy of a startup's success prediction if a predictive model is built with only known information in the early stages of the company's operation?

Research question 2 Does the performance of the predictive model improve when the class of dependent variables in the learning data solves an unbalanced problem?

Predicting the success of startups using machine learning

The use of machine learning techniques has been prevalent in predicting business success for a long time. Machine learning involves the utilization of algorithms to enable computers to discover new rules and patterns or predict new data outcomes through data learning [47]. Two primary approaches to machine learning include supervised and unsupervised learning. Supervised learning involves the labeling of data to predict future outcomes, with classification and regression tasks used to categorize data based on result characteristics. Unsupervised learning is the process of discovering hidden structures and patterns in data without labels, with clustering and principal components analysis approaches utilized. This study focused on the use of classification techniques in supervised learning to classify data according to specific criteria for predicting discrete outcomes related to the digital divide. A model was developed through the utilization of various data dimensions to distinguish data and predict discrete outcomes for new data [48].

Machine learning offers the advantage of utilizing both categorical and numerical predictors to create models by assessing linear and nonlinear relationships and the importance of each predictor. Unlike traditional statistical methods such as regression analysis, which struggle to maintain basic assumptions about independent variables, machine learning-based prediction models assume that the dependent and independent variables are associated [49]. Moreover, the roles played by dependent variables in predicting independent variables are analyzed, ensuring that predictive power is maintained even in the presence of multicollinearity. Consequently, machine learning can perform analysis even with many variables, using classification algorithms like logistic regression (LR), support vector machine (SVM), and extreme gradient boosting (XGB) [50].

Studies have been conducted to predict the success of startups using these machine learning techniques. Dellermann [20] proposed a success prediction frame for early startups through machine learning. Sharchilev et al. [26] analyzed the impact on success by focusing on factors related to investors and founders. Ross et al. [51] proposed Capital VX, a startup success prediction model, using machine learning and identified influencing factors. More recently, research has been conducted to use crunchbase's data, which provides business information of startups and venture capitalists, for prediction. Xiang et al. [25] performed topic modeling and mergers and acquisitions prediction using techcrunch and crunchbase data. Liang & Yuan [19] established a social network using crunchbase and predicted investor financing behavior through it. Deias & Magrini [30] analyzed the impact of stock financing dynamics on venture success through crunchbase data and logistic regression analysis. Pan et al. [52] proposed a method to predict the success of startups using crunchbase data and logistic regression, k-nearest neighbors, and random forest algorithms. Forecasting business success is an essential but challenging undertaking that has significant implications for both public and private stakeholders, including those who shape economics, make investment and funding decisions, and establish companies. As a company matures and undergoes tests of its product-market fit, as well as the selection processes of angel investors and VC funds, the task of predicting success becomes more manageable. Therefore, most previous studies recognize the class imbalance state of the dependent variable, [21, 22, 25, 29, 30], but are limited in that they fail to suggest a solution to this problem.

Imbalance data problem

The intricacy of imbalanced data classification surpasses that of balanced data classification, as highlighted in prior studies [53, 54]. Addressing imbalanced classification necessitates a distinct focus on enhancing the performance of minority classes, beyond overall performance. Achieving higher recognition rates for minority samples while maintaining overall recognition poses a formidable challenge. In this investigation, we denote the class with abundant samples as the majority, the class with fewer as the minority, and the ratio between them as the imbalance rate (It is expressed as the number of majority class samples and the number of minority samples, respectively expressed in formula (1)) [55].

$$IR = \frac{Number\ of\ majority\ class\ samples}{Number\ of\ minority\ class\ samples} \tag{1}$$

The classification problem in machine learning or deep learning is to find an appropriate classification boundary for each class from the training data set, and to predict the class of new data with this model. If the characteristics of each class are clear and the number of data is equal, an ideal classifier will be formed, but if the data is concentrated in one class, a classification boundary line concentrated in multiple classes will be formed. Namely, when constructing a predictive model through machine learning, an imbalanced class distribution in the training data can lead to bias toward the larger class [31], adversely affecting model performance [32, 33]. This phenomenon, known as class imbalance, demands careful consideration in prediction research, particularly within finance, healthcare, and manufacturing [56]. For instance, in marketing, when devising a churn prediction model, an imbalance in training data comprising predominantly non-churning customers can lead the model to struggle in correctly classifying new customers [57].

While classification tasks generally emphasize minority classes [34], class imbalance introduces challenges in obtaining satisfactory results, potentially causing harm by misclassifying minority instances [35, 58]. This could result in missed medical treatment opportunities for patients with rare illnesses or overlooking fraudulent activities, such as unauthorized credit card usage [36]. The accurate classification of minority classes becomes crucial, prompting the need to address data imbalance. There are largely two methods to overcome the imbalance data problem a data-level methods and an algorithm-level methods.

Data-level methods are categorized into undersampling reducing majority class data to match the minority class and oversampling generating data to match the majority class [59], depicted in Fig. 1 below.

Undersampling offers efficiency and cost savings by reducing data collection, yet it sacrifices valuable information [60]. Conversely, oversampling, although extending model training time due to data generation, mitigates information loss. To address imbalanced data, numerous researchers have delved into this topic, suggesting pertinent algorithms. One of the representative techniques of oversampling is SMOTE (Synthetic Minority Over-Sampling Technique). SMOTE is a technique for generating data through bootstrapping and the k-nearest neighbor methods. For specific data belonging to a minority

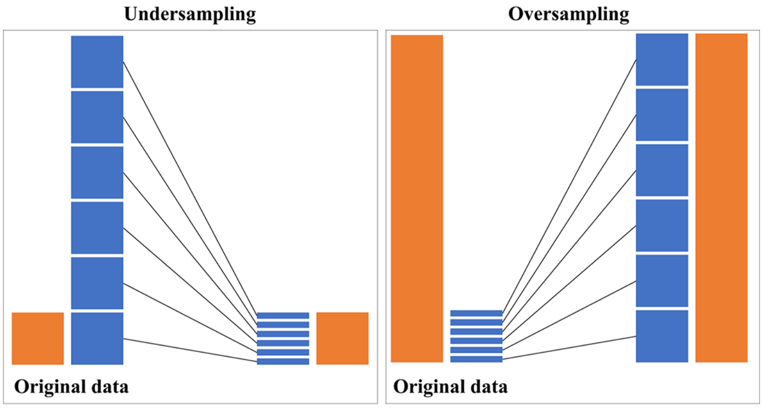

Fig. 1 Data sampling process

class, k-nearest neighbors of the same minority class are found, and new data are generated between them by creating a linear connection structure with the neighbors [61]. However, when SMOTE generates data belonging to a minority class, it does not consider the location of data belonging to the adjacent majority class. Therefore, the problem of overlapping positions occurs [62]. In addition, since data is generated by relying only on the relationship between a few classes, overfitting may occur and prediction performance may deteriorate [63].

Standard machine learning classifiers often address the imbalance problem by improving or creating new algorithms [2, 7]. One of the most prevalent methods at the algorithm level is cost-sensitive learning, which adjusts the weight of misclassified minority class samples by incorporating misclassification costs into the classification decision. For instance, Chung et al. [25] incorporated cost information into the loss function of convolutional neural networks (CNN), which helped mitigate the imbalance problem in CNN classification. Zhang [26] proposed two cost-sensitive algorithms for KNN classifiers, namely Direct-CS-KNN and Distance-CS-KNN, both of which aim to minimize misclassification costs. However, the need for a cost matrix provided by domain experts in cost-sensitive algorithms restricts their practicality. Moreover, the specificity of cost-sensitive algorithms to particular domains reduces their general applicability across different fields.

Apart from sampling strategies, ensemble methods based on bagging and boosting have also been widely applied to the class imbalance problem. The fundamental concept of classifier ensemble learning involves creating multiple classifiers from the original dataset and then combining their predictions to classify new samples. The primary reason for merging classifiers in redundant ensembles is to enhance their generalization capabilities: each individual classifier is likely to make errors due to being trained on a limited dataset, but the errors made by different classifiers are not necessarily identical. Related research includes a comprehensive study by Seiffert et al. [64], which compared sampling methods with boosting to enhance the performance of decision tree models for identifying defective software modules. Chawla et al. [61] proposed a novel approach called SMOTEBoost, which combines the SMOTE algorithm with the boosting procedure to learn from imbalanced datasets. However, in each iteration of the original bagging and boosting methods, the class imbalance problem may still persist because the sampled subset in a given iteration has a similar class distribution to the original dataset.

As such, there are several methodologies for resolving the data imbalance, but they still have limitations. Therefore, it is necessary to generate new data that does not overlap with the existing data while considering the overall distribution of the data when generating data was derived [65]. To overcome these issues, we adopted the generative adversarial network (GAN) concept. GAN can solve overfitting and data superposition problems because it learns the actual data distribution of minor class and then generates similar data [66]. Through this, GAN can overcome the limitations of existing oversampling techniques. Recently, GAN have found broad utility in image, voice, and text domains [67, 68]. Capitalizing on their adept data fitting, GAN have been extensively explored for data augmentation and enhancement [69].

Generative adversarial networks (GAN)

Generative adversarial networks (GAN) functions as an unsupervised neural network where a generator produces data and a discriminator assesses it against real data, engaging in a competitive learning process for optimization [66]. This mutual competition enhances the quality of generated data, aligning it closely with actual data quality. Illustrated in Fig. 2, the generator employs random noise (Z) to create synthetic data, which is then evaluated by the discriminator against real data. The outcome yields a Loss value, utilized to guide the generator in mimicking authentic data, thereby advancing the learning process. This iteration continues until the discriminator struggles to differentiate between real and synthetic data, signifying optimal generator performance [66]. The architectural depiction of the GAN framework is presented in Fig. 2 below.

The generator aims to maximize the probability $D(G(z))$ that the discriminator distinguishes fake data from real data by 1 by generating fake data similar to the real data, and the discriminator aims to maximize the probability $D(x))$ that the generated fake data is fake. The objective function equation for proceeding with this learning is as follows.

$$\underset{G}{min}\underset{D}{max}V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2}$$

$\underset{G}{min}\underset{D}{max}V(D,G)$ represents the minimax optimization process between the generator $(G)$and the discriminator (D). The discriminator (D) tries to maximize this value to correctly distinguish real data from generated data, while the generator (G)tries to minimize this value to generate data that can fool the discriminator. $\left(E_{x\sim p_{dt}(x)}[logD(x)]\right)$ is the expected value of the log probability assigned by the discriminator $(D)$to real data samples. Specifically, it captures the expectation over the real data (x) sampled from the true data distribution $(p_{data}(x))$. The discriminator (D) aims to maximize this term to accurately classify real data as authentic. Formally, this term reinforces the discriminator ability to recognize genuine data points. $\left(E_{z\sim p_z(z)}[log(1-D(G(z)))]\right)$ is the expected

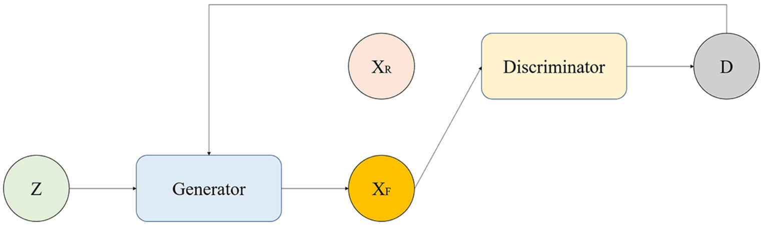

Fig. 2 Generative adversarial networks

value of the log probability assigned by the discriminator (D) to fake data generated by the generator (G). Here, the expectation is taken over the latent variable (z)sampled from a prior distribution $(p_z(z))$. The generator (G) aims to minimize this term, striving to produce fake data $(G(z))$ that the discriminator (D) will misclassify as real. This term penalizes the discriminator for correctly identifying generated data as fake, thus driving the generator to produce more realistic data samples. This adversarial training process encapsulated by the minimax objective encourages the generator to create increasingly realistic data while simultaneously pushing the discriminator to become more adept at distinguishing real from fake data. Under the framework of GAN, the generator and discriminator are trained iteratively. The generator improves its ability to produce realistic data, while the discriminator enhances its capability to identify genuine data, thereby refining the overall performance of the model. With the advent of GAN, studies using them have been conducted in various fields. Due to the concept of learning existing data to generate new, similar but non-overlapping data, it has been mainly used in the field of dealing with unstructured data such as images. Recently, as it has begun to be applied to databaseized structured data, studies applying generative adversarial networks in the oversampling process to solve the problem of class imbalance data have been attempted [44–46]. Therefore, in this study, the research field related to synthetic data generation and class imbalance solution was expanded by using GAN for success prediction modeling of startups.

Materials & methods

Research process

Our research process is largely divided into (1) data preprocessing, (2) data analysis, and (3) results stages. A schematic diagram of this is shown in Fig. 3 below, and the contents of each process are described below.

Data source

This study collected bulk data containing information from global startups through the API of crunchbase (as of March 2023). The data sets used in the analysis were generated using information from the organization, degree, people, job and funding tables. The rest

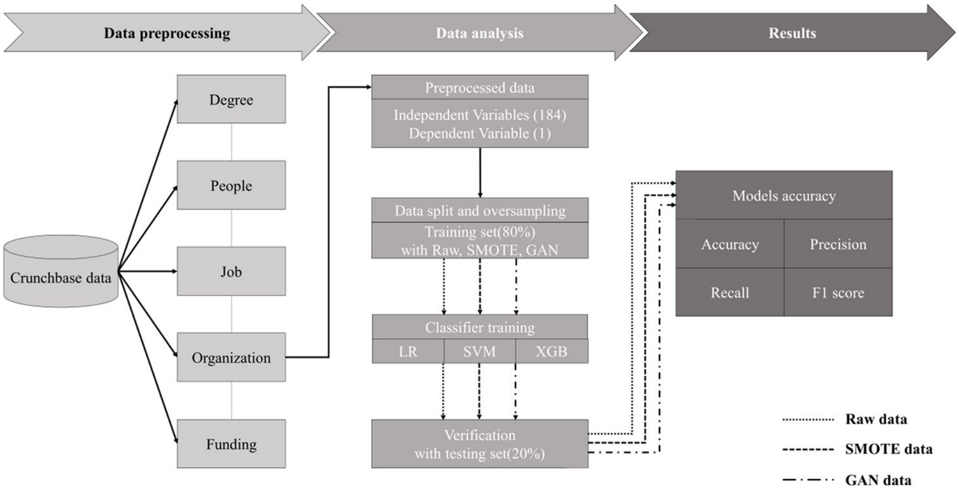

Fig. 3 Research process

of the table was not used because it had text data or contained information known after the company was established. Through this, we tried to reduce the bias of predictors and solve our first research question. We extracted only the categories_list, the experience of attracting funding more than once, and the rows in which the company rule is set as a company. The dataset used in the final analysis consists of 190,773 rows and 185 columns. Table 1 below is a list of independent variables used in the study.

After reducing the number of unique values in the dataset, the resulting set of attributes should be encoded before using it as a dataset in model training. In this study, category and category group columns performed 'one-hot encoding'. Category columns have many unique values, but most rarely appear. We used only columns whose number of instances for category values corresponds to or greater than the upper quartile to avoid a rapid increase in the number of dimensions. It consists of a total of 124 columns. In addition, a total of 47 columns were used for analysis in the category list group. Other variables used 'ordinal encoding'. On the other hand, if the company's operating period is used as it is, there is a possibility that the model will be biased. This is because it is recognized that the longer the company's operating period, the higher the likelihood of success in the market [70]. To offer more comprehensive data on a company's originator, the year of the company's founding has been substituted with the duration between the founder's graduation and the establishment of the firm. This fresh aspect provides better insights into the founder's professional background while creating the company. Similarly, instead of using specific matriculation and graduation dates of the founder, the number of years they spent studying in the university has been utilized. The missing values were filled with median values for each column.

We constructed predictors inspired by previous studies [21], but unlike previous studies, there is a difference in that the founder's major and previous work experience were added as predictors. The founder's major has a positive impact on attracting investment [71], and past work experiences have been reported to increase the efficiency of strategy and decision-making for the operation of environmentally sensitive startups [72, 73]. It is hoped that the addition of these predictors will increase the explanatory power of the model for predicting startup success and further contribute to the theoretical development of related research fields.

Table 1 Independent variables

Variables	Description	Type	Characteristics
Category_list	List of organizations' subcategories	Nominal	Company
Category_groups_list	List of organizations' categories	Nominal	
Region_org_size	Rank of region in number of startups	Categorical	
City_org_size	Rank of city in number of startups	Categorical	
Operation_year	Number of years between founder's graduation and company's foundation	Interval	
Gender	Founder's gender	Nominal	Founder
Major	Founder's major	Categorical	
Has_multiple_degrees	Founder has more than one degree	Boolean	
Studying_year	Number of years between founder's matriculation and graduation	Interval	
Work_experience	Founder's work experience	string	

Dependent variable

In our study, securing series B were selected as a classification criterion for success. If a startup receives a series B, it means that it has gone through the investment fund selection process from venture capital twice, which is proposed to be a strong indicator of a company's success [21, 74]. Therefore, through the organization and funding tables, we created a dependent variable by labeling companies currently operating and attracting Series B or higher as successful companies 1 and other companies as 0. The funding table was used to create dependent variables, and other information was not included in the data set. 13.7% (25,944 companies) correspond to 1 (success), and 86.3% (164,829 companies) correspond to 0 (failure). This is a highly disproportionate form of about 1/9, which meets the conditions for elucidating our second research question. The ratio of the dependent variable is shown in Fig. 4.

Classification algorithms

This research employed logistic regression (LR), support vector machine (SVM), and extreme gradient boosting (XGB) to build predictive models. LR and SVM are well-established methods utilized in prior crunchbase studies [22, 24, 25]. XGB, which combines decision trees and boosting techniques, has gained recent traction due to its performance in Kaggle competitions [75]. Default parameter values were applied to prevent bias towards specific datasets. A brief overview of each classification algorithm is provided below.

LR serves as an analytical tool to establish causal relationships between independent and dependent variables. It addresses categorical dependent variables, categorizing them into dichotomies for two categories or polynomials for more than three. This technique proves valuable for diverse classification tasks. The formula is outlined as follows.

$$ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \tag{3}$$

$ln\left(\frac{p}{1-p}\right)$ represents the logit transformation, which is the natural logarithm of the odds of the probability p of an event occurring to the probability $1-p$ of the event

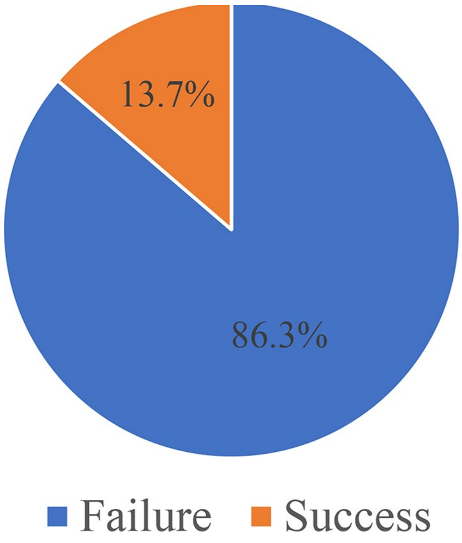

Fig. 4 Dependent variable rate

not occurring. This transformation allows the dependent variable p to be modeled as a linear combination of the independent variables $X_1, X_2, \ldots, X_p$. $ln\left(\frac{p}{1-p}\right)$ is the logit transformation, which is the natural log of the odds. It expresses the dependent variable p as a linear function of the independent variables. β_0 is the intercept. It represents the log odds of the dependent variable when all independent variables X_i are zero.$\beta_1 X_1, \beta_2 X_2, \ldots, \beta_p X_p$ are the coefficients and the corresponding independent variables. Each coefficient β_i indicates the impact of the respective independent variable X_i on the log odds of the dependent variable. The value of β_i shows the change in the log odds for a one-unit change in X_i.

The SVM employs optimal boundaries in three-dimensional space for data separation [76]. The boundary, termed a hyperplane, classifies new data based on a given dataset's category. The goal of SVM is to maximize the margin between the two classes. To maximize this margin, the following optimization problems must be solved. SVM is valuable for pattern recognition and classification tasks. The formula is outlined below.

$$f(x) = w^T x + b \tag{4}$$

The function $f(x)$ represents a linear combination of the input vector x. This linear combination includes the dot product of the weight vector w and the input vector x plus an additional bias term b. b is a scalar known as the bias term. It allows the model to make predictions even when all input features are zero. This linear function can be used in various machine learning models, such as linear regression, logistic regression, and support vector machine (SVM). In the context of linear regression, this equation models the dependent variable y as a linear function of the independent variables in x. In logistic regression, this linear combination $w^T x + b$ is passed through a sigmoid function to predict probabilities.

In SVM, the goal is to find the optimal hyperplane that separates different classes in the feature space. The equation defines the hyperplane, where w determines the orientation and b adjusts the position. In summary, $f(x) = w^T x + b$ is a foundational equation in machine learning, representing the linear relationship between input features and the output prediction.

XGB, an enhanced version of the boosting approach with decision trees, incorporates an internal mechanism to counter overfitting and applies internal cross-validation to each trial [77]. Renowned for its exceptional classification performance, XGB is a favored choice in competitions like kaggle. Notably, its greatest asset lies in its practical utility. XGB permits the extraction of vital indices signifying relatively influential variables among numerous independent factors, enabling an assessment of their relative predictive strength. Hence, XGB was selected for this investigation. The formula is outlined below.

$$L^{(t)} = \sum\nolimits_{i=1}^{n} l\left(y_i, \widehat{y_1^{(t-1)}} + f_t(x_i)\right) + \Omega\ (f_t) \tag{5}$$

$L^{(t)}$ denotes the loss function at iteration t. $\sum\nolimits_{i=1}^{n}$indicates the summation over all data points in the dataset. y_i represents the actual target value for the i-th data point. $\widehat{logy_1^{(t-1)}}$ is the logarithm of the predicted value for the i-th data point from the previous iteration $(t-1)$. $f_t(x_i)$ is the model's prediction for the i-th data point at iteration t. $\Omega\ (f_t)$ is a regularization term that penalizes the complexity of the model at iteration

t . In summary, this objective function combines the prediction error and a regularization term to ensure that the model not only fits the data well but also remains as simple as possible.

Classification evaluation metrics

To forecast startup success, a supervised learning-based prediction model was trained through machine learning. The subsequent analysis gauged prediction accuracy. The procedure encompassed the subsequent steps. Initially, a turnover intention prediction model was established, incorporating all variables from Table 1 as explanatory factors and startup success as the outcome. Next, the data was divided into 70% training and 30% test sets. Following training on the turnover intention model with the training set, prediction accuracy was assessed using the test set. Lastly, LR, SVM, and XGB were employed to analyze the prediction model. Assessment metrics included accuracy, precision, recall, and f1-score to evaluate prediction performance. The definitions and formulas for each metric are provided below.

Accuracy Accuracy is the most intuitive performance measure and is simply a ratio of correctly predicted observations to the total observations.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{6}$$

Precision Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall Recall is the ratio of correctly predicted positive observations to all observations in the actual "yes" class.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

F1-score The f1-score is the weighted average of precision and recall.

$$F1 - score = 2 \times \frac{Precision \times \ Recall}{Precision + Recall} \tag{9}$$

Data oversampling

In this study, to address the issue of class imbalance, we employed Generative Adversarial Networks (GAN) to perform oversampling exclusively on the training dataset. This approach ensures that the test dataset remains unaltered, thereby maintaining the integrity and fairness of the model evaluation process. Adhering to standard practices, this methodology allows for an unbiased assessment of the model's performance and facilitates robust comparisons with other techniques. This approach is supported by recent studies that highlight the efficacy of using GAN for training data augmentation while keeping the test data intact to ensure valid performance evaluation [78]. The generator layer consists of Z (50) → 64 → 128 → 256 → 184 (number of explanatory variables). The discriminator layer consists of the number of generator outputs (184) → 256 → 128 → 64 → 32 → 16 → 1 (sigmoid). This is shown in Fig. 5 below. We learned the distribution of data corresponding to minority classes in the constructed neural network. The size of the batch size was 50, and the epoch was performed 2,000 times. Since learning 1,000 times, the loss values of the generator and discriminator have not changed significantly near 0.4, so only 2,000 times were performed and the learning was terminated. This is shown in Fig. 6 below. The X axis in Fig. 6 represents the number of times the model

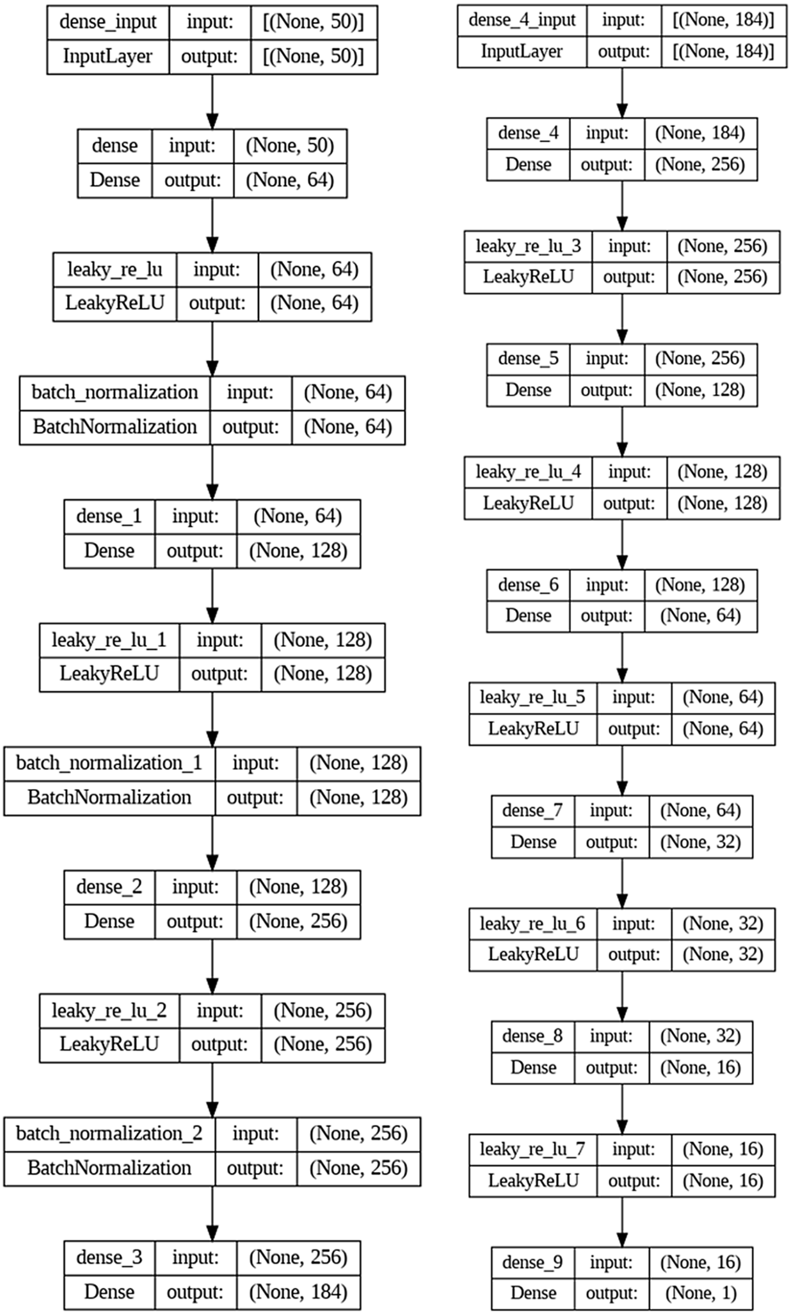
dense_input | input: | [(None, 50)]
InputLayer | output: | [(None, 50)]
dense | input: | (None, 50)
Dense | output: | (None, 64)
leaky_re_lu | input: | (None, 64)
LeakyReLU | output: | (None, 64)
batch_normalization | input: | (None, 64)
BatchNormalization | output: | (None, 64)
dense_1 | input: | (None, 64)
Dense | output: | (None, 128)
leaky_re_lu_1 | input: | (None, 128)
LeakyReLU | output: | (None, 128)
batch_normalization_1 | input: | (None, 128)
BatchNormalization | output: | (None, 128)
dense_2 | input: | (None, 128)
Dense | output: | (None, 256)
leaky_re_lu_2 | input: | (None, 256)
LeakyReLU | output: | (None, 256)
batch_normalization_2 | input: | (None, 256)
BatchNormalization | output: | (None, 256)
dense_3 | input: | (None, 256)
Dense | output: | (None, 184)
dense_4_input | input: | [(None, 184)]
InputLayer | output: | [(None, 184)]
dense_4 | input: | (None, 184)
Dense | output: | (None, 256)
leaky_re_lu_3 | input: | (None, 256)
LeakyReLU | output: | (None, 256)
dense_5 | input: | (None, 256)
Dense | output: | (None, 128)
leaky_re_lu_4 | input: | (None, 128)
LeakyReLU | output: | (None, 128)
dense_6 | input: | (None, 128)
Dense | output: | (None, 64)
leaky_re_lu_5 | input: | (None, 64)
LeakyReLU | output: | (None, 64)
dense_7 | input: | (None, 64)
Dense | output: | (None, 32)
leaky_re_lu_6 | input: | (None, 32)
LeakyReLU | output: | (None, 32)
dense_8 | input: | (None, 32)
Dense | output: | (None, 16)
leaky_re_lu_7 | input: | (None, 16)
LeakyReLU | output: | (None, 16)
dense_9 | input: | (None, 16)
Dense | output: | (None, 1)

Fig. 5 Generator and discriminator layers

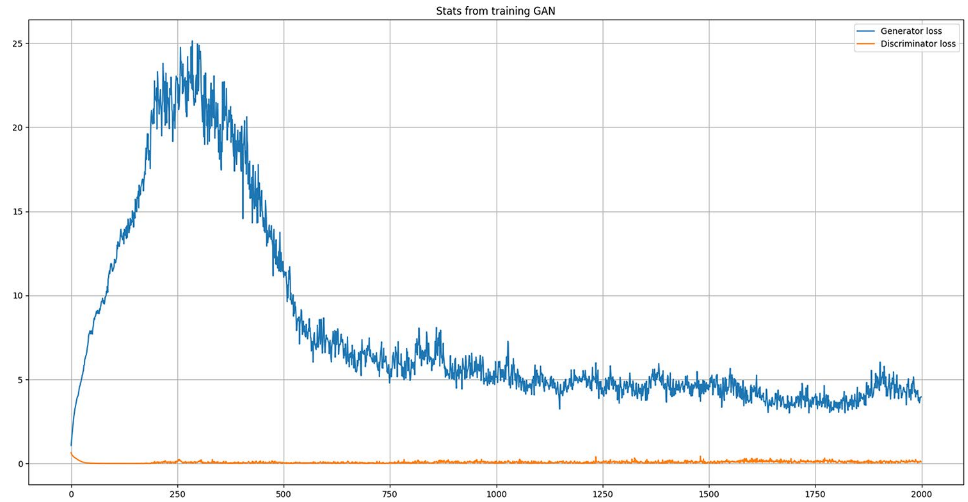

Fig. 6 Training process

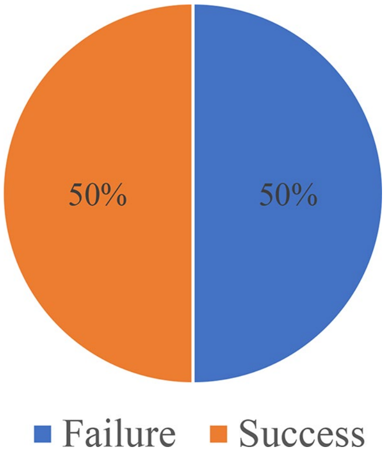

Fig. 7 Percentage of final dependent variable

was trained. In other words, values from 0 to 2000 mean that a total of 2000 epochs have occurred. The Y axis represents loss values. The loss value is an indicator of how well the model is learning. Through this learned generator, we generated insufficient minority class data to match the class ratio of the dependent variable. This is shown in Fig. 7. In this study, to demonstrate the superiority of the proposed methodology, data were oversampled through SMOTE, which has been traditionally used in the oversampling field, and used for analysis of results.

Result

When the class ratio of the imbalanced raw data was approximately 9:1, the results from training the predictive model indicated that among the three algorithms, extreme gradient boosting (XGB) yielded the highest average value across all classification performance evaluations at 81.1%. This was followed by support vector machine (SVM) at 51.8% and logistic regression (LR) at 50% (refer to Table 2). Based on accuracy, which is a simple accuracy, it can be seen as a contrast to all classifiers showing compliance performance of more than 87%. This suggests that when evaluating the performance of

Table 2 Prediction analysis results

Classifier	Accuracy			Precision			Recall			F1 score		
	R*	**S****	**G*****	***R***	**S**	**G**	***R***	**S**	**G**	***R***	**S**	**G**
LR	0.876	0.678	0.904	0.734	0.691	0.944	0.146	0.646	0.859	0.244	0.667	0.900
SVM	0.879	0.788	0.927	0.165	0.738	0.859	0.757	0.819	0.943	0.271	0.777	0.899
XGB	0.941	0.925	0.965	0.846	0.939	0.979	0.695	0.908	0.951	0.763	0.923	0.965

*R: Raw data, **S: SMOTE, ***G: GAN

models that have learned unbalanced data, accurate performance evaluation may be difficult with accuracy alone.

In this context, LR demonstrated high precision but low recall, whereas SVM exhibited the opposite, with low precision and high recall. As briefly discussed earlier, precision represents how many successful companies are included in the predicted results, and recall represents how well successful companies are predicted. In general, precision and recall have an inverse relationship with each other. The conflicting results of LR and SVM can be seen as reflecting this tendency. Therefore, these results suggest that when verifying the model, it is necessary to examine the two values overall through F1-score, etc., rather than evaluating the performance with only one of the values of precision or recall.

Additionally, when examining the harmonic mean of these indicators, known as the F1-score, XGB performed respectably at 76.3%, in contrast to LR and SVM, which were at 24.4% and 27.1% respectively. This suggests that using LR or SVM, our model misclassified approximately 70% of the successful startups as failures. Taking a comprehensive look at these results, it suggests that it may be difficult to derive high performance if class imbalanced data is used as training data when developing a prediction model that discovers startups with high chances of success.

Furthermore, when the class ratio was balanced to 5:5 through generative adversarial networks (GAN) and the balanced data were trained in the predictive model, XGB again demonstrated the highest average value across all classification performance metrics at 96.5%, followed by SVM at 90.7%, and LR at 90.1%. In other words, not only XGB but also other algorithms like LR and SVM were able to increase performance by more than threefold when trained with balanced data as opposed to imbalanced original data. These analytical results suggest that when developing predictive models, resolving the class imbalance in the dependent variable of the training data can be an effective strategy for achieving higher performance enhancement.

On the other hand, in the case of data that solved the imbalance problem through SMOTE, the precision and recall values were improved compared to the raw data in all classifiers. However, SMOTE also showed lower overall predictive performance than data using GAN. These results can be seen to prove that the method proposed in this study can better solve the sample imbalance problem.

Conclusion

In this study, we have proposed methodologies to mitigate two biases that may emerge in predicting startup success via machine learning. The significance of our research is articulated across four main domains.

Firstly, a constant flow of research utilizing machine learning for predicting startup success has emerged over the years [10, 19, 25, 26, 30, 51, 79]. While these studies have highlighted various approaches to complement the limitations of traditional econometric models, most have relied on variables available post-inception, such as networking level, investment records, and activity information. Despite enhancing prediction accuracy, this focus may induce bias towards specific firms and overlook startups with latent potential. To address this, we have proposed the use of early-stage information, an approach that enhances the reproducibility of results in real-world scenarios. Secondly, the founder's major and past work experience have a positive effect on the startup's performance [39–41, 43, 80]. It is known that founders with a high level of education in the field of start-ups are relatively more likely to establish appropriate strategies and make correct decisions according to environmental changes [73]. Therefore, this study attempted to increase the explanatory power of the model by adding the founder's major and past work experience in addition to the eight variables suggested by the previous study.

Thirdly, we introduced an approach to overcome model bias by addressing class imbalance in training data through oversampling with generative adversarial networks (GAN). According to our analysis, models trained on balanced data exhibited significantly enhanced predictive performance. This aligns with previous studies demonstrating improved model performance when class imbalance is resolved [10, 29, 81]. The novelty in our study lies in the application of GAN to produce data representing minority class distribution without duplication, a significant contribution to bias reduction. Fourthly, the methodology proposed in this study can be effective for transfer learning when building a prediction model. For example, it can be used for research to predict which companies will succeed in IPO or which companies will grow into unicorn companies in the future. As a result, this approach can be helpful in the development of a decision-making system to help venture capitalists make investment decisions, and can be useful in the search for successful startups.

Despite these contributions, this study has the following limitations. First, in this study, various text data (e.g., company description) of company information provided by crunchbase could not be utilized. Company descriptions on crunchbase are basically written by the person in charge of the company, so objectivity may be lacking. However, if information hidden in text is explored through various text mining techniques (e.g., topic modeling, sentiment analysis), it can be applied to the development of new predictors. In addition, in this study, various algorithms such as LR, SVM, and XGB were used to build the predictive model, but the hyperparameters were set to default values and analyzed. In the future, if the prediction model is tuned using gridsearch, which finds hyperparameters optimized for the prediction model and training data, higher performance can be derived.

Abbreviations

GAN	Generative Adversarial Networks
LR	Logistic Regression
SVM	Support Vector Machine
XGB	eXtreme Gradient Boosting

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40537-024-00993-8.

Supplementary Material 1

Acknowledgements
Not applicable.

Author contributions

JRP: designed the idea and experiments, and he wrote the manuscript and implemented the code. SSC: provided ideas during the analysis process. YTF: has finalized the structure and content of the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government. [24ZF1130, ICT National Technology Strategy Policy Research].

Data availability
Data is provided within the supplementary information file.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
The publisher has the author's permission to publish this paper.

Competing interests
The authors declare no competing interests.

Author information
JRP[1] is a researcher at the Electronics and Telecommunications Research Institute (ETRI), Technology Strategy Research Division. He received his Ph.D., M.S., and B.S. degrees in Management Information Systems (MIS) from Chungbuk National University. He has published papers related to machine learning and deep learning in several journals, including the Journal of Big Data, Scientific Reports, and PLOS ONE. His main areas of interest are machine learning, deep learning-based business prediction, and technology growth prediction. SSC[2] is a principal researcher in the Technology Strategy Research Division of the Electronics and Telecommunications Research Institute, Republic of Korea. He received his B.S. and M.S. degrees from KAIST. He has published papers in such journals as Online Information Review, Journal of Organizational and End User Computing, International Telecommunications Policy Review, and Korean Journal of Information Technology Applications and Management, and has presented at several IEEE-sponsored international conferences. His research interests include technology management, data driven policy, and IT user behavior. YTF[3*] earned both his Master's and Doctorate degrees in Management Information Systems (MIS) from the Graduate School of Business at Chungbuk National University, South Korea. He currently works at Yunnan Minzu University in Information Management and Information Systems. Dr. Feng has made significant contributions to the academic community with his works published in esteemed SSCI and SCI journals. Additionally, he has served as a reviewer for various SSCI and SCI journals. His research encompasses machine learning, deep learning, data analytics, Structural Equation Modeling, personnel management, business forecasting, and link prediction.

Received: 14 April 2024 / Accepted: 26 August 2024
Published online: 03 September 2024

References
1. Lee S, Geum Y. How to determine a minimum viable product in App-based lean Start-ups: Kano-based Approach. Total Qual Manage Bus Excellence. 2021;32(15–16):1751–67.
2. Miski A. Development of a Mobile application using the lean startup methodology. Int J Sci Eng Res. 2014;5(1):1743–8.
3. Soto-Simeone A, Sirén C, Antretter T. New Venture Survival: a review and extension. Int J Manage Reviews. 2020;22(4):378–407.
4. Robinson K-C. An examination of the influence of industry structure on eight alternative measures of New Venture performance for high potential Independent New ventures. J Bus Ventur. 1999;14(2):165–87.
5. Song M, Podoynitsyna K, Van Der Bij H, Halman J-I-M. Success factors in New ventures: a Meta-analysis. J Prod Innov Manage. 2008;25(1):7–27.
6. Freeman J, Carroll G-R, Hannan M-T. The liability of newness: Age Dependence in Organizational Death Rates. Am Sociol Rev, 1983;692–710.
7. Morse E-A, Fowler S-W, Lawrence T-B. The impact of virtual embeddedness on New Venture Survival: overcoming the liabilities of newness. Entrepreneurship Theory Pract. 2007;31(2):139–59.
8. Picken J-C. From startup to Scalable Enterprise: laying the Foundation. Bus Horiz. 2017;60(5):587–95.
9. Weking J, Böttcher T-P, Hermes S, Hein A. Does Business Model Matter for Startup Success? A Quantitative Analysis, 2019.
10. Khoda M-E, Kamruzzaman J, Gondal I, Imam T, Rahman A. Malware Detection in Edge devices with fuzzy oversampling and dynamic class weighting. Appl Soft Comput. 2021;112:107783.
11. Shepherd D-A, Souitaris V, Gruber M. Creating New ventures: a review and research agenda. J Manag. 2021;47(1):11–42.
12. Stuart R, Abetti P-A. Start-up ventures: towards the prediction of initial success. J Bus Ventur. 1987;2(3):215–30.
13. Vaughan L-Q. The contribution of information to Business Success: a LISREL Model Analysis of Manufacturers in Shanghai. Inf Process Manag. 1999;35(2):193–208.
14. Makridakis S. Factors affecting success in business: management Theories/Tools Versus Predicting Changes. Eur Manag J. 1996;14(1):1–20.
15. Cooper A-C. Challenges in Predicting New Firm performance. J Bus Ventur. 1993;8(3):241–53.
16. Del Sarto N, Cruz Cazares C, Di Minin A. Startup accelerators as an Open Environment: the impact on startups' innovative performance. Technovation. 2022;113:102425.
17. Ugur M, Vivarelli M, Innovation. Firm Survival and Productivity: the state of the art. Econ Innov New Technol. 2021;30(5):433–67.
18. Yuxian E-L, Yuan S-T-D. Investors are social animals. Predicting Investor Behavior using Social Network Features via Supervised Learning Approach; 2013.
19. Liang Y-E, Yuan S-T-D. Predicting Investor Funding Behavior using Crunchbase Social Network features. Internet Res. 2016;26(1):74–100.
20. Dellermann D. Going East: a Framework for Reverse Innovation in SMEs. J Bus Strategy. 2017;38(3):30–9.

21. Żbikowski K, Antosiuk PA, Machine Learning. Bias-Free Approach for Predicting Business Success using Crunchbase Data. Inf Process Manag. 2021;58(4):102555.
22. Bento F-R-S-R. Predicting Start-Up Success with Machine Learning. Universidade Nova de Lisboa; 2018.
23. Huang W-B, Liu J, Bai H, Zhang P. Value Assessment of companies by using an Enterprise Value Assessment System based on their public transfer specification. Inf Process Manag. 2020;57(5):102254.
24. Krishna A, Agrawal A, Choudhary A. Predicting the Outcome of Startups: Less Failure, More Success, In. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016;798–805.
25. Xiang G, Zheng Z, Wen M, Hong J, Rose C, Liu C. A Supervised Approach to Predict Company Acquisition with Factual and Topic Features Using Profiles and News Articles on TechCrunch, In Proceedings of the International AAAI Conference on Web and Social Media, 2012;6(1):607–610.
26. Sharchilev B, Roizner M, Rumyantsev A, Ozornin D, Serdyukov P, De Rijke M. Web-based Startup Success Prediction, In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018;2283–2291.
27. Maxwell A-L, Jeffrey S-A, Lévesque M. Business Angel early stage decision making. J Bus Ventur. 2011;26(2):212–25.
28. Omri A, Frikha M-A, Bouraoui M-A. An empirical investigation of factors affecting Small Business Success. J Manage Dev. 2015;34(9):1073–93.
29. Arroyo J, Corea F, Jimenez-Diaz G, Recio-Garcia J-A. Assessment of Machine Learning performance for decision support in Venture Capital Investments. IEEE Access. 2019;7:124233–43.
30. Deias A, Magrini A. The impact of Equity Funding Dynamics on Venture Success: an empirical analysis based on Crunchbase Data. Economies. 2023;11(1):19.
31. O'Brien R, Ishwaran H. A Random forests quantile classifier for Class Imbalanced Data. Pattern Recogn. 2019;90:232–49.
32. Burez J, Van den Poel D. Handling Class Imbalance in customer churn prediction. Expert Syst Appl. 2009;36(3):4626–36.
33. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from Class-Imbalanced Data: review of methods and applications. Expert Syst Appl. 2017;73:220–39.
34. Leevy J-L, Khoshgoftaar T-M, Bauder R-A, Seliya N. A survey on addressing high-class Imbalance in Big Data. J Big Data. 2018;5(1):1–30.
35. Hasanin T, Khoshgoftaar T-M, Leevy J-L, Bauder R-A. J Big Data. 2019;6(1):1–25. Severely Imbalanced Big Data Challenges: Investigating Data Sampling Approaches.
36. Benchaji I, Douzi S, El Ouahidi B, Jaafari J. Enhanced Credit Card Fraud Detection based on attention nechanism and LSTM Deep Model. J Big Data. 2021;8:1–21.
37. Seliya N, Abdollah Zadeh A, Khoshgoftaar T-M. A literature review on one-class classification and its potential applications in Big Data. J Big Data. 2021;8:1–31.
38. Chandler G-N, Jansen E. The founder's self-assessed competence and venture performance. J Bus Ventur. 1992;7(3):223–36.
39. Cooper A-C, Gimeno-Gascon F-J, Woo C-Y. Initial human and Financial Capital as predictors of New Venture performance. J Bus Ventur. 1994;9(5):371–95.
40. Delmar F, Shane S. Does experience matter? The Effect of Founding Team experience on the survival and sales of newly founded ventures. Strategic Organ. 2006;4(3):215–47.
41. Kearney C, Hisrich R-D, Roche F. Public and Private Sector Entrepreneurship: similarities, differences or a combination? J Small Bus Enterp Dev. 2009;16(1):26–46.
42. Macmillan I-C, Block Z, Narasimha P-N-S. Corporate venturing: Alternatives, Obstacles encountered, and Experience effects. J Bus Ventur. 1986;1(2):177–91.
43. Toft-Kehler R, Wennberg K, Kim P-H. Practice makes Perfect: entrepreneurial-experience curves and Venture Performance. J Bus Ventur. 2014;29(4):453–70.
44. Engelmann J, Lessmann S. Conditional Wasserstein GAN-based Oversampling of Tabular Data for Imbalanced Learning. Expert Syst Appl. 2021;174:114582.
45. Esmaeilpour S, Liu B, Robertson E, Shu L. Zero-Shot Out-of-Distribution Detection based on the Pre-Trained Model Clip, In Proceedings of the AAAI Conference on Artificial Intelligence, 2022;36(6):6568–6576.
46. Xu D, Wu Y, Yuan S, Zhang L, Wu X. Achieving Causal Fairness through Generative Adversarial Networks, Presented at the Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019.
47. Murphy K-P. Machine learning: a probabilistic perspective. MIT Press; 2012.
48. Ngai E-W-T, Hu Y, Wong Y-H, Chen Y, Sun X. The application of Data Mining techniques in Financial Fraud detection: a classification Framework and an academic review of literature. Decis Support Syst. 2011;50(3):559–69.
49. Woo J-P. Concepts and Understanding of Structural Equations Model, Hannarae Academy, 2022.
50. Varian H. Machine learning and Econometrics. Slides Package Talk Univ. Wash; 2014.
51. Ross G, Das S, Sciro D, Raza H. CapitalVX: a machine learning model for Startup Selection and Exit Prediction. J Finance Data Sci. 2021;7:94–114.
52. Pan X, Zhang J, Song M, Ai B. Innovation resources Integration Pattern in High-Tech Entrepreneurial enterprises. Int Entrepreneurship Manage J. 2018;14:51–66.
53. Huang Z-A, Sang Y, Sun Y, Lv J. A neural network learning algorithm for highly Imbalanced Data classification. Inform Sci. 2022;612:496–513.
54. Wang H, Xiao Y, Su X, Li X, Team Social Media Usage and Team Creativity. The role of Team Knowledge sharing and Team-Member Exchange. Front Psychol. 2021;12:755208.
55. Mirzaei B, Nikpour B, Nezamabadi-pour H. CDBH: a clustering and Density-based Hybrid Approach for Imbalanced Data classification. Expert Syst Application. 2021;164:114035.
56. Tanha J, Abdi Y, Samadi N, Razzaghi N, Asadpour M. Boosting methods for Multi-class Imbalanced Data classification: an experimental review. J Big Data. 2020;7:1–47.
57. Ahmad A-K, Jafar A, Aljoumaa K. Customer Churn Prediction in Telecom using machine learning in Big Data platform. J Big Data. 2019;6(1):1–24.
58. Hasan M-N, Toma R-N, Nahid A-A, Islam M-M-M, Kim J-M. Electricity theft detection in Smart Grid systems: a CNN-LSTM Based Approach. Energies. 2019;12(17):3310.
59. Van Hulse J, Khoshgoftaar T-M, Napolitano A. Experimental Perspectives on Learning from Imbalanced Data, In Proceedings of the 24th International Conference on Machine Learning, 2007;935–942.

60. Kim H-Y, Lee W. On Sampling algorithms for Imbalanced Binary Data: performance comparison and some caveats. Korean J Appl Stat. 2017;30(5):681–90.
61. Chawla N-V, Bowyer K-W, Kegelmeyer W-PSMOTE. Synthetic minority over-sampling technique. J Artif Intell Res. 2022;16:321–57.
62. Santoso B, Wijayanto H, Notodiputro K-A, Sartono B. Synthetic Over Sampling Methods for Handling Class Imbalanced Problems: A Review. In IOP Conference series: earth and environmental science, vol. 58. 2017. pp. 1–8.
63. Krawczyk B. Learning from Imbalanced Data: Open challenges and future directions. Prog Artif Intell. 2016;5(4):221–32.
64. Seiffert C, Khoshgoftaar T-M, Van Hulse J. Improving Software-Quality Predictions with Data Sampling and Boosting. IEEE Trans Syst Man Cybernetics-Part A: Syst Hum, 39(6), 1283–94.
65. Bagui S, Li K. Resampling Imbalanced Data for Network Intrusion Detection Datasets. J Big Data. 2021;8:6.
66. Goodfellow I-J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative Adversarial Nets, Advances in Neural Information Processing Systems, 2014;27.
67. Kaliyev A, Zeno B, Rybin S-V, Matveev Y-N, Lyakso E-E. GAN Acoustic Model for Kazakh Speech Synthesis. Int J Speech Technol. 2021;24:729–35.
68. Yu T-K, Lin M-L, Liao Y-K. Understanding factors influencing Information Communication Technology Adoption Behavior: the moderators of Information Literacy and Digital Skills. Comput Hum Behav. 2017;71:196–208.
69. Li W, Fan L, Wang Z, Ma C, Cui X. Tackling Mode Collapse in Multi-generator GANs with orthogonal vectors, Pattern Recognition. Pattern Recogn. 2021;110:107646.
70. Chorev S, Anderson A-R. Success in Israeli High-Tech Start-Ups; critical factors and process. Technovation. 2006;26(2):162–74.
71. Barringer B-R, Jones F-F, Neubaum D-O. A quantitative content analysis of the characteristics of Rapid-Growth firms and their founders. J Bus Ventur. 2005;20(5):663–87.
72. Cassar G. Industry and startup experience on Entrepreneur Forecast performance in New firms. J Bus Ventur. 2014;29(1):137–51.
73. Cohen W-M, Levinthal D-A. Innovation and Learning: the two faces of R & D. Econ J. 1989;99(397):569–96.
74. Te Y-F, Wieland M, Frey M, Pyatigorskaya A, Schiffer P, Grabner H. Making it into a successful series a funding: an analysis of Crunchbase and LinkedIn Data. J Finance Data Sci, 2023;100099.
75. Kaggle. https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets/notebook
76. Burges C-J. A Tutorial on Support Vector machines for Pattern Recognition. Data Min Knowl Disc. 1998;2(2):121–67.
77. Chen T, Guestrin C, Xgboost. A Scalable Tree Boosting System, In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016;785–794.
78. Johnson J-M, Khoshgoftaar T-M. Survey on Deep Learning with Class Imbalance. J Big Data. 2019;6(1):1–54.
79. Pan C, Gao Y, Luo Y. Machine Learning Prediction of Companies' Business Success, CS229: Machine Learning, 2018.
80. Tripathi N, Seppänen P, Boominathan G, Oivo M, Liukkunen K. Insights into Startup Ecosystems through Exploration of Nulti-Vocal Literature. Inf Softw Technol. 2019;105:56–77.
81. Jo W, Kim D. OBGAN: Minority Oversampling Near Borderline with Generative Adversarial Networks. Expert Syst Appl. 2022;197:116694.

Publisher's note