IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version August 14, 2024.

Digital Object Identifier 10.1109/ACCESS.2024.xxxxxxx

Synthetic Data Generation and Evaluation Techniques for Classifiers in Data Starved Medical Applications

WAN D. BAE[1], SHAYMA ALKOBAISI[2], MATTHEW HORAK[3]*, SIDDHESHWARI BANKAR[1], SARTAJ BHUVAJI[1] SUNGROUL KIM[4], and CHOON-SIK PARK[5],
[1]Department of Computer Science, Seattle University, WA 98122 USA (e-mail: baew, sbankar, sbhuvaji@seattleu.edu)
[2]College of Information Technology, United Arab Emirates University, United Arab Emirates (e-mail: shayma.alkobaisi@uaeu.ac.ae)
[3]Amazon AWS Lambda, USA (e-mail: horakmatt@gmail.com)
[4]Department of ICT Environmental Health System, Graduate School, Soonchunhyang University, Asan, South Korea (e-mail: sungroul.kim@gmail.com)
[5]Department of Internal Medicine, Soonchunhyang Bucheon Hospital, South Korea (e-mail: mdcspark@hanmail.net)

Corresponding author: Shayma Alkobaisi (e-mail: shayma.alkobaisi@uaeu.ac.ae).
* The work for this manuscript was done prior to the author joining Amazon and is not related to his position at Amazon.

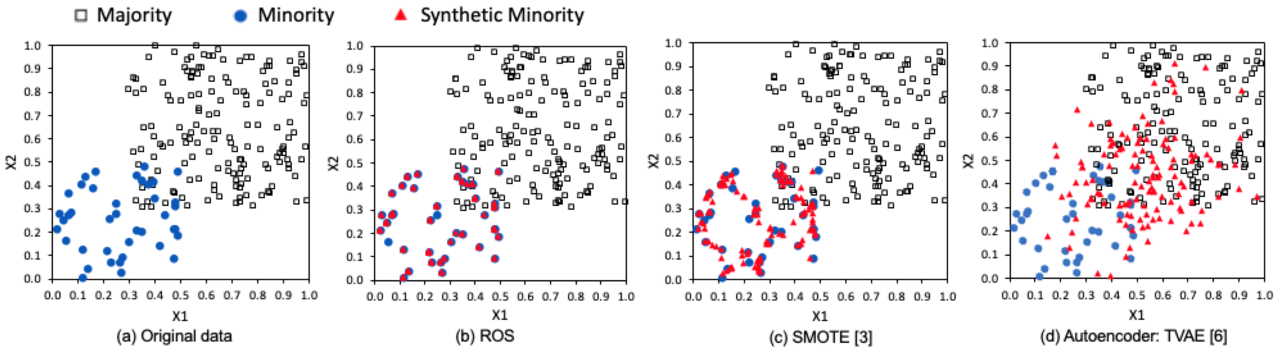
This study received support from United Arab Emirates University, UAE under UAEU-NFRP grant (Grant No. G00004281), the Ministry of Science and ICT, South Korea under National Research Foundation of Korea grant (NRF-2022R1A2C1010172), Seattle University, USA under Thomas Bannan Chair Engineering Award (11-0-1-480530) and Soonchunhyang University, South Korea.

ABSTRACT With their ability to find solutions among complex relationships of variables, machine learning (ML) techniques are becoming more applicable to various fields, including health risk prediction. However, prediction models are sensitive to the size and distribution of the data they are trained on. ML algorithms rely heavily on vast quantities of training data to make accurate predictions. Ideally, the dataset should have an equal number of samples for each label to encourage the model to make predictions based on the input data rather than the distribution of the training data. In medical applications, class imbalance is a common issue because the occurrence of a disease or risk episode is often rare. This leads to a training dataset where healthy cases outnumber unhealthy ones, resulting in biased prediction models that struggle to detect the minority, unhealthy cases effectively. This paper addresses the problem of class imbalance, given the scarcity of training datasets by improving the quality of generated data. We propose an incremental synthetic data generation system that improves data quality over iterations by gradually adjusting to the data distribution and thus avoids overfitting in classifiers. Through extensive experimental assessments on real asthma patients' datasets, we demonstrate the efficiency and applicability of our proposed system for individual-based health risk prediction models. Incremental SMOTE methods were compared to the original SMOTE variants as well as various architectures of autoencoders. Our incremental data generation system enhances selected state-of-the-art SMOTE methods, resulting in sensitivity improvements for deep transfer learning (TL) classifiers ranging from 4.01% to 7.79%. Compared with the performance of TL without oversampling, the improvement achieved by the incremental SMOTE methods ranged from 27.18% to 40.97%. These results highlight the effectiveness of our technique in predicting asthma risk and their applicability to imbalanced, data-starved medical contexts.

INDEX TERMS autoencoders, class imbalance problem, control coefficient, data starved contexts, rare event prediction, synthetic minority oversampling technique, transfer learning

I. INTRODUCTION

Machine learning has gained popularity recently with a wide range of applications ranging from fraud detection and weather prediction to online products recommendations. In healthcare, machine learning has demonstrated promising potential in transforming the future of health management. Predictive models have begun to play a crucial role in early disease prediction and enhancement of patience care and health management by providing early forecasts to enable preventative care [1]. For example, 63% of annual asthma costs have been attributed to avoidable asthma attacks that could have been prevented given an advance health care



(a) Original data (b) ROS (c) SMOTE [3] (d) Autoencoder: TVAE [6]

FIGURE 1: Data visualization: original data vs. ROS vs. SMOTE vs. Autoencoder

capable of predicting the risks of asthma attacks, so advance warning of these episodes would significantly reduce health cost [2], [3].

However, building robust predictive models in healthcare often faces the challenge of class imbalance within medical datasets. Class imbalance occurs when certain health conditions, demographics, or outcomes are rare or underrepresented, resulting in the minority class having significantly fewer samples than the majority class. While prediction accuracy is a valid metric for models trained on balanced data, relying solely on prediction accuracy is problematic when the model is trained on imbalanced datasets because the model can improve accuracy by biasing towards the majority class [4]. In certain medical contexts, it is of equal or of more importance to accurately predict rare events belonging to the minority class. For example, in predicting asthma attacks, which may represent only a small fraction, say 5 - 10%, of the data, accurately identifying these events is very important.

One possible approach to address class imbalance is to balance the classes by acquiring more minority data samples, but this is not be feasible in many medical applications. An alternative is to generate synthetic data while maintaining data quality and avoiding overfitting of the model. Common data augmentation techniques for balancing training data include synthetic minority oversampling techniques (SMOTE) and its variants, autoencoders, and generative adversarial networks (GAN).

Despite their success in applications with ample training data, these techniques are under-developed in data-starved contexts, which is the main technical deficiency in the field of machine learning for medical applications that this paper addresses.

A. MOTIVATION

A wide range of solutions to the class imbalance problem exist with SMOTE-based methods and autoencoders generally showing the strongest results and promise in most applications. However, up to now their success in data-starved contexts such as medical applications with few daily observations per patient has been limited. In our asthma risk prediction modeling, the average dataset size per individual patients is 168, with an average imbalance ratio of 3.98, where the imbalance ratio = $\frac{N_{MA}}{N_{MI}}$, N_{MA} and N_{MI} are the numbers of majority and minority class samples, respectively.

The large imbalance ration means that a high volume of synthetic data samples is needed to rebalance the datasets, which may degrade the performance of models trained on data augmented with traditional SMOTE methods and autoencoders. This motivates our meta-algorithmic approach that improves the quality of generated synthetic data by complementing SMOTE-based data-level improvements with a selection step based on the performance changes of a model trained on incrementally augmented datasets. Our objective is to develop a new way to generate synthetic data that is diverse but stays in a similar data distribution of the original data.

Figure 1 illustrates the overall pattern of the synthetic data generated by three data generation methods: a simple random over-sampling (ROS), SMOTE [5], and an autoencoder TVAE [6]. We used a small synthetic dataset containing 200 data points (x_1 and x_2 values in [0, 1] from two slightly different uniformly random distributions, one for each class). Scatter plot (a) shows the original data and its imbalance ratio is 4.0. Scatter plot (b) shows that the generated data generated by ROSare overlapped with the existing data. The duplicated data ratio was 52.21% in our analysis and this lack of data diversity would lead to model overfitting.

The augmented dataset in (c) demonstrates the data augmented by the SMOTE method. Visually, we see that the class boundary of the synthetic data follows a similar pattern to the original data but more diverse configuration of the data comparing to the data generated by ROS. On the other hand, synthetic data generated by TVAE (trained with 200 epochs) deviate significantly from the original minority data. As shown in (d), approximately a half of the generated data were far outside of the class boundary with a few outliers.

B. OUR CONTRIBUTIONS

In this paper, we address the class imbalance problem, focusing on predicting individual patients' health risk in data-starved medical applications. We propose an incremental synthetic data generation system that enhances the quality of synthetic data generated any arbitrary data augmentation method by introducing a selection step based on the performance changes of a model trained on incrementally augmented datasets. Because of the limited data available for individual patients, we use two models to test the incremental algorithm, first is the transfer learning model proposed in [7] and second is a set of conventional classifiers. SMOTE methods and autoencoder methods are among the top performing and most popular data augmentation tools today, so we compare the performance results of classifiers trained on data generated by our incremental system with results of classifiers trained on data enhanced by conventional SMOTE methods and four different autoencoder architectures.

Model performance results and data-level metrics demonstrate that the incremental synthetic data generation system enriches diversity within the feature space while preserving the original data distribution. We achieve this by iteratively improving the quality of the augmented training data based on performance of the classifier trained on it. This forces the algorithm to focus on diversifying the minority data without introducing new data-level correlations or patterns not supported by the true data. We believe that this periodic focus on the end classifier is the key factor that leads to improvements in classifiers trained on the boosted training data. While we demonstrate its effectiveness with state-of-the-art SMOTE variants paired with conventional classifiers and with transfer learning (TL) based classifiers for asthma patient health risk prediction, it is important to note that this system is easily used with other data generation methods and other classifiers.

The remainder of the paper is organized as follows: Section II provides a comprehensive review of the related literature, focusing on synthetic data generation methods such as SMOTE and autoencoders. In Section III, we present our approaches to rebalancing training data containing rare health events. Section IV evaluates our proposed approaches on asthma patients' datasets and presents the metrics and experimental results. Finally, Section V discusses our findings and concludes with potential future directions for research.

II. BACKGROUND

There are generally four main techniques to balance a dataset: cost-sensitive classification, one-class classification, resampling, and mixed approaches [8], [9]. In cost-sensitive classification, weights or costs are associated with labeling each event and more weight is given to the minority class with the goal of minimizing the expected cost. One-class classification can be used in imbalanced classification problems to detect a negative case as a normal event belonging to the majority class and a positive case as an outlier belonging to the minority class. Two resampling techniques can be used to solve class imbalance; oversampling to the minority class or undersampling of the majority class. Finally, mixed approaches that utilize two or more strategies can be used to alleviate class disparity.

As the scope of this paper is to investigate oversampling techniques for mitigating data imbalance problem through synthetic data generation, our review of related literature primarily addresses the details of SMOTE variants, autoencoders methods and boosted classification algorithm with SMOTE variants.

A. SMOTE-BASED METHODS

The most common data augmentation techniques for balancing training data are Synthetic minority oversampling (SMOTE) and its variants [5], [10]–[13]. provide a While the specific function varies depending on the implementation, SMOTE methods attempt to balance classes by adding high-quality synthetic data to the minority class. They achieve this by randomly selecting a data point and one of its k-nearest neighbors and then creating a new point somewhere in between the two. This process is repeated as necessary to achieve a more balanced distribution of classes. SMOTE successfully generates new data points that are sufficiently distinct from the originals to reduce overfitting, yet similar enough to improve the model's fit when trained on the augmented data. However, as the synthetic data is a linear combination of existing points, it has the potential to obscure underlying data distributions and blur boundaries between overlapping classes. Moreover, determining the nearest neighbors can become computationally intensive as the dataset grows. Addressing these limitations remains an active area of research [14].

To leverage nonlinear relationships in the minority data, the authors in [12] proposed Gaussian SMOTE (G-SMOTE) that also generates synthetic samples between minority samples using k-nearest neighbors and linear interpolation but utilizes the Gaussian (normal) distribution to generate new samples that deviate from the line, but not so far that it degrades performance. On the other hand, Gamma Distribution SMOTE (Gamma-SMOTE) [11] utilizes the gamma distribution to create new minority class points and produces data in a nonlinear fashion, thus giving rich geometric structure. Since the Gamma distribution is asymmetric, new minority points are generated close to the existing minority data sample.

Similar to SMOTE, Sample Density Distribution SMOTE (SDD-SMOTE) [13] generates synthetic samples similar to SMOTE [5] but considers the distribution of synthetic data points relative to the dense area of minority samples. The method reduces fuzzy classification boundaries and provides the practitioner with enhanced control over randomness of the SMOTE algorithm. It works by calculating the density of minority class samples, generating synthetic data points in regions of high density, and ensuring a balanced dataset. Specifically, it identifies the k-nearest neighbors of minority samples, measures their density, and generates synthetic samples with controlled coefficients to balance class distribution. SDD-SMOTE aims to improve the training of machine learning models by addressing the challenges posed by overlapping class distributions and the uncertainty of class boundaries expressed in the training data.

While SMOTE and its variants have shown effectiveness, there remains considerable room for improvement, particularly when applied to small training datasets. To address this challenge, researchers have proposed various boosting techniques by adopting general machine learning (ML) boosting techniques. However, these techniques improve model performance by enhancing the training algorithm for the models themselves without improving the underlying datasets on which the models are trained. Additionally, the ML community has recently begun to explore the potential of deep neural network-based approaches, such as autoencoders to address class imbalance in larger datasets.

B. DEEP NEURAL NETWORK BASED METHODS

With the rise in popularity of deep learning, the methods for data augmentation that use deep learning have also grown. The foremost proposed deep learning architecture for creating synthetic data is Generative Adversarial Networks (GAN) [15]. This approach employs two networks which compete against each other; a generator tries to create new data points from noise and a discriminator attempts to correctly distinguish between the generated data and the actual data. As both of these networks compete and improve, the generated data becomes closer to the original data. However, this architecture struggles with tabular data. To address this, [16] proposed conditional tabular GAN (CTGAN) which implements a number of measures to better reproduce the underlying distribution by evenly sampling any discrete attributes present in the original data set. However, GAN-based methods have shown only limited promise for model improvement on small imbalanced datasets, especially imbalanced tabular data.

Another approach to generate synthetic minority data involves utilizing autoencoders [17]–[19], which are neural networks designed to encode input data into a compressed representation and then reconstruct similar data from this representation. By forcing the data into the compressed representation, autoencoders encourage the neural net to learn key correlations between data features. This makes them suitable for synthetic data generation. Moreover, neural networks have the capability to extract complex relationships from the data, potentially preserving patterns that SMOTE may overlook.

Authors in [17] trained a variational autoencoder (VAE) to multiply the number of samples exhibiting power stealing among data set of power consumption curves. This gave their final model a substantial boost in detecting power stealing. Similarly, authors in [18] utilized an autoencoder in modeling the survival chance of COVID-19 patients. They had 300 samples from survivors and 20 samples from deceased patients, and they managed to train an autoencoder which generated 200 more minority samples to train on. These additional samples gave a modest increase to the specificity of their tested models with negligible impact on other scores. It is important to note however that the data used in both work are image data and hence additional work is necessary to prove the effectiveness of autoencoder on tabular data.

The work in [20], [21] explores using VAE to generate synthetic data and confirmed that the synthetic data confirmed to the original data distribution. Both work found increased sensitivity for the classifiers trained on their generated data. Using relatively large datasets, they were able to create abundant synthetic data.

An autoencoder to augment imbalanced datasets was presented in [19]. The work focused on clustering and data discretization to improve the performance and generalization of the proposed auto encoder. Their architecture outperformed traditional autoencoders and CTGAN on some datasets, but fell short on others. This approach was presented as an interesting tool that may be useful in some problems but needs to be used selectively and with comparison to traditional augmentation techniques on a case-by-case basis.

The authors in [22] trained a VAE to perform data augmentation. However, they focused not on unbalanced data but on data with a scarce number of labeled instances. Similarly, the authors in [23] employed a VAE to fill in known gaps of in their dataset. While neither of these approaches performed unsupervised minority class augmentation, they found that the augmented data did improve the performance of their end models. They demonstrated the potential of auto encoders to stretch limited tabular data and to improve the performance of models trained on this augmented data.

Overall, autoencoders focus on reconstruction loss, which never entirely disappears, so a well-constructed autoencoder produces novel outputs that conform to the same general distribution as the input data. This ability to generate data samples similar but not identical to the original makes autoencoders a viable candidate for synthetic data generation. Additionally, given the non-linear nature of neural networks, autoencoders can capture relationships that traditional techniques like SMOTE may overlook. Similar arguments apply to GANs. However, autoencoders and GANs are deep learning models, which require significant data and for which development has focused on non-tabular data formats, so there remains much work to be done to see how far these methods can be developed in the case of data-starved applications involving tabular data.

The GAN and autoencoders methods incrementally train the model but do not incrementally expand the training data. To address overfitting, they rely on standard deep learning techniques such as Dropout and early stopping. In contrast, our proposed incremental synthetic data generation system directly mitigates overfitting by incrementally expanding the training data, which effectively reduces the risk of overfitting through increased sample diversity.

C. BOOSTED CLASSIFICATION WITH SMOTES

Boosting techniques, such as integrating AdaBoost in the oversampling process is a natural way to enhance the performance of classifiers. Because of the popularity of SMOTE data augmentation, some of these methods have been specifically modified and tuned for use on SMOTE-augmented datasets. SMOTEBoost [24] is a combination of SMOTE and Boosting algorithm to improve the SMOTE algorithm [5]. SMOTE generates the synthetic samples for a particular class distribution while the boosting algorithm combines the weak learners' predicted outcomes to convert them into strong learners by assigning weights to the dataset instances and stressing misclassified cases. In a given scenario, a weak learner, i.e., AdaBoost, decision tree is first trained on the augmented data created by the SMOTE algorithm, and then the weights of the instances are adjusted iteratively, based on the misclassification rate until the predefined boosting iterations are finished.

Similar techniques boosting various combining SMOTE and other classifiers include SMOTE-SVM and AdaBoost-SVM. SMOTEBoost was evaluated for the regression-based task in addition to its use for classification [25] using 30 different datasets with evaluation of its four variants of SMOTEBoost. The main difference between this approach from SMOTE with AdaBoost is that it introduces the pre-processing steps and some additional approaches before the weak learner. The ranking evaluation method was adopted, which showed that the proposed approach is a better rank than AdaBoost and other variants. In [26], authors implemented a SMOTEBoost method for binary classification on imbalanced microarray based on two datasets, i.e., colon cancer and myeloma, by applying SMOTEBoost and the support vector machine (SVM) algorithm. The study results showed that SMOTEBoost with SVM outperforms SMOTE-SVM and AdaBoost-SVM in terms of geometric mean on both datasets.

The authors in [27] adopted the SMOTEBoost technique for a flood prediction model, applying SMOTE for synthetic data generation and AdaBoost for training purposes as usual but then further embedding a sparse Bayesian model with weight constraints for the flood prediction. This approach avoided the model overfitting and confirmed an improvement in the model performance. A similar work utilizing SMOTE and AdaBoost was introduced in [28]. The proposed technique used the weight adjustment on synthetic data to overcome the noise created by SMOTE. The study results demonstrated that SMOTE with a boosting method reduced performance degradation caused by the noisy area between the two classes.

All of the above techniques borrow general ML boosting techniques that improve model performance by focusing on the training or model construction process without directly addressing the quality of the underlying data. Our work focuses on a new algorithm-level approach to enhance the quality of synthetic data generated by arbitrary data augmentation techniques, especially SMOTE methods.

III. METHODS

In this section, we present the formal algorithms of our incremental data generation system. As mentioned above ‘‘Incremental boosting" typically refers to a technique where boosting algorithms are applied sequentially during model construction or training in multiple stages, with each stage building on the results of the previous one. The goal is to iteratively correct the mistakes made by previous learners and thereby improve the overall quality of the model, rather than enhancing the underlying training data, except through data augmentation.

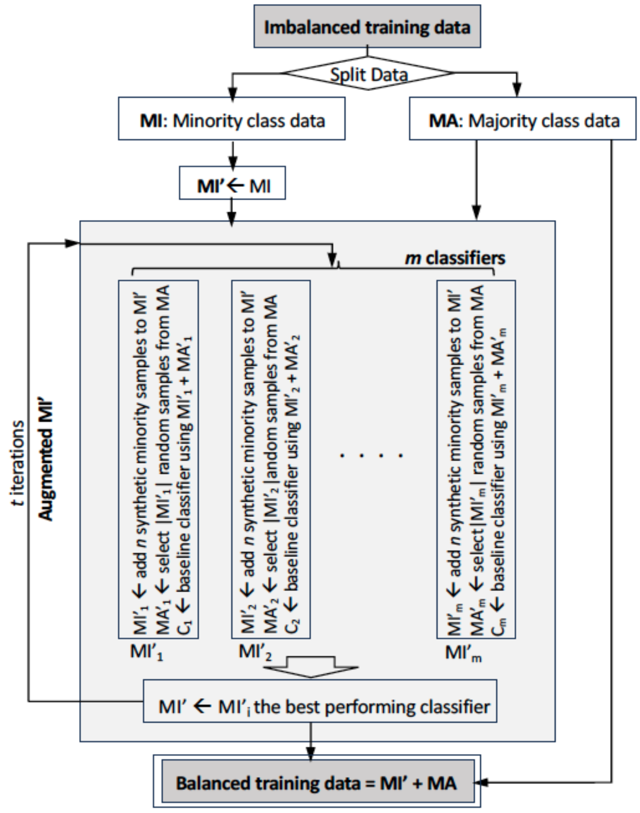

Imbalanced training data
Split Data
MI: Minority class data
MA: Majority class data
MI’← MI
t iterations
Augmented MI’
m classifiers
MI’1 ← add n synthetic minority samples to MI’
MA’1 ← select |MI’1| random samples from MA
C1 ← baseline classifier using MI’1 + MA’1
MI’2 ← add n synthetic minority samples to MI’
MA’2 ← select |MI’2| random samples from MA
C2 ← baseline classifier using MI’2 + MA’2
. . . .
MI’m ← add n synthetic minority samples to MI’
MA’m ← select |MI’m| random samples from MA
Cm ← baseline classifier using MI’m + MA’m
MI’1
MI’2
MI’m
MI’ ← MI’i the best performing classifier
Balanced training data = MI’ + MA

FIGURE 2: The incremental synthetic data generation system

In this context, incremental boosting is applied in an iterative process that gradually builds enhanced training data for minority classes to improve classification performance. It starts by generating a small set of synthetic samples using a method like SMOTE and adds these to the current training data. Multiple variations of the training dataset are then created, and classifiers are trained and evaluated on a separate validation dataset that was not involved in the data augmentation step. This process helps determine which subset of synthetic data contributes most to enhancing the model’s performance.

In each iteration, only the synthetic samples that result in the greatest improvement are retained for the next round, while less effective samples are discarded. The updated training data, now augmented with the best-performing synthetic samples, is used to generate new synthetic data. This iterative boosting process continues, gradually improving the training dataset and balancing it to better represent minority class instances. By repeating these steps until the desired volume of synthetic data is reached, the algorithm ensures data diversity, helps prevent overfitting, and strengthens the model’s ability to accurately predict minority class events.

A. INCREMENTAL SMOTE METHODS

For simplicity here, we phrase the iterative algorithm and examples in terms of a SMOTE-based data generation process,

Algorithm 1 *iSMOTE*$_{CC}$ (D_{train}, A, k)

```
Input: $D_{train}$ is a set of number of class-labeled training data points; $A$ is a SMOTE method; $k$: $k$-nearest neighbor used in $A$
Output: $D_{balanced}$ is a balanced training dataset augmented by synthetic data
Method:
Split $D_{train}$ into majority class dataset ($MA$) and minority class dataset ($MI$) and record their number of instances: $N_{MA} = |MA|$ and $N_{MI} = |MI|$
$N_{syn} \leftarrow N_{MA} - N_{MI}$
$t \leftarrow$ the number of iterations in the incremental data generation based on $N_{syn}$
$G \leftarrow$ Split $N_{syn}$ into $G$. $G$ is an array of size $k$, that will store the numbers of synthetic data that is generated for each iteration, roughly $n = \lceil \frac{N_{syn}}{t} \rceil$.
Initialization: $i \leftarrow 1$, $MI' \leftarrow MI$; $MI'$ is the augmented minority class data
repeat
    $n \leftarrow G[i]$, the size of synthetic data for iteration $i$
    $N \leftarrow MI' + n$, $N$ is the number of samples for both classes
    $MA' \leftarrow$ Randomly select $N$ number of majority samples from $MA$
    $MI' \leftarrow generateData(MI', MA', m, A, k)$
    $i \leftarrow i + 1$
until $i$ reaches $t$; all synthetic data specified in G are generated
$D_{balanced} \leftarrow MI' + MA$
return $D_{balanced}$
```

but the algorithm applies to any arbitrary data generation process. The system breaks down the synthetic data generation process into multiple iterations, as illustrated in Figure 2. The following are the steps for the incremental data generation:

1) **Create multiple training datasets**: Generate m different training datasets. For each training dataset, apply a SMOTE method to create n synthetic data samples for the minority class. Add these samples to the current minority class training data. Then, balance the dataset by combining the minority training data with an equal number of the majority class training data.
2) **Train classifiers**: Train m classifiers using the training datasets created in Step (1).
3) **Validate classifiers**: Assess the performance of each trained classifier on unseen validation data.
4) **Select the best subset of synthetic data**: Evaluate each classifier using a weighted evaluation function that focuses on correctly predicting minority class events. Then identify the subset of synthetic data that results in the best-performing classifier.
5) **Update training data**: Incorporate the selected subset of synthetic data into the minority class training dataset.

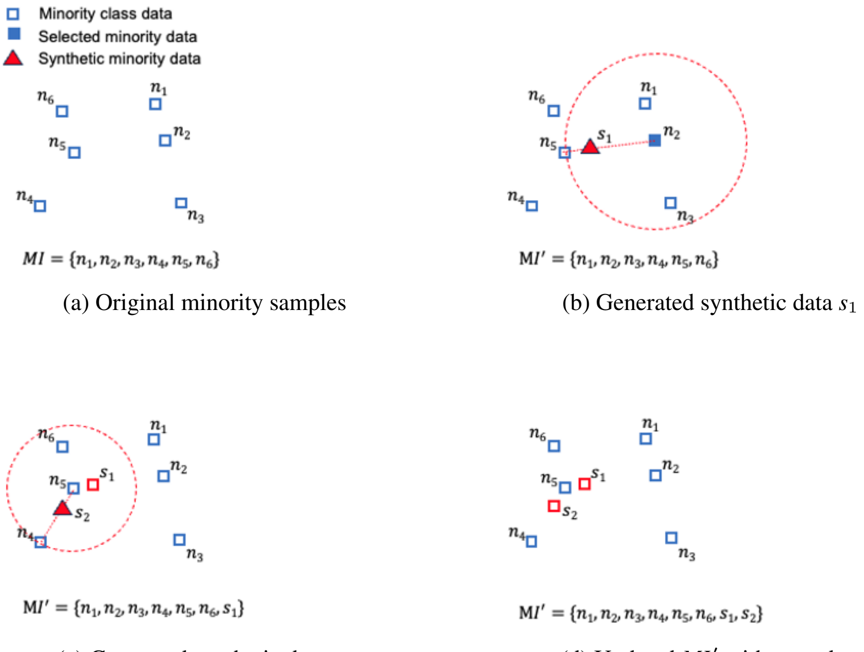
Algorithm 2 *generateData* (MI', MA', m, A, k)

```
Input: $MI'$ and $MA'$ are the current minority and majority class datasets ($|MI'| \approx |MA‘| - n$); $m$ is the number of classifiers; $A$ is a SMOTE method, $k$ is $k$ nearest neighbor used in $A$
Output: A minority dataset $MI'$ augmented by $n$ synthetic data
Method:
$S \leftarrow S$ is an array of size $m$ storing a synthetic dataset, initially assign $\{\}$ for all element in $S$
$C \leftarrow C$ is an array of size $m$ storing performance evaluation metrics, initially set to 0.
$i \leftarrow 1$
repeat
    $S[i] \leftarrow$ Generate $n$ synthetic data using a SMOTE $A$ with $k$ and the precalculated control coefficient
    $D'_{train} \leftarrow MI' + S[i] + MA'$
    $C[i] \leftarrow$ Train a classifier and evaluate the model using $D'_{train}$
until $i$ reaches $m$
$S_{best} \leftarrow$ Find a synthetic dataset in $S[i]$ that results in the top performing classifier $C[i]$.
$MI' \leftarrow MI' + S_{best}$, augment $MI'$ with the best synthetic dataset
return $MI'$
```

6) **Repeat the process (Steps through 5)**: In subsequent iterations, generate new synthetic minority class samples using the updated training data. Repeat this cycle until the total amount of synthetic data reaches the target level.

To select the optimal subset, we use a simple boosting technique with baseline classifiers such as decision trees and logistic regression to incrementally enhance classifier performance through the iterative process. By incorporating the best-performing synthetic data from previous iterations, the proposed system generates diverse synthetic data, which reduces overfitting in classification models and thus improves the classifier's performance. Details of the proposed system are described in Algorithms 1 and 2.

Figure 3 illustrates an example of the incremental data generation system that utilizes a SMOTE method [5] to augment the minority training data. Figure 3 (a) shows the original minority training dataset, $MI = \{n_1, n_2, n_3, n_4, n_5, n_6\}$, consisting of six data samples. Examples of the synthetic data generation for the first two data point are shown in (b) and (c). In (b), the minority data n_2 is selected as the first sample and n_5 is one of n_2's three nearest neighbors, and a synthetic data point s_1 is generated between n_2 and n_5. Then the system adds s_1 into MI resulting in the augmented minority training dataset MI'. Similarly, the second synthetic data point s_2 is generated by using n_5 and its neighbor n_4 and added into MI'. MI' in (d) is combined with the majority class dataset MA for training a classifier.


□ Minority class data
■ Selected minority data
▲ Synthetic minority data
n_1
n_2
n_3
n_4
n_5
n_6
s_1
s_2
$MI = \{n_1, n_2, n_3, n_4, n_5, n_6\}$
$MI' = \{n_1, n_2, n_3, n_4, n_5, n_6\}$
$MI' = \{n_1, n_2, n_3, n_4, n_5, n_6, s_1\}$
$MI' = \{n_1, n_2, n_3, n_4, n_5, n_6, s_1, s_2\}$

(a) Original minority samples

(b) Generated synthetic data s_1

(c) Generated synthetic data s_2

(d) Updated MI' with s_1 and s_2

FIGURE 3: An example of incremental data generation using SMOTE [5] with $k = 3$

B. SMOTE-SPECIFIC IMPROVEMENTS USED

We demonstrate and test our proposed system with four state-of-the-art SMOTE methods: SMOTE [5], Gaussian SMOTE (G-SMOTE) [12], Gamma Distribution SMOTE (Gamma-SMOTE) [11], and Sample Density Distribution SMOTE (SDD-SMOTE) [13]. The performance of each SMOTE method is compared with the proposed incremental system in Section IV.

To further enhance the SMOTE methods, we adopt the Control Coefficient (CC) from the SDD-SMOTE algorithm [13], which addresses the limitation of the uniform random function for synthesizing new samples. The CC value is calculated during data preprocessing and applied when the SMOTE$_{incrCC}$ method generates synthetic data points. A random probability distribution function is used for SDD and SDD-SMOTE methods. A Gaussian function is used for G-SMOTE, and a Gamma function is used for Gamma-SMOTE.

IV. EXPERIMENTS

A. DATASETS

Our experiments were conducted for individual-based health risk prediction using 20 asthma patients' datasets. All patients are non-smoking adults aged 45 to 83 years old, who participated in the ESCORT (environmental health smart study with connectivity and remote sensing technologies) study [29] between 28 December 2017 and 31 June 2018. The original study protocols related to our study were approved by the research ethics committee ofthe Soonchunhyang University (IRB No. 1040875-201608-BR-030); written informed consent was obtainedfrom all participants. Each patient's dataset consists of 27 min-max normalized variables including the patient' peak expiratory flow rate (PEFR), environmental exposure data (indoor/outdoor air quality) and behavioral data (home location, cooking habit and income level) along with a binary class label as 'class' [0, 1] representing a health risky day or not. Major variables and measurements are summarized in Table 1.

Patients' exposures to environmental variables were estimated using 24-hour time window at each AM/PM PEFR measurement. The high risk zone is defined as a PEFR below an individual patient's critical cutoff (PEFR$_C$). In consultation with medical practitioners, this cutoff was set at 20% quantile PEFR value of the patient's dataset in our experiments, with samples below PEFR$_C$ being minority samples (positive samples). Patients' datasets contain between 88 and 210 data samples (average 168), with minority class samples ranging from 16 to 38 (average 35) and majority class samples from 72 to 172 (average 132). The class imbalance ratio ranged from 2.32 to 5.52 (average 3.98) and this is higher than the ratios in the datasets used in the SDD-SMOTE [13].

B. EXPERIMENTS SETUP

In our experiments, we analyzed the performance of our incremental augmentation algorithm based on (1) classification model performance and (2) data-level quality analysis. The performance results of classifiers trained on data augmented with the incremental data generation system with the four SMOTE methods were compared to the results of the same classifiers trained on data augmented with the corresponding original SMOTE variants.

TABLE 1: Major variables and measurements in 20 asthma patients' datasets

Data Category		Variables	Measurement
Physiological data		yesterday's PEFRs	twice a day (am/pm)
Indoor	Air pollutants Meteorological data	$PM_{2.5}$, CO_2 temperature, humidity	every 60 sec. (remote sensors)
Outdoor	Air pollutants Meteorologic data	SO_2, CO, O_3 , NO_2, PM_{10} temperature, humidity, air pressure	every 30 min. Weather Center
Others:	Home location Life style Cooking habit	home distance from major roads income level frequency of frying	level 1 - level 5 level 1 - level 9 level 1 - level 7

TABLE 2: A summary of datasets

Dataset	# features	# samples	# minority class (MI)	# majority class (MA)	Imbalance ratio
20 asthma patients' datasets	27	88 - 210 (avg. 168)	16 - 38 (avg. 35)	72 - 172 (avg. 132)	2.32 - 5.52 (avg. 3.98)

TABLE 3: Autoencoder architectures

Methods	Layer	Architecture
Single Encoder	Encoder dense	[[8], [10], [12], [14], [16], [18] ,[20]]
	Bottle neck	[8, 10, 12, 14, 16, 18]
	Decoder dense	[[8, 10], [10, 12], [12, 14], [14, 16], [16, 18], [18, 20], [20, 22]]
Heavy Decoder	Encoder dense	[[10, 8], [12, 10], [14, 12], [16, 14], [18, 16], [20, 18], [22, 20]]
	Bottle neck	[8, 10, 12, 14, 16, 18]
	Decoder dense	[[6, 8, 10, 12], [8, 10, 12, 14], [10, 12, 14, 16], [12, 14, 16, 18], [14, 16, 18, 20], [16, 18, 20, 22], [18, 20, 22, 24]]
Balanced	Encoder dense	[[10, 8], [12, 10], [14, 12], [16, 14], [18, 16], [20, 18], [22, 20]]
	Bottle neck	[8, 10, 12, 14, 16, 18]
	Decoder dense	[[8, 10], [10, 12], [12, 14], [14, 16], [16, 18], [18, 20], [20, 22]]

* numbers in [ ] represent the number of neurons of the layer.

TABLE 4: Autoencoder and *TL* classifier training hyperparameters

Model	Autoencoders for synthetic data	TL classifiers
Optimizer	Adam (learning rate = 0.001)	Adam (learning rate = 0.001)
# epochs	200	100 - 1,000
Batch size	16	16
Validation split rate	0.25	0.20 (source model), 0.33 (target model)

We also compared SMOTE variants and incremental SMOTE methods with autoencoder-based data augmentation using four different architectures: (a) a single encoder, (b) a heavy decoder, (c) balanced autoencoders, and (d) a baseline autoencoder, TVAE [6]. For data-level quality assessment, we evaluated and contrasted the performance of SMOTE and incremental SMOTE methods with each other, as well as against the four autoencoder architectures and the TCGAN method [16].

All our data generation methods and classification algorithms were implemented in Python 3.8 and Keras framework, with data analysis and model performance evaluation conducted using scikit-learn.

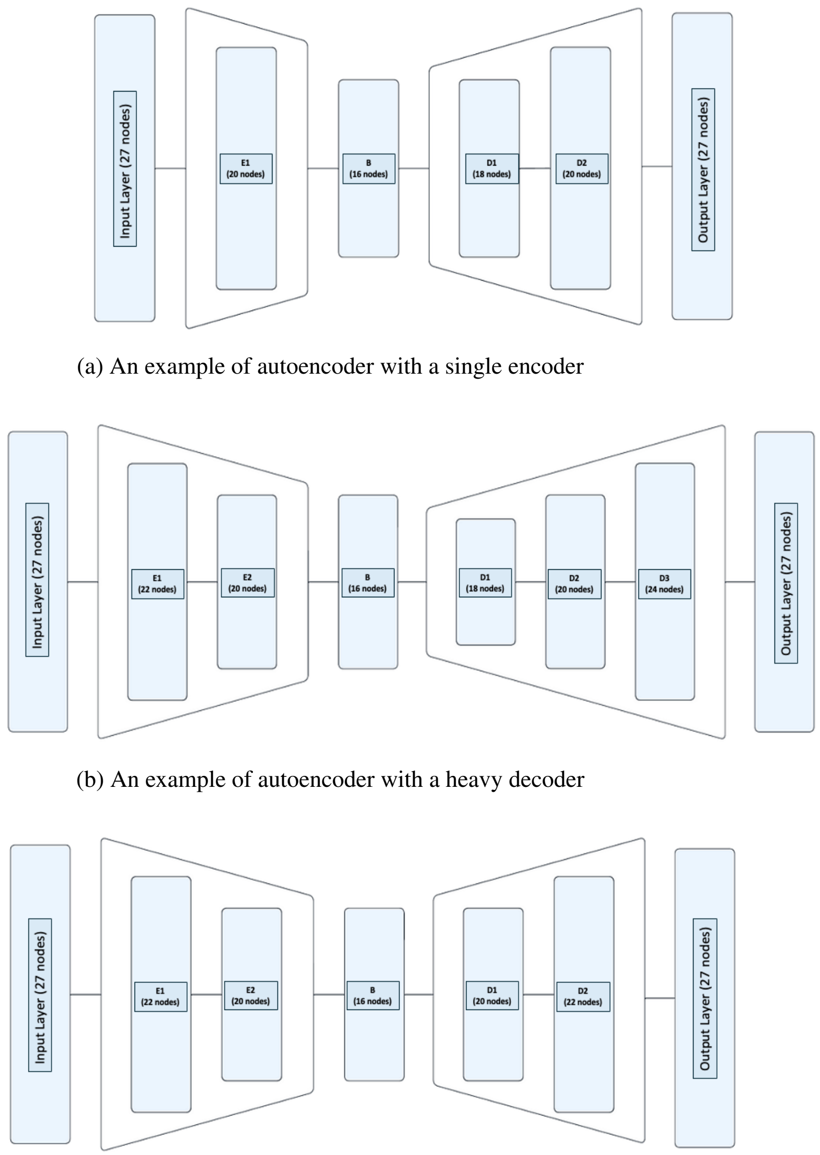
1) Autoencoder Training

The various network architectures of autoencoders we tested are shown in Table 3 and architecture examples of autoencoders are illustrated in Figure 4. Following best practices, model hyperparameters for autoencoders were selected through extended training and validation with k-fold cross validation (CV) to avoid overfitting and enhance performance. The second column in Table 4 lists the hyperparameters used for training the autoencoders: Adam optimizer (learning rate = 0.001), 200 epochs, 16 batch size, and a validation split rate of 0.25.

2) Classifier Training

The classification algorithms that we used for classifier performance based performance analysis of our algorithm are (1) four conventional classifiers - Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Naive Bayes (NB); and (2) TL-based classifiers.

Conventional classifiers used 3-fold CV. Each dataset was split into 80% for training/validation and 20% for testing. The training/validation dataset was augmented with synthetically generated data while the testing data retained the original imbalance ratio.



(a) An example of autoencoder with a single encoder

(b) An example of autoencoder with a heavy decoder

(c) An example of a balanced autoencoder

FIGURE 4: An overview of three autoencoder architectures

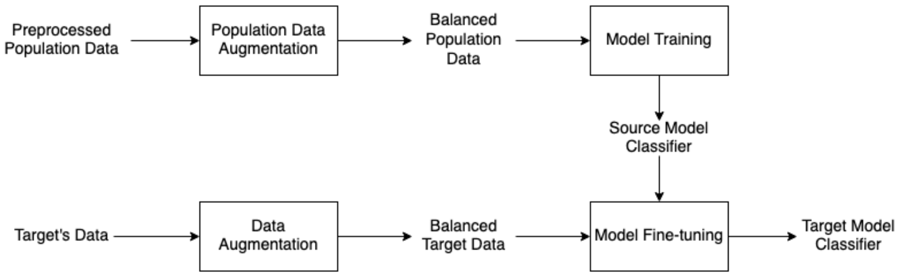


FIGURE 5: Transfer learning in classification

TABLE 5: Neural netwrok architectures in transfer learning

TL model	# hidden layer	architecture
`TL-1`	1	[[32]]
`TL-2`	2	[[10, 20, 32], [10, 20, 32, 48, 64, 128]]
`TL-3`	3	[[10, 20, 32], [6, 10, 20, 32], [4, 10, 20, 32]]
`TL-4`	4	[[10, 32], [32], [10, 20, 32], [10, 20, 32]]

TABLE 6: SMOTE variants in conventional classifiers

Classifier	Oversampling method	Accuracy	Sensitivity	Specificity	Precision	F_1 score	ROCAUC
DT	`No oversampling`	0.5801	0.2780	**0.8822**	0.5821	0.5663	0.5865
	`SMOTE`	0.5813	0.3663	0.7963	0.5726	0.5692	0.5843
	`G-SMOTE`	0.5731	0.3781	0.7680	0.5668	0.5577	0.5779
	`Gamma-SMOTE`	0.5849	0.3901	0.7797	0.5798	0.5676	0.5898
	`SDD-SMOTE`	**0.5955**	**0.3979**	0.7931	**0.5849**	**0.5772**	**0.5908**
KNN	`No oversampling`	0.5443	0.1279	**0.9607**	0.5575	0.5195	0.5665
	`SMOTE`	0.5988	**0.5297**	0.6679	**0.5828**	**0.5673**	0.5898
	`G-SMOTE`	**0.6016**	0.5241	0.6791	0.5798	0.5635	**0.5905**
	`Gamma-SMOTE`	0.5901	0.5261	0.6541	0.5665	0.5452	0.5860
	`SDD-SMOTE`	0.5974	0.5225	0.6722	0.5783	0.5615	0.5851
LR	`No oversampling`	0.5303	0.0762	**0.9844**	0.4656	0.4864	0.4865
	`SMOTE`	0.6168	0.5028	0.7308	0.5933	0.5852	0.5960
	`G-SMOTE`	0.6199	0.5260	0.7338	**0.5992**	**0.5884**	0.5969
	`Gamma-SMOTE`	**0.6215**	0.5195	0.7134	0.5918	0.5812	0.5970
	`SDD-SMOTE`	0.6180	**0.5290**	0.7089	0.5894	0.5805	**0.5997**
NB	`No oversampling`	0.5383	0.0987	**0.9778**	0.5024	0.5019	0.5680
	`SMOTE`	0.5992	0.3783	0.8201	0.5995	**0.5910**	**0.6023**
	`G-SMOTE`	0.5935	**0.3921**	0.7948	0.5880	0.5823	0.5998
	`Gamma-SMOTE`	0.5892	0.3761	0.8023	0.5838	0.5753	0.5901
	`SDD-SMOTE`	**0.6011**	0.3880	0.8141	**0.6016**	0.5876	0.5992

TL classifiers trained the source model using population data from 19 asthma patients (excluding the target patient) and then retrained the target model using the target patient' data. The target model subsequently predicted the patient's health risk. For the *TL* classifiers, we adopted the transfer learning framework outlined in [7], as illustrated in Figure 5. The TL classifiers first capture common patterns across all patients' training data and then adapt to the individual-specific characteristics of medical conditions during the retraining process. We evaluated various TL architectures for both the source and target models to identify the best-performing configuration. The optimal architecture, used for all *TL* classifiers, was `TL-3`, which featured a 3 hidden-layer structure: [[32], [20], [20]]. *TL* architectures are shown in Table 5.

First, we balanced and augmented population training data, excluding the individual's data in which the model is developed for. We then trained a deep model on this population data, creating a source model. Next, we prepared and augmented data specific to the target individual and retrained the source model using the target data, yielding the target model. Finally, the target model was tested on validation data (unaugmented imblanced data) and we analyzed the results of the target mode. The TL classifiers used 5-fold CV for the source model and 3-fold CV for the target model, with the Adam optimizer (learning rate = 0.001) and 100 to 1,000 epochs, as shown in Table 4.

3) Evaluation Metrics for Classification Performance

We evaluated classifiers using common binary classification metrics: (1) weighted accuracy, (2) sensitivity, (3) specificity, (4) precision average, (5) F_1-score average, and (6) Receiver Operating Characteristic Area Under the Curve (ROCAUC). While all these metrics are important, we focused on improving sensitivity, representing the model's ability to predict health risk correctly. The effectiveness of the incremental data generation system using each SMOTE-based method and the autoencoder-based augmentation were tested by comparing the above metrics for the four conventional classifiers trained on the variously augmented data.

In the following subsections, we present the performance of conventional classifiers with the existing SMOTE methods and the improvement achieved by the incremental data generation technique. We then present the performance improvement through achieved by the incremental data generation technique together with neural network-based transfer learning classifiers. Finally, we compare the performance of the incremental SMOTE methods and the autoencoders.

C. EVALUATION OF CLASSIFIERS

1) Conventional Classifiers: SMOTE vs. Incremental SMOTE

We conducted extensive experiments with four selected conventional classifiers on each SMOTE variant.

First, we present the average performance analysis of the existing SMOTE methods compared to no oversampling with the baseline classifiers. As shown in Table 6, all four SMOTE methods significantly outperformed the no oversampling in

TABLE 7: SDD-SMOTE and incremental SDD-SMOTE in conventional classifiers

Classifier	Method	Accuracy	Sensitivity	Specificity	Precision	F_1 score	ROCAUC
DT	SDD-SMOTE	0.5955	0.3979	**0.7931**	0.5849	0.5772	0.5908
	SDD-SMOTE$_{incrCC}$	**0.5987**	**0.4194**	0.7780	**0.5884**	**0.5783**	**0.5974**
		(+0.54%)	(+5.40%)	(-1.90%)	(+0.60%)	(+0.19%)	(+0.11%)
KNN	SDD-SMOTE	0.5974	0.5255	0.6722	0.5783	0.5615	0.5851
	SDD-SMOTE$_{incrCC}$	**0.6177**	**0.5287**	**0.7127**	**0.5904**	**0.5710**	**0.5976**
		(+3.40%)	(+0.61%)	(+6.02%)	(+2.09%)	(+1.69%)	(+2.13%)
LR	SDD-SMOTE	0.6180	0.5290	0.7089	0.5894	0.5805	0.5997
	SDD-SMOTE$_{incrCC}$	**0.6453**	**0.5447**	**0.7459**	**0.6097**	**0.6043**	**0.6178**
		(+4.42%)	(+2.97%)	(+5.22%)	(+3.44%)	(+4.10%)	(+3.02%)
NB	SDD-SMOTE	0.6011	0.3880	**0.8141**	0.5881	0.5876	0.5992
	SDD-SMOTE$_{incrCC}$	**0.6080**	**0.4095**	0.8065	**0.5951**	**0.5890**	**0.6070**
		(+1.15%)	(+5.54%)	(-0.93%)	(+1.19%)	(+0.24%)	(+1.30%)

TABLE 8: *TL* classifiers with SMOTE and autoencoders comparing to *TL* with no oversampling

Method		Accuracy	Sensitivity	Specificity	Precision	F1 score	ROCAUC
TL_{NN} with no oversampling		0.6484	0.4223	0.8750	0.6591	0.6390	0.6483
TL_{NN} with SMOTE	SMOTE	0.6697 (**+3.29**%)	0.5449 (**+29.03**%)	0.7846 (**-10.33**%)	0.6503 (**-1.34**%)	0.6489 (**+1.55**%)	0.6647 (**+2.53**%)
	G-SMOTE	0.6592 (**+1.67**%)	0.5461 (**+29.32**%)	0.7723 (**-11.74**%)	0.6308 (**-4.29**%)	0.6282 (**-1.69**%)	0.6592 (**+1.68**%)
	Gamma-SMOTE	0.6737 (**+3.90**%)	0.5505 (**+30.36**%)	0.7969 (**-8.93**%)	0.6508 (**-1.26**%)	0.6501 (**+1.74**%)	0.677 (**+4.43**%)
	SDD-SMOTE	0.6853 (**+5.69**%)	0.5720 (**+35.45**%)	0.7986 (**-8.73**%)	0.6556 (**-0.53**%)	0.6543 (**+2.39**%)	0.6853 (**+5.71**%)
TL_{NN} with Autoencoder	TVAE Synthesizer [6]	0.6653 (**+2.61**%)	0.5062 (**+19.87**%)	0.8233 (**-5.91**%)	0.6474 (**-1.78**%)	0.6461 (**+1.11**%)	0.6650 (**+2.58**%)
	Single Encoder (E20-B16-D18-20)	0.6674 (**+2.93**%)	0.5003 (**+18.47**%)	0.8343 (**-4.65**%)	0.6531 (**-0.90**%)	0.6410 (**+0.31**%)	0.6621 (**+2.13**%)
	Heavy decoder (E22-20-B16-D18-20-22)	0.6732 (**+3.82**%)	0.5002 (**+18.45**%)	0.8450 (**-3.43**%)	0.6594 (**+0.05**%)	0.6553 (**+2.55**%)	0.6731 (**+3.83**%)
	Balanced (E22-20-B16-D20-22)	0.6774 (**+4.47**%)	0.5413 (**+28.18**%)	0.8134 (**-7.04**%)	0.6530 (**-0.93**%)	0.6583 (**+3.02**%)	0.6774 (**+4.49**%)

weighted accuracy, sensitivity, precision, F_1 score and AUC ROC for all conventional classifiers. The highest specificity values were reported when no oversampling was applied as expected in imbalanced datasets.

Table 7 shows that the proposed system with SDD-SMOTE$_{incrCC}$ improved classifier performance on augmented data in all metrics except specificity, with improvements of 0.54%-4.42% in accuracy, 0.61%-5.54% in sensitivity, 0.60%-3.44% in precision, 0.19%-4.10% in F1 score, and 0.11%-3.02% in ROCAUC. Due to space limitations, we present results using SDD-SMOTE but note similar performance was achieved with other SMOTE methods.

2) TL Classifiers: SMOTE vs. Autoencoders

In this section, we present an analysis of the results comparing TL classifiers with different data augmentation techniques. Table 8 shows the performance results of *TL* classifiers enhanced with various data augmentation techniques, including SMOTE and autoencoders, to a baseline TL model without oversampling.

Using the SMOTE method improves the performance of *TL* models, with G-SMOTE increasing accuracy by 1.67% and sensitivity by 29.32%. However, it results in an 11.74% reduction in specificity, 4.29% in precision, and 1.69% in F1 score. This trade-off is observed in other SMOTE variants as well. SMOTE and Gamma-SMOTE show similar trends with sensitivity increases of 29.03% and 30.36%, respectively, and 2.53% and 4.43% rise in ROC AUC. The results indicate that these methods effectively address class imbalance, though they reduce specificity and precision. SDD-SMOTE shows the highest improvement achieved by increasing sensitivity by 35.45% and ROC AUC by 5.71%, but it results in a 8.73% reduction in specificity. This shows potential to enhance sensitivity while still maintaining a reasonable balance in overall classification performance.

Autoencoder-based methods show promising results; TVAE and Single Encoder models improve accuracy by 2.61% and 2.93%, and sensitivity by 19.87% and 18.47%, while precision and F1 score decline slightly. The Heavy Decoder model delivers a balanced improvement in accuracy (+4.82%), sensitivity (+18.45%), and ROC AUC (+3.83%). The Balanced Autoencoder model achieves the most significant gains, with a 4.47% increase in accuracy, 28.18% in sensitivity, and 4.49% in ROC AUC. It suggests the potential of a well-configured autoencoder to enhance performance on complex, imbalanced datasets.

TL classifiers trained using data augmented by SMOTE variants outperformed those augmented with autoencoder-generated data. This performance discrepancy can be attributed to the inherent limitations of autoencoders when

TABLE 9: SMOTEs vs. incremental SMOTEs in *TL* classifiers

Classifier	Method	Accuracy	Sensitivity	Specificity	Precision	F_1 score	ROCAUC
TL_{NN}	SMOTE	0.6697	**0.5449**	0.7846	0.6503	0.6489	0.6647
	SMOTE$_{incrCC}$	**0.6762**	0.5371	**0.8153**	**0.6599**	**0.6572**	**0.6762**
		(+0.97%)	(**-3.21**%)	(**+3.91**%)	(+1.48%)	(+1.26%)	(+1.72%)
	G-SMOTE	0.6592	0.5461	0.7723	0.6308	0.6282	0.6592
	G-SMOTE$_{incrCC}$	**0.6973**	**0.5886**	**0.8060**	**0.6549**	**0.6597**	**0.6973**
		(+5.79%)	(**+7.79**%)	(**+4.37**%)	(+3.81%)	(+5.01%)	(+5.79%)
	Gamma-SMOTE	0.6737	0.5505	0.7969	0.6508	0.6501	0.6770
	Gamma-SMOTE$_{incrCC}$	**0.6975**	**0.5726**	**0.8225**	**0.6708**	**0.6704**	**0.6975**
		(+3.54%)	(**+4.01**%)	(**+3.21**%)	(+3.08%)	(+3.12%)	(+3.04%)
	SDD-SMOTE	0.6853	0.5720	0.7986	0.6556	0.6543	0.6853
	SDD-SMOTE$_{incrCC}$	**0.6982**	**0.5953**	**0.7998**	**0.6591**	**0.6598**	**0.6975**
		(+1.88%)	(**+4.07**%)	(**+0.15**%)	(+0.54%)	(+0.84%)	(+1.78%)

working with small training datasets. Autoencoders rely heavily on large and diverse datasets to effectively learn meaningful latent representations. In scenarios where training data is limited, autoencoders face significant challenges in capturing the underlying data distribution, leading to poor generalization. The primary issue is that with a small dataset, the autoencoder may memorize the training data rather than learning to generalize from it, resulting in overfitting. On the other hand, SMOTE and its variants use simpler algorithms to synthesize new instances based on existing data points, which can be more effective in maintaining the statistical properties of the original dataset without succumbing to overfitting as readily as autoencoders do in data-scarce situations.

Overall, the results show the effectiveness of incorporating SMOTE variants and autoencoders into TL classifiers. These techniques improve sensitivity and overall classification capability, as reflected by higher F1 scores and ROC AUCs. However, there is a consistent trade-off with specificity, suggesting that while these models detect minority class instances more effectively, they may also yield higher false positive rates. Depending on the application's requirements, choosing between SMOTE-based methods or autoencoders will help achieve an optimal balance between sensitivity and precision. In our context of health risk prediction, gain in sensitivity is desirable with a moderate reduction in specificity and prediction.

3) TL Classifiers: SMOTE vs. Incremental SMOTE

This section presents the comparative performance of classifiers trained with incremental SMOTE versus those trained with existing SMOTE techniques in *TL* classifiers. The primary focus of this analysis is to evaluate how incremental data generation affects the classifiers' predictive capabilities and overall robustness.

Results presented in Table 9 indicate a consistent trend of improved classifier performance when using incremental SMOTE compared to traditional SMOTE. Classifiers trained with incremental SMOTE show higher accuracy, sensitivity, specificity, precision, F_1 score, and ROCAUC metrics across different SMOTE variations. SMOTE$_{incrCC}$ was the only method where a reduction in sensitivity was observed.

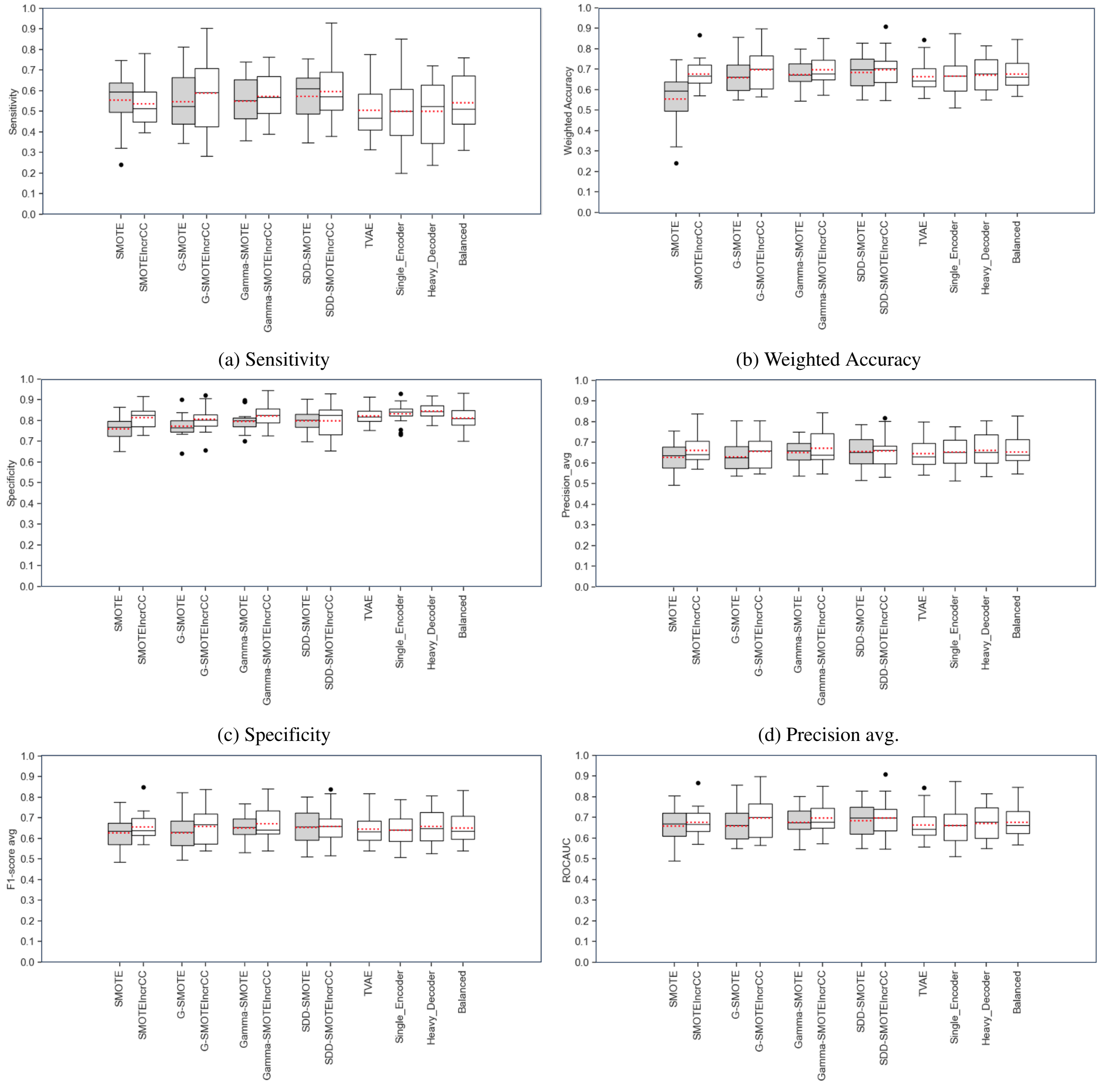
One of the most significant areas of improvement is observed in sensitivity and specificity. Incremental SMOTE led to notable gains in sensitivity for three incremental SMOTE methods, 7.79% increase for G-SMOTE$_{incrCC}$, 4.01% increase for Gamma-SMOTE$_{incrCC}$, and 4.07% increase for SDD-SMOTE$_{incrCC}$. In most class imbalance problems, there is trade-off between sensitivity and specificity. However, TL models using incremental SMOTE methods achieved gain in specificity: 3.91% for SMOTE$_{incrCC}$, 4.37% for G-SMOTE$_{omvtVV}$, 3.21% for Gamma-SMOTE$_{incrCC}$, and 0.15% for SDD-SMOTE$_{incrCC}$. In addition, incremental SMOTE methods led to more balanced increases across metrics such as precision, F_1 score and ROCAUC. For example, G-SMOTE$_{incrCC}$ achieved a 5.01% improvement in F_1 score and a 5.79% improvement in ROCAUC compared to G-SMOTE. This shows that incremental SMOTE enhances the robustness of classifiers and thus yields reliable performance improvements.

Figure 6 presents sensitivity values of TL_{NN} classifiers for 20 asthma patients' individual-level prediction using the proposed method and the original SMOTE methods. The values of median (solid lines) and mean (dotted red lines) of the data generated by the proposed methods are presented as well as the dispersion of the data. SDD-SMOTE$_{incrCC}$ performed the best in sensitivity but Gamma-SMOTE$_{incrCC}$ produced more tighter data distribution.

Incremental SMOTE methods provide a more effective approach to balancing the minority class, resulting in better generalization and overall performance, thereby improving the performance of *TL* classifiers. The results show that the classification of minority class instances was improved without compromising the accurate classification of majority class instances. This balanced enhancement is especially valuable in applications where both sensitivity and specificity are crucial for more reliable and comprehensive model outcomes.

4) Relevancy versus Population Size Trade-off in *TL*

One of the choices that must be made when developing a *TL* model is the selection of the source model. When making this selection one must often weigh the trade-off between using a high quality source model whose training data or target task is less related to the target model's task and using a



(a) Sensitivity

(b) Weighted Accuracy

(c) Specificity

(d) Precision avg.

(e) F_1 score avg.

(f) ROCAUC

FIGURE 6: Incremental SMOTE vs. autoencoders in TL_{NN} classifiers in sensitivity (20 patients' individual models) (dotted and solid lines represent mean and median, respectively)

TABLE 10: Performance of *TL* classifiers with varying source model training data

Training data for source model		Accuracy	Sensitivity	Specificity	Precision avg.	F_1 score avg.	ROCAUC
Grouping	*10 randomly selected patients	0.6377	0.5477	0.7278	0.5890	0.6004	0.6278
	G1 (9 patients)	0.7025	0.6189	0.7860	0.6430	0.6424	0.6780
	G2 (10 patients)	0.7125	0.6134	0.8115	0.6666	0.6665	0.7094
	Average	**0.7075**	**0.6161**	**0.7988**	**0.6548**	**0.6545**	**0.6937**
All 19 patients except the target patient		0.6982	0.5953	0.7998	0.6591	0.6598	0.6975

Notes: * Average results of 100 trials for 10 randomly selected patients. Incremental data generation method: SDD−SMOTE$_{incrCC}$

lower quality source model whose task is closer to that of the target model. The work in [30] showed that in the domain of image classification for cancer screening, the former is better, and they hypothesize that this is because the relatively high specialization of medical images used in training the closely related source model causes relatively little of what is learned by the source model to be directly relevant to the target task.

We performed a similar study by grouping patients into two sub-populations with similar environment and lifestyle characteristics such as income, cooking style, and home location. Each group had only 9 or 10 patients, but the patient-to-patient variance in the sub-populations is much smaller than in the full population. For each of these sub-populations, $G1$ and $G2$, we trained a source model and retrained the model using the target patient's data. We also develop *TL* classifiers using a randomly selected sub-population of size 10 for a source model and retrained it for the target patient. The performance results of *TL* classifiers with $G1$ and $G2$ were compared to those of *TL* classifiers with the training data of 10 random patients as well as the performance of *TL* classifiers with the full population. The results are presented in Table 10.

TL classifiers using $G1$ and $G2$ show greater improvement compared to *TL* classifiers trained with data from randomly selected patients, with improvements in all metrics: 18.79% in weighted accuracy, 17.97% in sensitivity, 9.76% in specificity, 11.17% in precision, 9.01% in F_1 score, and 10.50% in ROCAUC. This trend contrasts with the findings in [30] for image classification, which may be attributed to two main factors. First, the source models in this study may be of lower quality due to the relatively small size of even the full population dataset. Second, the source task and target task may be more similar in this case, and a significant amount of information learned by the specialized source model is useful to the target model.

The performance of the *TL* classifier with $G1$ and $G2$ was comparable to that with the full population, showing slightly higher weighted accuracy and sensitivity but slightly lower results in other metrics. Given that the sub-populations are only half the size of the full population, these findings suggest that better subgrouping could enhance the performance of *TL* classifiers.

D. DATA-LEVEL EVALUATION

1) Data-level Evaluation Metrics

Synthetic data generation for rebalancing classes focuses on three factors: (1) maintaining similar probability density functions of variables within the augmented training dataset, (2) preserving class boundaries, and (3) increasing data diversity to reduce overfitting of models trained on the augmented data. Data diversity refers to a robust synthetic data set that is sufficiently representative of the original data to prevent biasing so that it works well with various classification algorithms. In our experiments, we used several standard statistical metrics to analyze the synthetic data generated by the SMOTE-based methods and autoencoders. The metrics used for measuring the quality of the generated data are as follows:

Mean and standard deviation analysis: Factor (1) is measured by the difference between the means of the original and generated datasets, with a smaller difference indicating better maintenance of data distribution. Factor (2) is assessed by the difference in standard deviations (STD) because a large increase in STD typically suggests boundary distortions.

Density estimation analysis: Factor (3) is measured by probability density functions, such as Kullback-Leibler (KL) divergence and the difference in areas in the Kernel density estimation (KDE) distributions. The KL divergence is a mathematical measure that quantifies the difference between two probability distributions. The KL value is zero indicates a perfect match between the synthetic data and real data. Conversely, a higher KL divergence value suggests greater dissimilarity between the distributions. On the other hand, the KDE is the process of estimating an unknown probability density function by applying a kernel function to each data point, and it sums the diverging areas of the two distributions. Visualizing KDE distribution is also helpful for gaining insights into the generated data. Both KL divergence and KDE are popular metrics for measuring synthetic data quality, but it is uncertain what level of divergence in KL and KDE best enhances the performance of classifiers. Therefore, the values of these metrics must be tracked and tuned specifically for each application, but their trends can be used to indicate the degrees to which synthetic data from different methods matches the actual data relative to each other.

Gretel Score: "Gretel" [31] is an open platform that assesses the overall quality of synthetically generated data. Gretel measures data quality by comparing the distributional distance between the principal components in the synthetic and original data, with closer principal components indicating higher data quality. As with the KL divergence and KDE areas, it is unlikely that there is a universal optimal score for all applications. In fact, to our knowledge, no formal study has been conducted to verify the relationships between "Gretel" scores and prediction results.

To begin to address the gap in the community's understanding of the relationship of the KL divergence score, KDE are and Gretel score with classifier performance, we performed a lengthy evaluation of the correlation of these metrics with the performance of the classifiers analyzed in this paper. The results are presented in Section IV-D2.

2) Statistical Analysis

This section presents a statistical analysis of the data generated by the proposed incremental SMOTE methods, comparing to the existing SMOTE variants and the autoencoders. A summary of the analysis is provided in Table 11.

The data generated by ROS shows the lowest mean and STD difference from those in the original data. This is mainly due to a high ratio of duplicated samples, which results in overfitting. The mean differences between the synthetic and original data using the incremental generation system are

TABLE 11: Statistical analysis on the generated synthetic data

Method		Mean diff.	STD diff.	KL	KDE area	Gretel score
SMOTE	SMOTE	**0.27%**	13.08%	0.0081	1.02	92.63
	SMOTE$_{incrCC}$	0.33%	**12.31%**	**0.0069**	**0.85**	**92.95**
	G-SMOTE	1.33%	14.17%	**0.0122**	**1.32**	**91.88**
	G-SMOTE$_{incrCC}$	**0.96%**	**10.32%**	0.0157	1.44	91.58
	Gamma-SMOTE	0.69%	14.73%	0.0116	1.17	90.25
	Gamma-SMOTE$_{incrCC}$	**0.56%**	**12.03%**	**0.0043**	**0.69**	**92.84**
	SDD-SMOTE	2.24%	12.95%	**0.0051**	**0.82**	**93.74**
	SDD-SMOTE$_{incrCC}$	**1.84%**	**11.03%**	0.0062	0.85	91.52
Autoencoder	TVAE Synthesizer [6]	3.84%	33.21%	0.0701	2.84	77.34
	Single Encoder	3.16%	**6.29%**	0.0324	1.57	84.64
	Heavy Decoder	**1.46%**	12.19%	0.0553	1.95	81.05
	Balanced	2.88%	7.89%	**0.0261**	**1.45**	**84.90**
GAN	CTGAN [16]	1.43%	1.52%	0.2783	4.90	68.88

relatively small, ranging from 0.33% to 1.84%. G-SMOTE, Gamma-SMOTE, and SDD-SMOTE reduced the mean values, while SMOTE increased it. All SMOTE methods reduced the STD values: 5.9% in SMOTE, 27.2% in G-SMOTE, 18.3% in Gamma-SMOTE, and 14.8% in SDD-SMOTE. The proposed system decreased KL divergence and KDE area in SMOTE and Gamma-SMOTE but increased them in G-SMOTE and SDD-SMOTE, a trend also seen in Gretel scores.

The percentages in mean difference between the original data and the synthetic data by the four autoencoders are higher than those generated by all SMOTE-based methods. On the other hand, the percentages in STD difference are within a wider range, between 6.29% and 33.21%. The values for the single encoder and balanced autoencoders are smaller than or comparable to those of the SMOTE-based methods. Similar data quality patterns can be observed in the Gretel scores and KL and KDE values. The proposed three autoencoders show lower Gretel scores and higher KL and KDE values compared to the SMOTE-based methods. In contrast, the data generated by TVAE shows the highest mean and STD differences from the original data, 3.84% and 33.21%, respectively. This resulted in the highest KL and KDE values and the lowest Gretel score.

Among the three proposed autoencoders, the balanced autoencoder is the most suitable choice for our dataset. The data quality produced by the balanced autoencoder is better than that produced by the single encoder and the heavy decoder methods. The balanced autoencoder likely maintains a better balance between model complexity and data representation for the dataset, resulting in better performance.

We also evaluated CTGAN, a GAN-based method outlined in [16] for its performance in generating synthetic data. While the mean and standard deviation differences between the CTGAN-generated data and the real data were minimal, 0.43% for the mean difference and 1.52% for the standard deviation, the method showed significant shortcomings in other statistical measures. The KL divergence was 0.2783, and the KDE area discrepancy was 4.0. These values were notably higher than those of other methods, leading to a low Gretel score of 68.88.

The inadequate performance of CTGAN can be attributed to its dependence on large and diverse training datasets to effectively learn robust latent representations. Like autoencoders, CTGAN struggles to accurately capture the true underlying distribution from a small training data. Consequently, this leads to synthetic data that fails to reflect the variability and complexity of the original data distribution, compromising its overall quality.

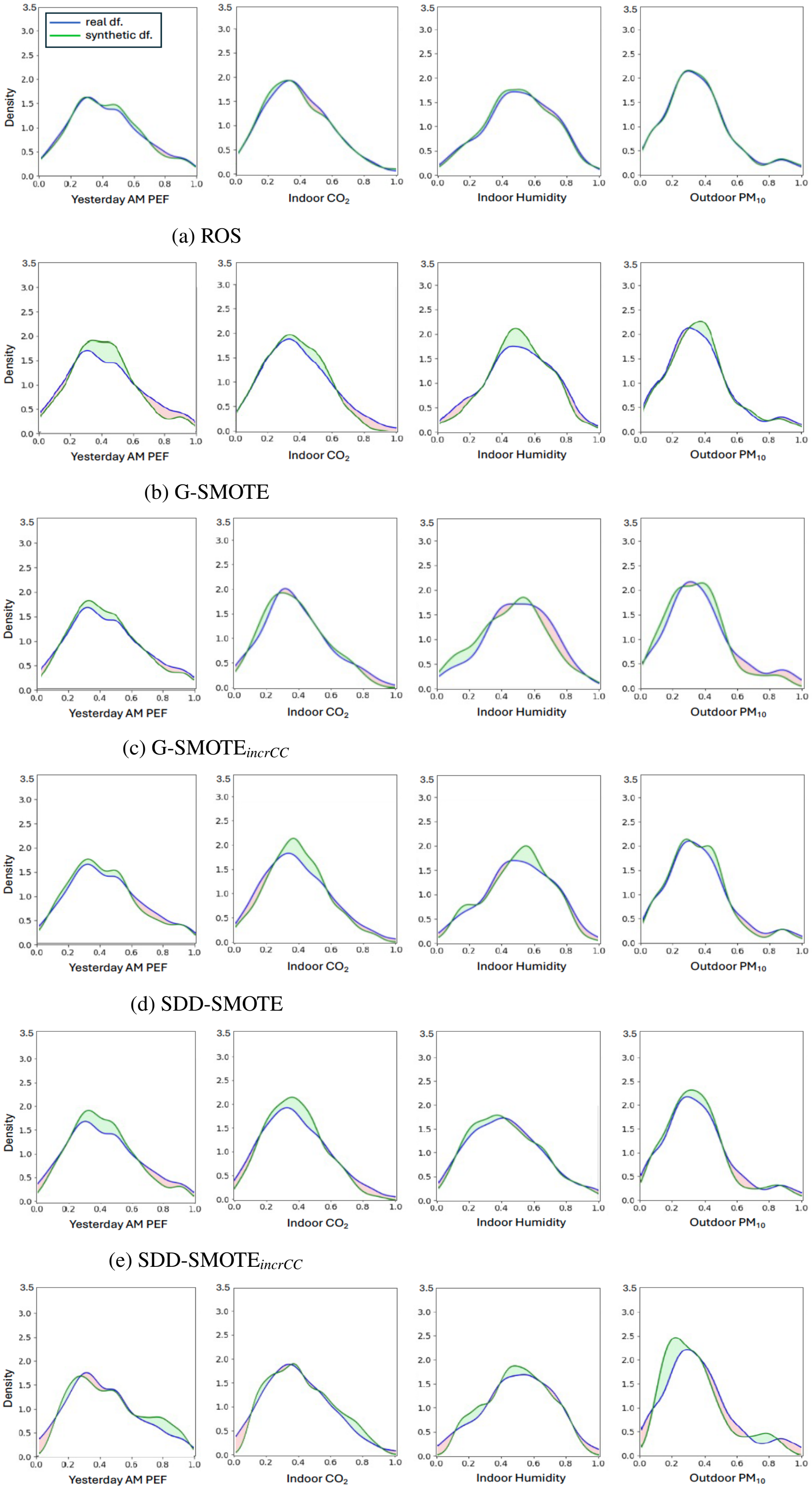
3) Visual Analytics

Data-level evaluation of synthetic data generation can be performed empirically by examining scatter-plots and visualizing KDE distributions.

Figure 7 illustrate the KDE divergence of four selected variables; patients' yesterday morning PEFR, indoor CO_2, indoor humidity, and outdoor PM_{10}, illustrating the distribution of the synthetic minority samples generated by ROS, two SMOTE variants (G-SMOTE and SDD-SMOTE) and their incremental SMOTE methods, and the balanced autoencoder, comparing to those of the real minority class samples for the four features.

The data distributions of the synthetic data generated by ROS are close to the real minority data distributions in all four features, while those generated by other methods present moderate-level diverging distributions which be seen as more data diversity in the synthetic data, and this can reduce overfitting in classification. As shown in (c), the divergence in the data by G-SMOTE$_{incrCC}$ is more distributed, following the distribution of the original data, than the divergence in the data by G-SMOTE. The similar pattern can be seen in KDE graphs in SDD-SMOTE in (c) and SDD-SMOTE$_{incrCC}$ in (d). While the KDE divergence of the data generated by the balanced autoencoder generally follows that of the real data for most data attributes, some autoencoder models like TVAE and single encoder generated the data with higher divergence.

In Figure 8, we present scatter plots that illustrate the overall pattern of the synthetic data generated by ROS, G-SMOTE and G-SMOTE$_{incrCC}$, SDD-SMOTE and SDD-SMOTE$_{incrCC}$, and the balanced autoencoder. We used the same synthetic dataset used in Figure 1. Scatter plots in (b)



(a) ROS

(b) G-SMOTE

(c) G-SMOTE$_{incrCC}$

(d) SDD-SMOTE

(e) SDD-SMOTE$_{incrCC}$

(f) Balanced autoencoder

FIGURE 7: Comparisons of KDE diverging area: ROS vs. SMOTE vs. incremental SMOTEs vs. autoencoder (selected variables: yesterday AM PEF, indoor CO_2 & humidity, and outdoor PM_{10})

and (c) illustrate the augmented data by the original SMOTE variants and scatter plots in (e) - (f) show the data by the incremental SMOTE methods. Visually, we see that the class boundaries of the synthetic data generated by these all of the SMOTE and incremental SMOTE methods methods follow a similar pattern to the original data but a slightly more diverse configuration of the data well inside the minority class boundary can be found in (e) and (f), which are by the incremental SMOTE methods. Overall, the visual difference in this synthetic example is less striking than differences in classifier performance on our real data sets.

V. DISCUSSION

In medical applications where predicting rare disease events or exacerbation is the main concern, class imbalance can significantly impact the accuracy of prediction models. Addressing this challenge is crucial for developing reliable and effective health management systems. In this paper, we proposed a meta-algorithm that incrementally boosts existing data generation methods. While the algorithm can be applied to any data set with class imbalance, we demonstrate that it is particularly effective data-starved applications, as many medical applications are. Although the incremental algorithm can be used with any base data generation algorithm, we demonstrate and test it with various SMOTE algorithms. When paired this way, we refer to the incremental algorithm as an *incremental SMOTE* method. We demonstrate that incremental SMOTE methods improved the quality and representativeness of training data and improved the performance and robustness of rare disease event classifiers trained on data augmented by the proposed algorithm.

We performed two main comparisons of the incremental algorithm with existing data generation algorithms. First, we compared our incremental SMOTE methods to the original SMOTE variants using real asthma patients' datasets with four conventional and TL-based classifiers. The findings demonstrate that the incremental SMOTE methods improved prediction accuracy by 4.01% to 7.79% in the sensitivity of TL models using three SMOTE variants. Results of our experiments show the effectiveness of the incremental data generation technique in addressing class imbalance in healthcare applications.

Models trained on data augmented with the proposed incremental SMOTE methods show the consistent improvement over those trained on data augmented with traditional SMOTE methods. However, in cases with very few minority samples, SMOTE-based methods may not generate enough non-duplicate data points to balance the dataset, and the incremental method is not able to address this problem. Open challenges include developing flexible, scalable SMOTE variants that are robust to different imbalanced ratios and small data sizes. Additionally, SMOTE methods are based on interpolation between known data points, so they do not generate any data point outside the class boundaries. It would be worthwhile to investigate the effect of generating data just outside the boundary within a threshold in the future.

The second comparison we did was to compare both the traditional SMOTE variants and the incremental SMOTE methods to various autoencoders, in which we have drawn valuable insights regarding their performance and suitability. The best-performing autoencoder model was the balanced model. Compared to the incremental SMOTE, the performance the balanced autoencoder model was slightly higher than SMOTE$_{incrCC}$ but was much lower than three other incremental SMOTE methods, which are G-SMOTE$_{incrCC}$, Gamma-SMOTE$_{incrCC}$, and SDD-SMOTE$_{incrCC}$. In our experiments of autoencoders, TVAE, the single encoder and heavy decoder models showed overfitting issues with small training data sizes, which led to a loss of generalization ability of the models.

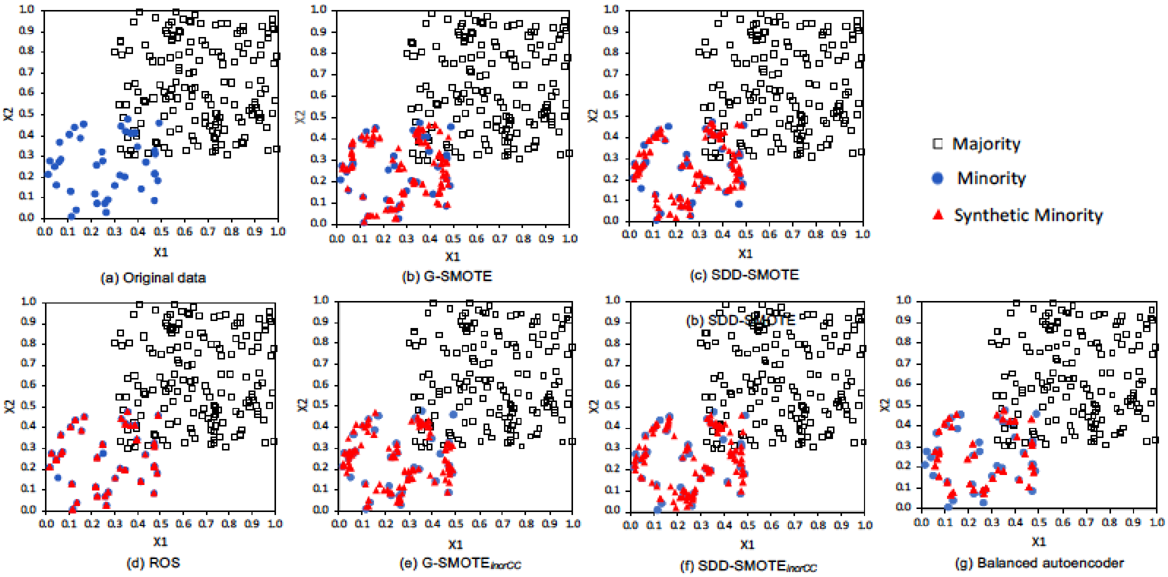
It is possible that the autoencoder models, with their increased number of layers (number of decoder layers in the single encoder model and number of both encoder and decoder layers in the heavy decoder model), might have been too large or overly complex to adequately encode into the compressed representation the specific patterns and characteristics present in our data. Our data-level evaluation on the autoencoders emphasizes the significance of architectural choices across different datasets and applications. By lengthy experimentation with various architectures and optimizing layer configurations and number of nodes in each layer, suitable autoencoder models for a particular application can often be developed. However, this process is time-consuming and prone to overfit the data on which the data generation model was developed. This is particularly important in data-starved contexts because protecting against overfitting generally requires holding out a significant amount of training data for validation and testing. Therefore our incremental SMOTE methods may be more effective and desirable when data is scarce.

We have also found that the performance of data generation methods can vary depending on the datasets and its specific features. Thus, it is important to select a data generation technique tailored to the size and characteristics of the dataset although robust and reliable data generation is desirable. We plan to further explore for various approaches including clustering-based approach to continue to yield valuable insights into synthetic data generation.

Finally, it is important to note that while this paper focuses on medical applications in its discussions and experiments, the proposed data generation techniques can be transferred to other applications that suffer from class imbalance with limited training datasets.

ACKNOWLEDGMENT

This study received support from United Arab Emirates University, UAE under UAEU-NFRP grant (Grant No. G00004281). The authors would like to thank the medical staff in the Division of Allergy and Respiratory Medicine at the Soonchunhyang University Bucheon Hospital for the provision of the data. All demographic and biodata were obtained from Soonchunhyang University Bucheon Hospital, a member of the Korea Biobank Network (KBN4-A06).



(a) Original data

(b) G-SMOTE

(c) SDD-SMOTE

(d) ROS

(e) G-SMOTE$_{incrCC}$

(f) SDD-SMOTE$_{incrCC}$

(g) Balanced autoencoder

FIGURE 8: Data visualization: Original data vs. ROS vs. SMOTE vs. incremental SMOTE vs. Autoencoder

REFERENCES

[1] C. S. Kelly, A. L. Morrow, J. Shults, N. Nakas, G. L. Strope, and R. D. Adelman, "Outcomes evaluation of a comprehensive intervention program for asthmatic children enrolled in medicaid," *Pediatrics*, vol. 105, no. 5, pp. 1029–1035, 2000.

[2] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health information science and systems*, vol. 2, no. 1, pp. 1–10, 2014.

[3] E. Forno and J. C. Celedón, "Predicting asthma exacerbations in children," *Current opinion in pulmonary medicine*, vol. 18, no. 1, p. 63, 2012.

[4] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Sciences*, vol. 513, pp. 429–441, 2020.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[6] MIT, "The synthetic data vault," accessed on October 4, 2023, https://sdv.dev.

[7] W. D. Bae, S. Kim, C.-S. Park, S. Alkobaisi, J. Lee, W. Seo, J. S. Park, S. Park, S. Lee, and J. W. Lee, "Performance improvement of machine learning techniques predicting the association of exacerbation of peak expiratory flow ratio with short term exposure level to indoor air quality using adult asthmatics clustered data," *Plos one*, vol. 16, no. 1, p. e0244233, 2021.

[8] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[9] Y. Zhao, Z. S.-Y. Wong, and K. L. Tsui, "A framework of rebalancing imbalanced healthcare data for rare events' classification: a case of look-alike sound-alike mix-up incident detection," *Journal of healthcare engineering*, vol. 2018, 2018.

[10] T. R. Hoens and N. V. Chawla, "Imbalanced datasets: from sampling to classifiers," *Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley*, 2013.

[11] F. Kamalov and D. Denisov, "Gamma distribution-based sampling for imbalanced data," *Knowledge-Based Systems*, vol. 207, p. 106368, 2020.

[12] H. Lee, J. Kim, and S. Kim, "Gaussian-based smote algorithm for solving skewed class distributions," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 4, pp. 229–234, 2017.

[13] Q. Wan, X. Deng, M. Li, and H. Yang, "Sddsmote: Synthetic minority oversampling technique based on sample density distribution for enhanced classification on imbalanced microarray data," in *2022 The 6th International Conference on Compute and Data Analysis*, 2022, pp. 35–42.

[14] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[16] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," *Advances in neural information processing systems*, vol. 32, 2019.

[17] X. Gong, B. Tang, R. Zhu, W. Liao, and L. Song, "Data augmentation for electricity theft detection using conditional variational auto-encoder," *Energies*, vol. 13, no. 17, p. 4291, 2020.

[18] F. Khozeimeh, D. Sharifrazi, N. H. Izadi, J. H. Joloudari, A. Shoeibi, R. Alizadehsani, J. M. Gorriz, S. Hussain, Z. A. Sani, H. Moosaei *et al.*, "Combining a convolutional neural network with autoencoders to predict the survival chance of covid-19 patients," *Scientific Reports*, vol. 11, no. 1, p. 15343, 2021.

[19] J. Jeong, H. Jeong, and H.-J. Kim, "An autoencoder-based numerical training data augmentation technique," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 5944–5951.

[20] Z. Islam, M. Abdel-Aty, Q. Cai, and J. Yuan, "Crash data augmentation using variational autoencoder," *Accident Analysis & Prevention*, vol. 151, p. 105950, 2021.

[21] T.-T.-D. Nguyen, D.-K. Nguyen, and Y.-Y. Ou, "Addressing data imbalance problems in ligand-binding site prediction using a variational autoencoder and a convolutional neural network," *Briefings in Bioinformatics*, vol. 22, no. 6, p. bbab277, 2021.

[22] J. Fang, C. Tang, Q. Cui, F. Zhu, L. Li, J. Zhou, and W. Zhu, "Semi-supervised learning with data augmentation for tabular data," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3928–3932.

[23] C. R. Wewer and A. Iosifidis, "Improving online non-destructive moisture content estimation using data augmentation by feature space interpolation with variational autoencoders," in *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. IEEE, 2023, pp. 1–7.

[24] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer, 2003, pp. 107–119.

[25] N. Moniz, R. Ribeiro, V. Cerqueira, and N. Chawla, "Smoteboost for regression: Improving the prediction of extreme values," in *2018 IEEE 5th international conference on data science and advanced analytics (DSAA)*. IEEE, 2018, pp. 150–159.

[26] R. F. W. Pratama, S. W. Purnami, and S. P. Rahayu, "Boosting support vector machines for imbalanced microarray data," *Procedia computer science*, vol. 144, pp. 174–183, 2018.

[27] Y. Wu, Y. Ding, and J. Feng, "Smote-boost-based sparse bayesian model for flood prediction," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–12, 2020.

[28] F. Sağlam and M. A. Cengiz, "A novel smote-based resampling technique trough noise detection and the boosting procedure," *Expert Systems with Applications*, vol. 200, p. 117023, 2022.

[29] J. Woo, G. Rudasingwa, and S. Kim, "Assessment of daily personal pm2. 5 exposure level according to four major activities among children," *Applied Sciences*, vol. 10, no. 1, p. 159, 2020.

[30] A. Menegola, M. Fornaciali, R. Pires, F. V. Bittencourt, S. Avila, and E. Valle, "Knowledge transfer for melanoma screening with deep learning," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, 2017, pp. 297–300.

[31] Gretel, "Gretel," accessed on October 4, 2023, https://gretel.ai/.

WAN D. BAE received her B.S. in Architectural Engineering from Yonsei University, South Korea in 1989, followed by her M.S. and Ph.D. in Computer Science from the University of Denver in USA in 2004 and 2007, respectively. She served as an associate professor at the University of Wisconsin–Stout from 2011 to 2019 before joining Seattle University, where she is currently a professor in the Department of Computer Science and holds the Thomas Bannan Endowed Chair of Engineering. From 2015 to 2017, she was a visiting professor at United Arab Emirates University in United Arab Emirates, and Hanyang University in South Korea. She has authored two book chapters and over forty research papers published in international journals and conference proceedings. Her research focuses on spatial and spatio-temporal databases, data mining, health informatics, mobile computing and optimization, big data analytics, and GIS. She is a member of the ACM and IEEE.

SHAYMA ALKOBAISI is currently an associate professor at the College of Information Technology (CIT) in the United Arab Emirates University (UAEU). She received her Ph.D. degree in Computer Science from the University of Denver in June 2008. In 2009, she was selected as a member of the Advisory Council to the Science and Technology Program at Emirates Foundation for Philanthropy, Abu Dhabi, and in 2010 she was appointed manager of that program for one year. She served in different admin positions including Department Chair, Acting Dean and the Vice Dean of CIT for several years. She also held the position of Acting Dean of the University College at UAEU, 2021-2022. Her research interests include uncertainty management in spatiotemporal databases, online query processing in spatial databases, geographic information systems (GIS) and health informatics. She has over 40 peer-reviewed articles published in international journals and conference proceedings.She is also a recipient of several local and international grants.

MATTHEW HORAK received his B.S. in Mathematics from Northern Arizona University in 1996 and his Ph.D. in Mathematics from Cornell University in 2003. From 2003 to 2006 he was a postdoctoral researcher at Trinity College in Hartford, Connecticut. From 2006 to 2014 he served as assistant and associate professor at the University of Wisconsin - Stout where he was PI in the first NSF-funded Research Experiences for Undergraduates site at the university. From 2015 to 2017, he held visiting professor positions at the United Arab Emirates University in the United Arab Emirates, and Hanyang University in Seoul, South Korea. In 2018, he joined Lockheed Martian as a Machine Learning and Artificial Intelligence Research Engineer. He is currently a Research Scientist in the Serverless Compute group of Amazon Web Services. His research publications span diverse fields including algebraic topology, computational geometry and machine learning.

SIDDHESHWARI BANKAR received her Bachelor of Engineering degree in Computer Engineering from Savitribai Phule Pune University, Pune, Maharashtra, India, in 2022, and her M.S. degree in Computer Science with a specialization in Data Science from Seattle University, Seattle, Washington, USA, in 2024. She is currently a Volunteer Researcher at the College of Science and Engineering, Seattle University. Her research interests include data science, machine learning, and generative AI.

SARTAJ BHUVAJI received his Bachelor of Engineering degree in Computer Engineering from Savitribai Phule Pune University, Pune, Maharashtra, India, in 2020. He earned his M.S. in Computer Science with a specialization in Data Science from Seattle University, Seattle, Washington, USA, in 2024. He currently serves as a mentor for Seattle University's Machine Learning Club. His research interests include deep learning, natural language processing, and generative artificial intelligence.

SUNGROUL KIM is currently Professor at Soonchunhyang University, Department of Environmental Health Sciences, South Korea. He received his Ph.D. degree in Environmental Health Science and Engineering from Johns Hopkins Bloomberg School of Public Health, Baltimore, South Korea, in 2005. He has been served as a dean of industry-academy cooperation foundation, Soonchunhyang University (2021 2022) and a director of BK21 four program of department of ICT environmental health system, graduate school of the University (2020 2023). He has been selected as a global top 2% scientist by citation which reported through Elsevier and Stanford University. His primary research focuses are the exposure assessment using real time sensors and biomarkers and its impact on human health, especially, in the areas of respiratory and mental health.

CHOON-SIK PARK is currently an honorary professor at Soonchunhyang University, Bucheon Hospital, South Korea. He received his B.S. degree in Medicine from Seoul National University, South Korea, in 1978, followed by his M.D. and Ph.D. degrees from the same institution in 1984 and 1988, respectively. From 2001 to 2018, he served as a professor at Soonchunhyang University Bucheon Hospital, where he also held the position of Director of the Institute of Clinical Medicine. Since 2018, he has held the position of honorary professor. Additionally, Dr. Park has been the President of the Korea Biomedical Resources Cooperation Bank since 2008. He has published over 320 research papers in international journals and conferences. His research primarily focuses on the pathogenesis of asthma and idiopathic pulmonary fibrosis among chronic respiratory, as well as omics research aimed at developing diagnostic markers and therapeutics.

• • •