Enhancing Sensor Systems Through Sensor Drift Identification and Compensation Using Jensen-Shannon Divergence and CTGAN

by

Shima MAHINNEZHAD

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER’S DEGREE
WITH THESIS IN ELECTRICAL ENGINEERING
M.A.Sc.

MONTREAL, APRIL 18, 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Prof. Andy Shih, Thesis supervisor
Department of Electrical Engineering, École de technologie supérieure

Prof. Kuljeet Kaur, Thesis Co-Supervisor
Department of Electrical Engineering, École de technologie supérieure

Prof. Georges Khaddoum, Chair, Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Prof. Qingsong Wang, External Examiner
Department of Electrical Engineering, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON FEBRUARY 28, 2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This thesis, submitted in fulfillment of the requirements for a Master's degree in Electrical Engineering at École de technologie supérieure, represents the original work of its author, Shima Mahinnezhad. The research, carried out from September 2022 to September 2024, addresses key challenges in sensor systems, particularly sensor drift. It explores methods for detecting and mitigating this issue through ML techniques and innovative data generation approaches, such as using generative models like Conditional Generative Adversarial Networks. This thesis introduces a novel approach that, for the first time, quantifies and addresses data drift by enhancing the quality of the data. This method can be applied to other sensor-related challenges where data quality is a primary concern.

The results presented in this thesis regarding this achievement were shared in a conference paper at IEEE I2MTC 2024 and have also been submitted to the journal IEEE Transactions on Instrumentation Measurement. In response to feedback received during the first review, we are addressing the reviewers' comments and revising the submission to ensure it is ready for the second round of evaluation.

ACKNOWLEDGEMENTS

I am grateful for the generous support and contributions of many individuals throughout the completion of this thesis. First and foremost, I extend my sincere gratitude to my advisor, Professor Andy Shih, for his invaluable guidance, mentorship, patience, and encouragement. His professional expertise has been instrumental during the challenging and lengthy duration of this project. I am hopeful that the culmination of this thesis reflects the fruition of his mentorship and patience.

Additionally, I would like to acknowledge Professor Kuljeet Kaur, my co-supervisor, for her insightful advice during the project, as well as express my gratitude to my committee members for their guidance, feedback, and unwavering support throughout the research process.

Lastly, I extend heartfelt thanks to my family for their unwavering support and patience throughout this endeavor.

Amélioration des systèmes de capteurs grâce à l'identification et la compensation de la dérive des capteurs en utilisant la divergence de Jensen-Shannon et CTGAN

Shima MAHINNEZHAD

RÉSUMÉ

De nombreux systèmes modernes de détection de gaz utilisent l'apprentissage automatique (ML) pour maintenir une précision constante, même face à des défis tels que la dérive des capteurs et le déséquilibre des classes. La dérive des capteurs fait référence à la diminution progressive des performances d'un capteur au fil du temps, ce qui complique l'identification correcte des types de gaz par les modèles de ML. De plus, de nombreux ensembles de données de détection de gaz sont déséquilibrés, ce qui signifie que certains types de gaz sont sous-représentés, ce qui affecte les performances des modèles.

Pour relever ces défis, deux techniques importantes sont utilisées dans ce travail, à savoir la détection de dérive des capteurs et l'augmentation de données. Cependant, les méthodes actuelles ont souvent du mal à gérer la complexité des dérives progressives ou non linéaires ainsi que des ensembles de données déséquilibrés. En outre, les techniques traditionnelles d'augmentation de données ne sont souvent pas adaptées aux données tabulaires, qui sont couramment utilisées dans les systèmes à base de capteurs.

Cette thèse propose une approche novatrice qui utilise la divergence de Jensen-Shannon (JS) pour détecter et mesurer la dérive des capteurs, et des réseaux adverses génératifs tabulaires conditionnels (CTGAN) pour générer des données synthétiques améliorant l'équilibre des ensembles de données et compensant le déséquilibre des classes. La divergence JS permet d'identifier précisément la dérive des capteurs en comparant les distributions de probabilité des données des capteurs au fil du temps, ce qui aide à mieux comprendre comment la dérive affecte la précision de la classification. CTGAN est utilisé pour créer des données tabulaires synthétiques de haute qualité, assurant une meilleure représentation des classes de gaz minoritaires.

Notre recherche se concentre sur un ensemble de données de capteurs de gaz collectées sur 36 mois, incluant plusieurs lots de relevés de capteurs. Nous utilisons la divergence JS pour détecter et mesurer la dérive, et CTGAN pour générer des données synthétiques qui répondent à la fois à la dérive des capteurs et au déséquilibre des classes. Cette combinaison de divergence JS et CTGAN est appliquée pour la première fois afin d'améliorer la précision des systèmes de détection de gaz.

En évaluant des modèles de ML tels que SVM et MLP, nous démontrons des améliorations significatives de la précision de classification, avec des gains allant jusqu'à 20% dans certains lots. Nos résultats mettent en évidence l'efficacité de cette approche pour traiter à la fois la dérive des capteurs et le déséquilibre des données, contribuant à une meilleure compréhension des moyens d'améliorer les systèmes de détection de gaz.

En résumé, cette thèse apporte des contributions importantes aux domaines de la détection de dérive des capteurs et de l'augmentation de données dans les systèmes de détection de gaz. Elle présente une nouvelle méthodologie pour quantifier la dérive à l'aide de la divergence JS et aborde le déséquilibre des classes avec CTGAN, améliorant ainsi la précision et la fiabilité des modèles de ML dans les applications basées sur des capteurs.

Mots-clés: CTGAN, Divergence de Jensen-Shannon, Équilibrage des données, Augmentation de données, Modèles génératifs, Systèmes de capteurs

Enhancing Sensor Systems Through Sensor Drift Identification and Compensation Using Jensen-Shannon Divergence and CTGAN

Shima MAHINNEZHAD

ABSTRACT

Many modern gas detection systems use machine learning (ML) to achieve consistent accuracy, even in the face of challenges like sensor drift and class imbalance. Sensor drift refers to the gradual decline in a sensor's performance over time, which makes it difficult for ML models to correctly identify gas types. Additionally, many gas detection datasets are imbalanced, meaning that some gas types are underrepresented, which affects the performance of the models.

To address these challenges, two important techniques are used, in this work, namely, sensor drift detection and data augmentation. However, current methods often struggle to handle the complexity of gradual or non-linear drift along with the imbalanced datasets. Moreover, traditional data augmentation techniques are often not suitable for tabular data, which is commonly used in sensor-based systems.

This thesis introduces a novel approach that uses Jensen-Shannon (JS) divergence to detect and measure sensor drift, and Conditional Tabular Generative Adversarial Networks (CTGAN) to generate synthetic data that improves dataset balance and compensates for class imbalance. JS divergence allows us to precisely identify sensor drift by comparing the probability distributions of sensor data over time, helping us better understand how drift affects classification accuracy. CTGAN is used to create high-quality synthetic tabular data, ensuring better representation of minority gas classes.

Our research focuses on a gas sensor dataset collected over 36 months, which includes multiple batches of sensor readings. We use JS divergence to detect and measure drift, and CTGAN to generate synthetic data that addresses both sensor drift and class imbalance. This combination of JS divergence and CTGAN is applied for the first time to improve the accuracy of gas detection systems.

By evaluating ML models such as SVM and MLP, we demonstrate significant improvements in classification accuracy, with gains of up to 20% in some batches. Our findings highlight the effectiveness of this approach in addressing both sensor drift and data imbalance, contributing to a better understanding of how to improve gas detection systems.

In summary, this thesis makes important contributions to the fields of sensor drift detection and data augmentation in gas detection systems. It presents a new methodology for quantifying drift using JS divergence and addresses class imbalance with CTGAN, enhancing the accuracy and reliability of ML models in sensor-based applications.

Keywords: CTGAN, Jensen-Shanon Divergence, Data balancing, Data augmentation, Generative Models, Sensor Systems

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machine
MLP	Multi-Layer Perceptron
GB	Gradient Boosting
LR	Logistic Regression
GAN	Generative Adversarial Network
CGAN	Conditional Generative Adversarial Network
CTGAN	Conditional Tabular Generative Adversarial Network
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
JS	Jensen-Shannon (Divergence)
KL	Kullback-Leibler (Divergence)
EWMA	Exponentially Weighted Moving Average
CUSUM	Cumulative Sum Control Chart
GLRT	Generalized Likelihood Ratio Test
SSIT	Solid-State Ion Transport
MOX	Metal-Oxide (Sensor)
VOC	Volatile Organic Compound
GLR	Generalized Likelihood Ratio

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

$\mathbf{x[k]}$	Sensor reading at time step k
$\mathbf{y[k]}$	Exponential Moving Average (EMA) at time step k
α	Smoothing parameter used in EMA
JS	Jensen-Shannon Divergence
KL	Kullback-Leibler Divergence
$\mathbf{P(x)}$	Probability distribution of sensor data at time t_1
$\mathbf{Q(x)}$	Probability distribution of sensor data at time t_2
$\mathbf{M(x)}$	Midpoint distribution between $P(x)$ and $Q(x)$
$\mathbf{L(y, f(x))}$	Loss function for gradient boosting
$\mathbf{f_m(x)}$	Current model in gradient boosting
η	Learning rate for gradient boosting
$\mathbf{G(z\|y)}$	Generator of synthetic data in CTGAN
$\mathbf{D(x\|y)}$	Discriminator in CTGAN
$\beta_\mathbf{0}$	Intercept in Logistic Regression
$\beta_\mathbf{1}, \beta_\mathbf{2}, ..., \beta_\mathbf{n}$	Coefficients of independent variables in Logistic Regression
$\log\left(\frac{\mathbf{P(y=1)}}{\mathbf{1-P(y=1)}}\right)$	Log-odds of the event happening in Logistic Regression
$\exp(\beta_\mathbf{i})$	Odds ratio for a predictor x_i in Logistic Regression

INTRODUCTION

Sensors play a pivotal role in modern technology, providing real-time data for diverse applications such as environmental monitoring, healthcare, and industrial automation (Kalsoom, Ramzan, Ahmed & Ur-Rehman, 2020a). Their ability to detect physical, chemical, and biological changes enhances efficiency and mitigates risks (Fine, Cavanagh, Afonja & Binions, 2010a). However, sensor drift—caused by factors like temperature fluctuations, contamination, and aging—degrades sensor accuracy over time, jeopardizing reliability and stability (Figure 0.1). In gas detection systems, sensor drift poses significant safety and operational risks, as inaccuracies can result in undetected leaks or misidentification of hazardous gases (Wu, Liu, Luo & Qiu, 2020). Therefore, robust drift detection and compensation methods are essential to ensure sustained performance.

Machine learning (ML) models are increasingly employed for sensor drift detection and compensation, enabling systems to adapt to changing environmental conditions and maintain reliable performance. These models utilize historical and real-time data to identify patterns and deviations that indicate drift. While ML algorithms leverage historical sensor data for real-time compensation (Yi *et al.*, 2023), class imbalance—arising from the infrequent occurrence of certain gases—biases predictions and hampers the detection of rare but critical events (Johnson & Khoshgoftaar, 2019).

Sensor drift and class imbalance are interrelated challenges that significantly affect the accuracy and reliability of classification models in sensor-based systems. Drift alters the underlying data distribution, leading to model degradation and increasing the likelihood of misclassifications, particularly for minority classes. Simultaneously, class imbalance amplifies this issue, as minority class instances—already underrepresented—become even harder to detect in drift-affected datasets. Together, these challenges exacerbate false negatives and reduce the robustness of gas detection systems. Addressing both drift and class imbalance concurrently is crucial, as

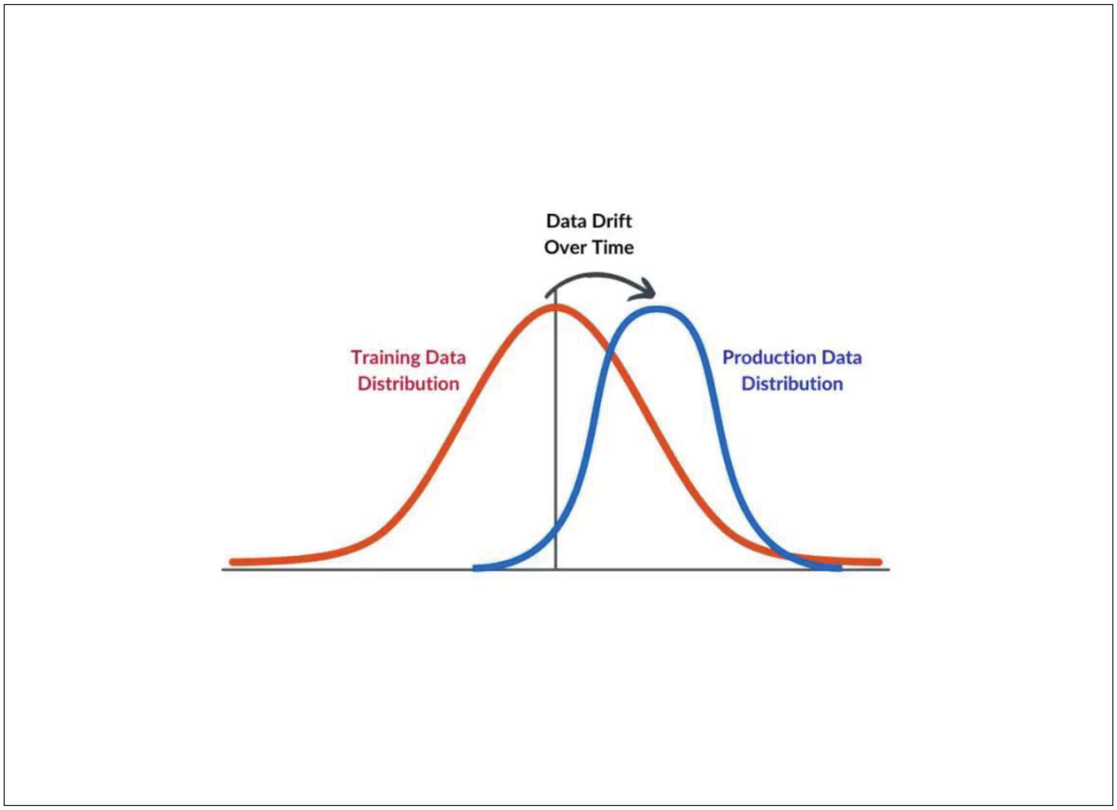
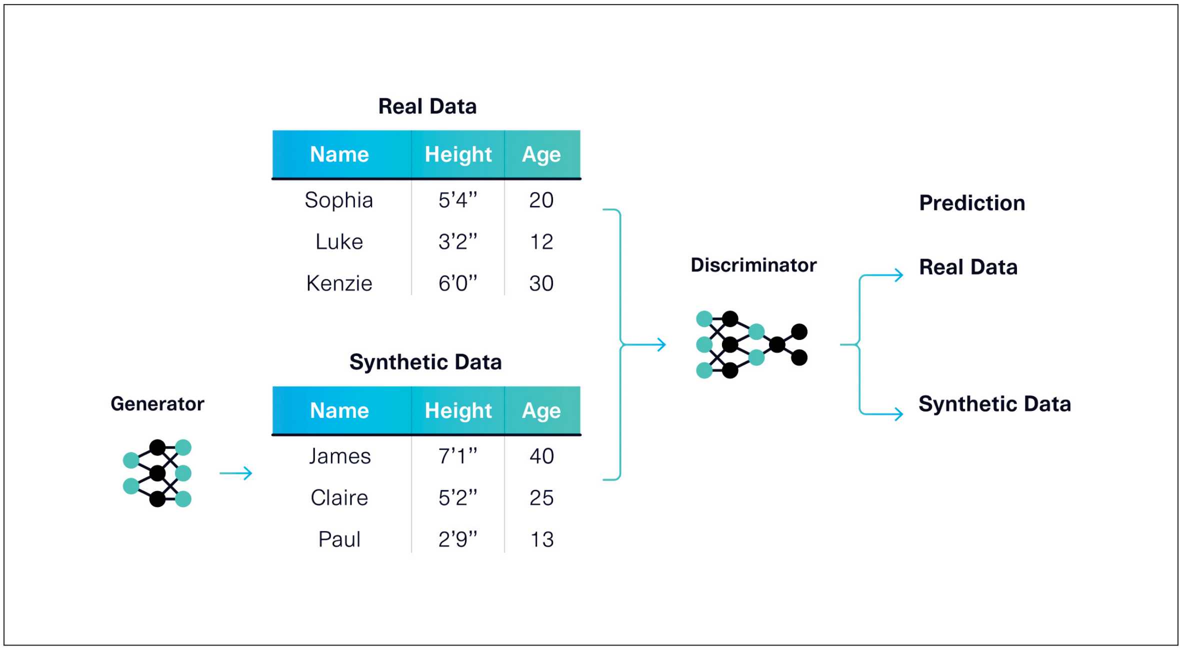


Figure 0.1 Data drift concept, adapted from (Intelligence, 2024). The shift in distribution from training data (red) to production data (blue) illustrates model degradation due to changing data environments

doing so ensures that ML models can adapt to dynamic environments while maintaining accurate detection of critical events. This thesis explores the integration of advanced drift detection and compensation techniques with synthetic data generation to provide a unified solution to these interconnected problems.

Traditional approaches to address class imbalance, such as oversampling or undersampling, often lead to overfitting or loss of valuable data (Good, 2006; Mohammed, Rawashdeh & Abdullah, 2020; Liu, Wu & Zhou, 2008). However, advanced methods, such as Conditional Tabular Generative Adversarial Networks (CTGANs), mitigate these limitations by generating high-quality synthetic data that preserves the original dataset's distribution (Xu, Skoularidou, Cuesta-Infante & Veeramachaneni, 2019a). Figure 0.2 illustrates how CTGAN balances class representation while reducing overfitting risks.



Real Data

Name	Height	Age
Sophia	5’4”	20
Luke	3’2”	12
Kenzie	6’0”	30

Synthetic Data

Name	Height	Age
James	7’1”	40
Claire	5’2”	25
Paul	2’9”	13

Figure 0.2 Overview of a GAN model, adapted from (AIMind, 2024). The generator creates synthetic data, while the discriminator evaluates its authenticity, enabling iterative improvement of the dataset

This thesis introduces a novel framework for gas detection systems by integrating CTGAN-based data augmentation with Jensen-Shannon (JS) divergence for sensor drift detection. JS divergence, a robust metric for comparing probability distributions, identifies changes in sensor behavior by monitoring distribution shifts and CTGAN-generated synthetic data addresses class imbalance, improves ML classification accuracy, and enhances robustness to drift-affected data (Liu, Wang, Sui & Zhu, 2024).

Experimental results demonstrate significant accuracy improvements—up to 20% for minority classes—highlighting the effectiveness of this approach. CTGAN also supports drift compensation by generating synthetic data that reflects varying sensor conditions, ensuring timely adjustments and extended sensor lifespan. This unified framework addresses two critical challenges in gas detection: class imbalance and sensor drift. Furthermore, the methods explored have broader applications in domains such as environmental monitoring, healthcare, and industrial safety systems.

0.0.1 Objectives of the Study

The primary goals of this research are:

- Develop algorithms for sensor drift detection using JS divergence.
- Assess the impact of drift on system reliability and accuracy by training various ML models on separate batches of data. Compare the results obtained from drift-free batches with those from batches affected by drift.
- Implement CTGAN for synthetic data generation to address class imbalance.
- Evaluate the quality of CTGAN-generated data by assessing its alignment with real-world characteristics. This is achieved by comparing the performance of ML models trained on the original dataset with those trained on modified datasets augmented with high-quality synthetic data.
- Enhance ML model performance through data augmentation, focusing on robustness to drift-affected data.
- Validate the proposed methods using real-world gas detection datasets.

0.0.2 Contributions of the Study

Key contributions of this research include:

1. Utilized JS divergence to effectively detect sensor drift under uncertain conditions.
2. Applied and optimized CTGAN to generate high-quality synthetic data, addressing class imbalance in sensor data.
3. Established robust metrics to assess the quality and real-world alignment of generated synthetic data.

4. Demonstrated improved accuracy and robustness of ML models through a comparative analysis before and after data augmentation.
5. Proposed a comprehensive framework combining data augmentation and drift detection to improve the reliability of gas detection systems.
6. Conducted a performance evaluation of traditional drift detection methods and highlighted the advantages of the proposed approach.

This research has been presented at the IEEE Instrumentation and Measurement Conference (I2MTC) 2024 (Mahinnezhad, Mahinnezhad, Kaur & Shih, 2024), and an extended version, titled *Sensor Drift Identification and Compensation using JS Divergence and CTGAN*, has been submitted to the IEEE Transactions on Instrumentation and Measurement.

0.0.3 Organization of the Thesis

Chapter 1 reviews traditional drift detection methods, challenges of class imbalance, and CTGAN applications. Chapter 2 details the proposed methodology, datasets, and evaluation metrics. Experimental results, including drift detection and classification improvements, are presented in Chapter 3. Chapter 4 concludes with a summary of findings, implications, and future research recommendations.

CHAPTER 1

LITERATURE REVIEW

Gas detection systems are critical for ensuring safety across industries such as manufacturing, environmental monitoring, healthcare, and public safety. These systems rely on accurate and timely sensor data to detect potentially hazardous substances. However, two significant challenges can compromise their effectiveness: class imbalance in the dataset and sensor drift over time.

Class imbalance refers to the disproportionate representation of certain gas types or concentrations in the training dataset (Narkhede, Walambe, Chandel, Mandaokar & Kotecha, 2022b). For instance, in real-world applications, gases like methane or carbon monoxide may dominate the data, while rare but hazardous gases such as ammonia or sulfur dioxide may be underrepresented (Rajakumar & Choi, 2023b). This imbalance can result in biased models that perform well for the majority classes but fail to accurately detect minority classes, which could have critical safety implications (He & Garcia, 2009). In safety-critical systems, such as gas detection, the misclassification or non-detection of rare hazardous gases could lead to catastrophic consequences (Cheung, Lin & Lin, 2018). This imbalance presents a unique challenge for machine learning models, as they must generalize effectively across all classes despite the skewed data distribution (Sun, Wong & Kamel, 2009).

Sensor drift further compounds these challenges. Over time, sensors lose accuracy due to factors like temperature fluctuations, humidity, and aging, leading to false alarms or missed detections (Kalsoom, Ramzan, Ahmed & Ur-Rehman, 2020b; Fine, Cavanagh, Afonja & Binions, 2010b; Righettoni, Amann & Pratsinis, 2015). Traditional techniques have been developed to detect and mitigate the effects of drift, focusing on identifying deviations in sensor performance and minimizing their impact on measurements. However, managing both sensor drift and class imbalance simultaneously is a complex task. Innovative approaches are required to develop robust and high-performing classification systems that maintain reliability and accuracy in gas detection applications.

1.1 Traditional Drift Detection Techniques and JS Divergence

1.1.1 Traditional Drift Detection Approaches

Drift detection in sensors has been extensively studied, with many algorithms focusing on identifying and mitigating drift effects. General fault detection algorithms, often based on deterministic models like Luenberger observers (Clark, Fosth & Walton, 1975) are widely discussed. These methods effectively detect sensor faults and separate drift from other effects, such as control inputs and external disturbances. However, deterministic methods like Luenberger observers may struggle in noisy environments(Mehra & Peschon, 1971b; Kailath, 1970).

Similarly, Kalman filters (Mehra & Peschon, 1971a) have been widely adopted in systems where randomness and uncertainty are present. They work by comparing the predicted sensor values with actual readings and are particularly useful in detecting subtle changes, such as sensor drift. However, Kalman filters are sensitive to biases introduced by model errors, leading to misdetection of drift when biases are present even without an actual fault. Additionally, Kalman filters may struggle with noisy sensors or situations where the model is not accurately tuned for the sensor's behavior, causing false positives or missed detections. The filter's slow adaptation to gradual drifts can also be problematic, as it may fail to detect small, continuous drifts until they have significantly affected the system.

To enhance Kalman filter performance, various bias decoupling techniques have been proposed. Friedland (Friedland, 1969) introduced a method to estimate bias without augmented Kalman filters, avoiding numerical inaccuracies. Subsequent work by Friedland and Grabousky (Friedland & Grabousky, 1982) incorporated random noise effects into the bias model. Improved techniques, such as those addressing dynamic and bias state correlations (Ignagni, 2002), and optimal two-stage estimators for random biases (Alouani, Xia, Rice & Blair, 1993; Keller & Darouach, 1997), provide robust frameworks for model adjustment before applying Kalman filters for drift detection. Despite these advances, these techniques assume constant or well-characterized bias, which may not hold in dynamic systems, reducing their adaptability (Friedland, 1969; Keller & Darouach, 1997).

In application-specific domains, such as nuclear power plants, Kalman filter-based techniques have been developed for detecting sensor and actuator faults. Functional redundancy concepts have been applied for early fault detection in pressurizer instrumentation (Clark & Campbell, 1982), while real-time detection systems for the same context were further explored (Tylee, 1983a). Off-line detection methods aimed at condition-directed sensor maintenance have also been investigated (Tylee, 1983b). A broader application of fault detection algorithms in nuclear reactor power systems highlights their versatility and importance in safety-critical environments. Nonetheless, these methods often rely on highly specialized system models, which are resource-intensive to develop and may not generalize across different contexts (Roy, Banavar & Thangasamy, 1998).

Residual generation is a key step in fault detection. A Kalman filter configured with a mathematical model generates residuals in the form of innovations, enabling statistical drift detection (Cho & Jiang, 2012). Advanced approaches, such as using Hamiltonian methods to evaluate steady-state covariance (Vaughan, 1970) and deriving drift rates analytically, further enhance detection accuracy. However, residual generation approaches are highly sensitive to model accuracy; any mismatch between the model and the system can result in false positives or degraded performance (Cho & Jiang, 2012; Vaughan, 1970).

Statistical change detection methods, including EWMA (Roberts, 2000), CUSUM (Page, 1954), GLRT (Willsky & Jones, 1976), and novel approaches like SSIT (Robinson & Ho, 1978), improve detection capabilities by leveraging residual statistical properties. SSIT, for instance, minimizes detection delays, albeit with higher computational complexity. A key limitation of these methods is their dependence on predefined thresholds, which can be challenging to set optimally and may lead to delays or false detections if system characteristics change over time (Robinson & Ho, 1978).

One of the other techniques, Luenberger observers (Roch, McCarter, Matheson, Clark & Olafson, 1982), offers a deterministic solution by predicting sensor behavior using a system model. The observer compares these predictions to actual sensor readings to identify discrepancies indicative of faults or drifts. However, this method is highly dependent on the accuracy of the system model. If the model does not perfectly represent the actual system, the Luenberger observer may fail to detect drift or may produce false alarms. Moreover, the approach is not well-suited for stochastic systems where sensor noise or environmental variability can obscure drift, limiting its effectiveness in dynamic or uncertain environments.

The Generalized Likelihood Ratio (GLR) method, proposed by Willsky and Jones (1976), is another approach designed for detecting changes in stochastic linear systems. While it is effective at estimating the time and magnitude of a fault, it faces significant limitations when applied to gas sensors. These sensors often exhibit non-linear drift behaviors, which the GLR method is not well-equipped to handle. Additionally, the method is computationally intensive, making it less feasible for real-time monitoring in resource-constrained systems. The high computational demand can significantly limit its practicality in large-scale or remote sensor networks.

Another method, functional redundancy (Roch *et al.*, 1982), relies on using multiple redundant sensors to cross-check measurements and detect drifts or incipient failures. While this approach improves fault detection reliability, it requires additional sensors, which increases the complexity and cost of the system. Furthermore, redundant sensors may suffer from similar drift issues, reducing the effectiveness of this method in the long term. The need for multiple sensors also introduces the challenge of data fusion and synchronization, adding another layer of complexity to the system.

In conclusion, while various techniques have been developed to detect sensor drift, each comes with its own set of challenges. Whether it is the reliance on accurate system models in Luenberger observers, the susceptibility of Kalman filters to biases and noise, the computational cost of GLR, or the complexities of functional redundancy, these methods still face significant hurdles in providing reliable and real-time drift detection, particularly in harsh industrial environments like nuclear power plants.

1.1.2 JS Divergence for Drift Detection

JS divergence offers a powerful approach for detecting shifts in sensor data distributions, allowing for real-time monitoring and proactive drift correction (Wei, He, Wang, Wang & Zhou, 2021). JS divergence is a powerful tool for detecting sensor drift by comparing current sensor data distributions with historical baselines. Drift is identified when the divergence exceeds a predefined threshold, triggering corrective actions such as sensor recalibration or model updates (Eom & Byeon, 2023). Here are the key advantages of using JS divergence for detecting and addressing sensor drift:

- **Real-time Monitoring**: JS divergence can be computed efficiently, making it well-suited for continuous, real-time drift monitoring in dynamic environments.
- **Early Detection**: The method is sensitive to small distributional changes, allowing early intervention before drift significantly impacts system accuracy (Gu *et al.*, 2021).
- **Versatility**: JS divergence is adaptable to a wide range of sensor types and data formats, including both categorical and continuous data, making it applicable to diverse use cases in gas detection systems (Menéndez, Pardo, Pardo & Pardo, 1997).

By integrating JS divergence into drift detection frameworks, gas detection systems can maintain high reliability and consistent performance, even under evolving environmental conditions and sensor aging. This proactive approach ensures that potential issues are addressed promptly, reducing downtime and improving safety (Gu *et al.*, 2021).

1.2 ML Solutions for Sensor Drift and Class Imbalance

1.2.1 Concept Drift in ML for Gas Sensors

In ML, concept drift refers to the change in the statistical properties of the target variable over time. This results in a model that no longer accurately reflects the real-world relationships it was designed to predict (Lin, Chang, Nie & Dong, 2024). In gas detection systems, concept drift occurs when the relationship between the sensor readings and the gas concentrations shifts due to sensor degradation, environmental changes, or the introduction of new gases into the environment (Žliobaitė, 2010).

Concept drift poses an additional challenge for ML models trained on historical data, as the data distribution may no longer be representative of the current environment (Mehmood *et al.*, 2021). If left unaddressed, concept drift can lead to significant declines in model performance. This can cause the system to generate inaccurate predictions or fail to detect critical events such as gas leaks (Chitsazian, Kashi & Nikanjam, 2023).

To mitigate concept drift, continuous model updates and retraining are necessary. Adaptive ML techniques, which allow models to evolve and learn from new data, are particularly useful in handling concept drift (Casado *et al.*, 2022). For example, online learning algorithms update the model incrementally as new data is collected, ensuring that the system remains responsive to changes in the data distribution (Gama, Žliobaitė, Bifet, Pechenizkiy & Bouchachia, 2014). Additionally, techniques such as drift detection mechanisms (e.g., the Drift Detection Method) can be implemented to trigger model updates when significant drift is detected, maintaining the accuracy and reliability of the system over time (Gama *et al.*, 2014).

1.2.1.1 ML Solutions for Sensor Drift

Traditional methods for managing sensor drift, such as periodic recalibration, are time-consuming and lack real-time effectiveness. Therefore, advanced approaches are required to monitor and correct drift dynamically, ensuring gas detection systems remain accurate and dependable (Xu *et al.*, 2019a). To overcome these challenges, ML and advanced data analysis have become valuable tools. By using large datasets to train models, engineers can help gas detection systems adapt to drift and maintain their reliability.

Recent advances in ML offer promising solutions to mitigate sensor drift in gas detection systems. By training ML models on historical sensor data, patterns of sensor drift can be detected and compensated for without requiring manual intervention. This can be achieved through supervised learning techniques, such as Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP), which are trained to identify deviations in sensor readings and make real-time adjustments to ensure the continued accuracy of the system (Wang *et al.*, 2017).

In addition to supervised learning, unsupervised ML models can be employed to detect drift in the absence of labeled data. Clustering techniques, for instance, can identify anomalies in sensor behavior. This allows the system to recognize when drift is occurring even without explicit drift examples in the training set. Furthermore, ensemble learning approaches combine multiple models to enhance the system's robustness against drift, increasing both the reliability and the resilience of gas detection systems in dynamic environments (Patel *et al.*, 2020).

By leveraging ML capabilities, gas detection systems can not only detect drift but also predict when recalibration will be needed. This optimizes maintenance schedules and reduces downtime (Cramer, Shaw, Tulalian, Angelo & van Stuijvenberg, 2015). ML-based drift detection techniques represent a significant improvement over traditional methods, which are reactive and costly, allowing systems to maintain high levels of accuracy for longer periods (Xiang, Zi, Cong & Wang, 2023b).

1.2.2 ML Solutions for Class Imbalance

1.2.2.1 Overview of Class Imbalance in Datasets

Class imbalance is a common issue in many ML applications, including gas detection systems. In these systems, some gas types or concentrations may occur far more frequently than others, resulting in an uneven distribution of data. This imbalance can negatively affect ML models, as they tend to favor the majority class, leading to poor generalization for minority classes (Krawczyk, 2016). This section discusses the problem of class imbalance in gas detection datasets and explores traditional methods for addressing it, as well as their limitations.

In the context of gas detection systems, class imbalance refers to the disproportionate representation of certain gas types or concentrations in the training dataset (Narkhede *et al.*, 2022b). For instance, in real-world applications, gases like methane or carbon monoxide may be detected frequently, while more dangerous but rare gases like ammonia or sulfur dioxide may have fewer samples (Rajakumar & Choi, 2023b). This imbalance can lead to biased models that perform well for the majority class but fail to accurately identify or classify minority classes. This might be critical for safety (He & Garcia, 2009).

Class imbalance is particularly problematic in safety-critical systems because rare gases could be misclassified. For example, if a gas detection system consistently misclassifies or fails to detect hazardous gases present in small concentrations, the consequences could be catastrophic (Cheung *et al.*, 2018). The imbalanced nature of gas detection datasets thus poses a unique challenge for ML models, as they must be able to generalize effectively across all classes despite the skewed distribution of data (Sun *et al.*, 2009).

1.2.2.2 Previous ML-based Approaches to Class Imbalance

Several traditional methods have been developed to address class imbalance in ML. The most common approaches include oversampling the minority class, undersampling the majority class, and cost-sensitive learning.

1. **Oversampling**: Oversampling involves generating synthetic samples for the minority class to ensure that all classes are represented equally in the training data. This is often done using techniques like the Synthetic Minority Oversampling Technique (SMOTE). Figure 1.1 illustrates the process of SMOTE: the left panel depicts an imbalanced dataset with fewer positive points (blue diamonds), the middle panel shows the generation of synthetic points

(red diamonds) between existing positives, and the right panel presents a balanced dataset, which enables better model performance (Chawla, Bowyer, Hall & Kegelmeyer, 2002).

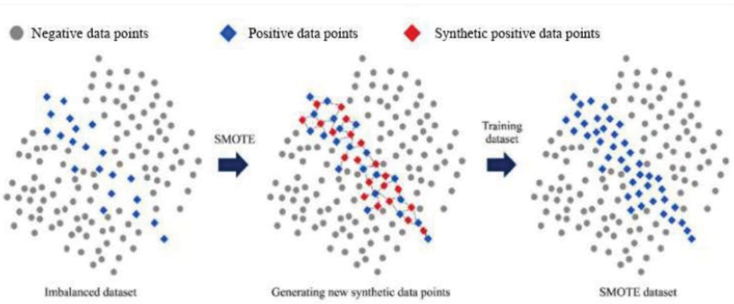


Figure 1.1 Process of Synthetic Minority Oversampling Technique (SMOTE) Algorithm, adapted from (Hairani *et al.*, 2023)

2. **Undersampling**: This method reduces the size of the majority class to match the minority class, thereby balancing the dataset. As shown in Fig 1.2, undersampling (left) selects fewer samples from the majority class (blue), while oversampling (right) duplicates the minority class (orange). Although effective, undersampling risks losing critical patterns from the majority class, which may lower model performance for those classes (Batista, Prati & Monard, 2004).
3. **Cost-sensitive learning**: By assigning higher penalties for misclassifying samples from the minority class, this method encourages the model to prioritize the minority class. Figure 1.3 illustrates the introduction of cost-sensitivity during feature selection and model induction. A cost matrix guides feature selection (FS) with instance weighting, followed by model induction (MI) to build a predictive model. Proper calibration is critical to avoid over-biasing toward the minority class (Zhou & Liu, 2005).

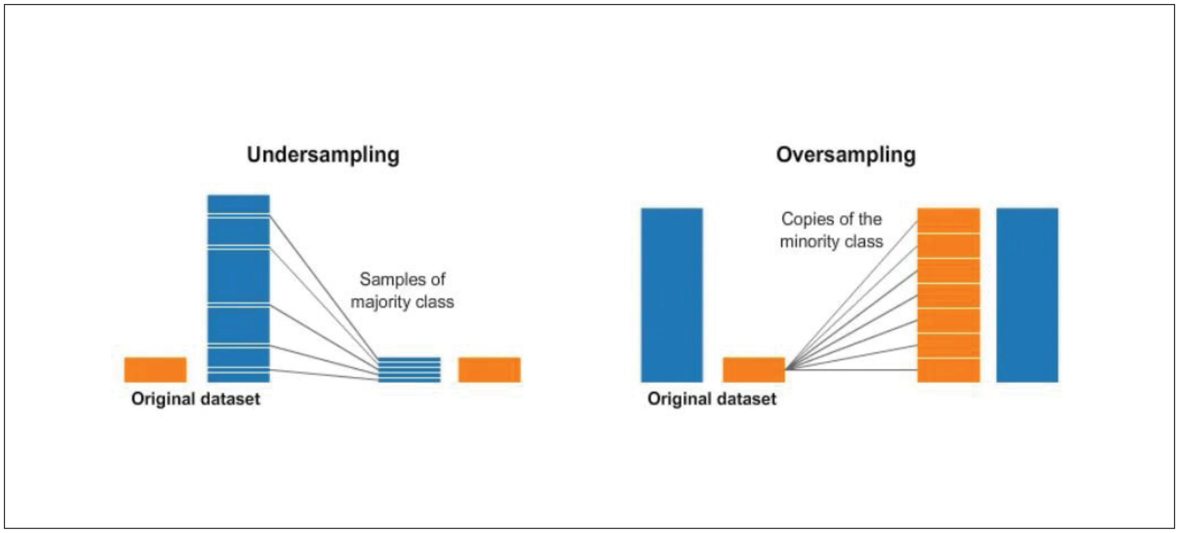


Figure 1.2 Resampling strategies for imbalanced datasets: undersampling (left) and oversampling (right)

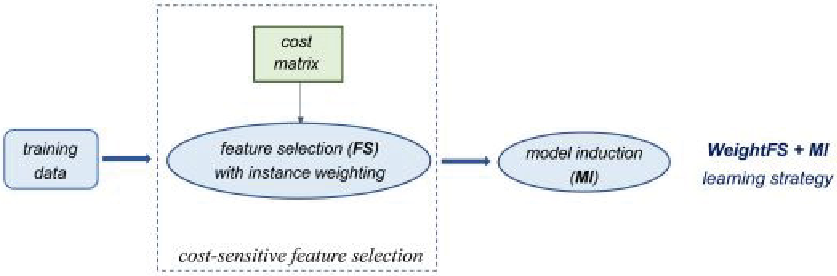

cost matrix
training data
feature selection (FS) with instance weighting
model induction (MI)
WeightFS + MI learning strategy
cost-sensitive feature selection

Figure 1.3 Introducing cost-sensitivity during feature selection, adapted from (Pes & Lai, 2021)

These methods address class imbalance effectively and improve model performance for underrepresented classes. However, each comes with its limitations that must be carefully considered based on the specific application.

1.2.2.3 Limitations of Traditional Approaches

Despite their benefits, traditional methods for handling class imbalance face several challenges:

1. **Oversampling Risks**: Oversampling increases the risk of overfitting by creating synthetic samples too similar to existing ones. This reduces the model's ability to generalize to new data (Japkowicz, 2000).
2. **Undersampling Drawbacks**: Undersampling can result in the loss of valuable data from the majority class, limiting the model's understanding of critical patterns. This is particularly problematic in contexts like gas detection, where majority class gases may represent typical environmental conditions (Zhou, Liu, Yuan & Jiang, 2024).
3. **Cost-sensitive Complexity**: Adjusting cost functions requires careful calibration to balance model focus. Improper settings may lead to biases that degrade overall performance. Additionally, cost-sensitive learning can be computationally intensive (Zhou & Liu, 2010).

These challenges highlight the need for advanced approaches such as CTGANs, which offer robust solutions by generating high-quality synthetic data (Miletic & Sariyar, 2024).

1.3 Generative Models to Solve Sensor Drift and Class Imbalance

Generative models like Generative Adversarial Networks (GANs) have emerged as powerful tools for creating synthetic data that closely reflects real datasets. These models are particularly effective in addressing challenges such as class imbalance and sensor drift in gas detection systems (Figueira & Vaz, 2022a).

1.3.1 Introduction to GANs

GANs are a class of ML models comprising two neural networks: a generator and a discriminator. The generator creates synthetic data, while the discriminator evaluates its authenticity by distinguishing between real and synthetic samples. These networks are trained adversarially, with the generator improving its outputs until the synthetic data becomes indistinguishable from real data (Figueira & Vaz, 2022b).

Initially developed for applications like image synthesis, GANs have demonstrated their versatility in generating tabular data for ML tasks. In gas detection systems, GANs can generate synthetic sensor data that simulates a wide range of operational conditions, thereby enhancing model training and improving performance (Alabdulwahab, Kim, Seo & Son, 2023). The structure of a Generative Adversarial Network (GAN) model, which forms the foundation of CTGANs, is illustrated in Fig 1.4.

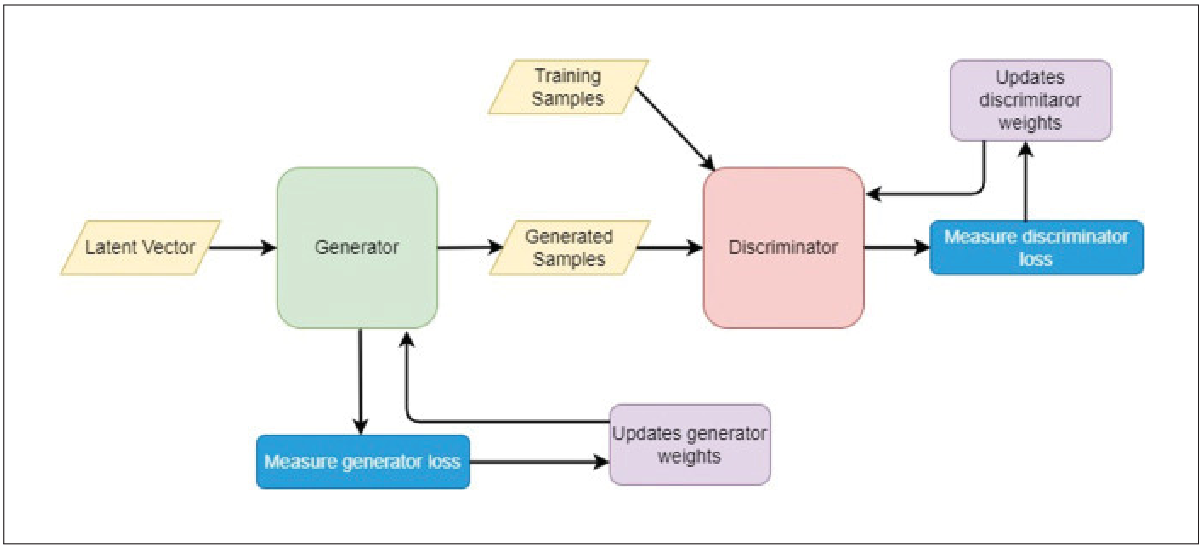


Figure 1.4 Generative Adversarial Network (GAN) model structure, adopted from (Habibi *et al.*, 2023b)

1.3.2 CTGANs

CTGANs extend the capabilities of traditional GANs to handle tabular datasets. CTGANs address challenges unique to tabular data, such as mixed data types (categorical and numerical features) and class imbalance. By conditioning the generator on specific features within the dataset, CTGANs create realistic synthetic samples that reflect the distribution of the original data, especially for underrepresented classes (Xu, Skoularidou, Cuesta-Infante & Veeramachaneni, 2019b).

Key advantages of CTGANs include:

- **Preserving Data Relationships**: CTGANs capture and maintain complex relationships between features, ensuring that the generated data reflects the statistical properties of the original dataset (Goyal & Mahmoud, 2024).
- **Avoiding Overfitting**: Unlike traditional oversampling techniques, CTGANs generate diverse and unique samples, reducing the risk of the model memorizing minority class patterns and improving generalization (Xu *et al.*, 2019b).
- **Handling Mixed Data Types**: CTGANs effectively process datasets containing both numerical and categorical variables, making them particularly suitable for real-world applications in domains like gas detection (Goyal & Mahmoud, 2024).

1.3.3 CTGANs for Addressing Class Imbalance

Class imbalance poses a major challenge in gas detection, as rare gases are often critical to identify but underrepresented in datasets. CTGANs effectively tackle this issue by generating synthetic samples for minority classes, creating a balanced dataset that enhances model performance. CTGAN-generated data improves the precision of ML models by addressing the scarcity of minority class samples. This reduces false negatives, enabling the detection of rare but hazardous gases more reliably. Studies have demonstrated the effectiveness of CTGANs in improving accuracy and robustness on imbalanced datasets, making them an essential tool for gas detection systems (Eom & Byeon, 2023).

1.4 Summary

This chapter reviewed key challenges in gas detection systems—sensor drift and class imbalance—and highlighted advancements in addressing these issues. Sensor drift, caused by factors such as aging and environmental changes, affects sensor accuracy over time, leading to potential false alarms or missed detections. Simultaneously, class imbalance in training datasets, where certain gas types dominate while rare but hazardous gases are underrepresented, can result in biased machine learning models that fail to generalize effectively across all classes.

These two challenges are interconnected, as sensor drift can exacerbate the effects of class imbalance by altering the distribution of detected gases over time, further complicating accurate classification. To evaluate these issues, JS divergence provides a robust metric for quantifying differences between distributions, enabling real-time detection of sensor drift and monitoring its impact on data imbalance.

To address these challenges, advanced generative models such as CTGAN offer a promising solution. CTGAN can generate high-quality synthetic data that maintains the original relationships within the dataset while avoiding overfitting, effectively mitigating class imbalance. By augmenting underrepresented classes, CTGAN enhances the ability of gas detection systems to identify rare and hazardous gases with greater accuracy. Together, JS divergence and CTGAN provide complementary tools for creating robust, high-performing systems capable of overcoming both drift and imbalance in gas detection applications.

CHAPTER 2

METHODOLOGY

This chapter provides a comprehensive exploration of methodologies for addressing sensor drift detection and class imbalance in gas detection systems. It introduces the Gas Sensor Array Dataset, details advanced techniques such as JS divergence for drift detection, and employs CTGAN for data augmentation to enhance ML model performance.

2.1 Gas Sensor Array Dataset

The Gas Sensor Array Dataset (Fonollosa, Rodríguez-Luján & Huerta, 2015) used in this study is publicly available and was introduced by J. Fonollosa *et al*. in the paper "Chemical Gas Sensor Array Dataset" (Fonollosa *et al.*, 2015). Collected over 36 months, the dataset comprises 13,910 measurements from an array of sixteen metal-oxide gas (MOX) sensors. It focuses on detecting six volatile organic compounds (VOCs), *i.e.*, Ethanol, Ethylene, Ammonia, Acetaldehyde, Acetone, and Toluene.

2.1.1 Key Dataset Characteristics

The dataset focuses on detecting six volatile organic compounds (VOCs), *i.e.*, Ethanol, Ethylene, Ammonia, Acetaldehyde, Acetone, and Toluene. The sensor array includes four widely used models from Figaro Engineering Inc.—TGS2600, TGS2602, TGS2610, and TGS2620—ensuring relevance to real-world applications like environmental monitoring, industrial safety, and healthcare. The data was generated under controlled conditions, with varying gas concentrations to simulate diverse environmental scenarios. This controlled setup ensures high-quality measurements, making the dataset suitable for ML applications.

2.1.2 Dataset Features and Challenges

This dataset is particularly valuable for its ability to address two critical challenges:

1. **Sensor Drift:** Sensor drift, caused by factors such as temperature changes, humidity, and aging, degrades sensor performance over time. The dataset's 36-month timespan enables analysis of both short-term and long-term drift, making it ideal for developing drift detection methods like JS divergence.
2. **Class Imbalance:** Significant variability exists in gas representation across batches. For instance, Batch 4 contains only 12 measurements for acetaldehyde, while Batch 10 includes 600 measurements per gas. This inherent imbalance poses challenges for traditional ML models, making it a suitable testbed for data augmentation techniques like CTGAN.

The distribution of data across the 10 batches is visualized in Fig. 2.1. These PCA plots demonstrate the variation in data patterns, reflecting the drift challenges addressed in this study.

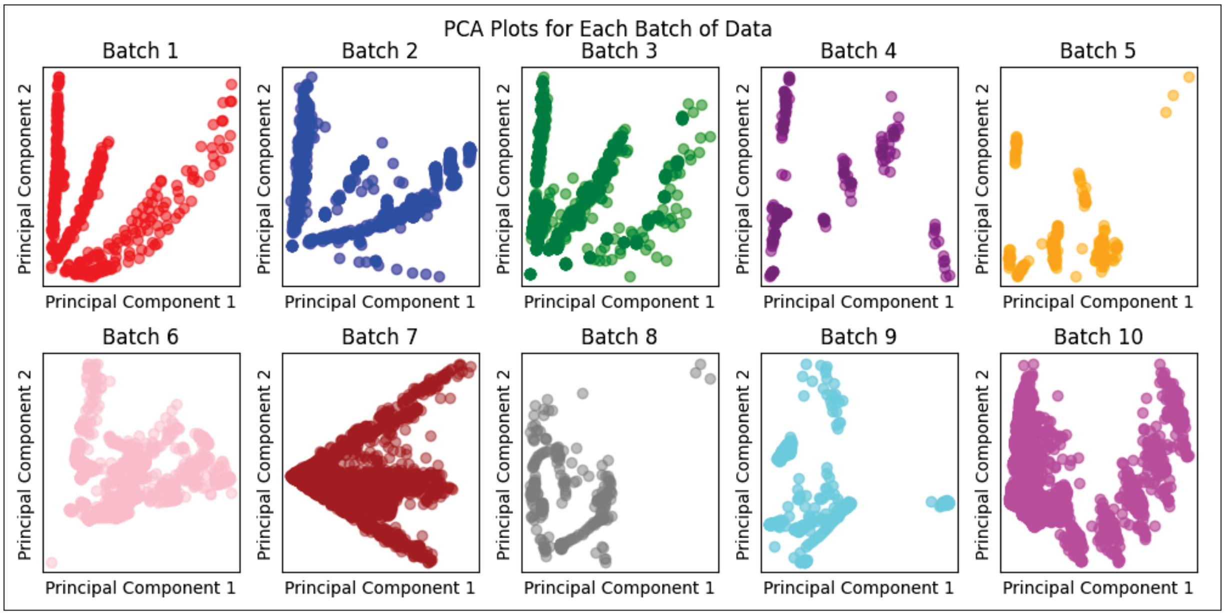


Figure 2.1 PCA plots of the Gas Sensor Array Dataset across 10 batches. Each plot illustrates the distribution of data within a batch, highlighting the variability in gas representation and the presence of drift over time

2.1.3 Dataset Organization

The dataset is organized into 10 batches, each representing distinct time periods. This temporal segmentation provides a structured framework for evaluating the effects of sensor drift. Furthermore, the inclusion of both steady-state and dynamic sensor behaviors allows for comprehensive feature extraction, capturing critical patterns essential for drift detection and gas classification.

By addressing real-world challenges such as sensor drift, class imbalance, and diverse gas concentrations, the Gas Sensor Array Dataset forms an excellent foundation for developing and testing advanced methodologies. Its detailed organization and controlled conditions ensure the reliability and applicability of findings derived from this study. Table 2.1 summarizes the distribution of gas instances across the batches.

Table 2.1 Distribution of gas instances into batches

Batch Number	**Month IDs**	**Ethanol**	**Ethylene**	**Ammonia**	**Acetaldehyde**	**Acetone**	**Toluene**	**Total**
Batch 1	1, 2	90	98	83	30	70	74	445
Batch 2	3, 4, 8, 9, 10	164	334	100	109	532	5	1244
Batch 3	11, 12, 13	365	490	216	240	275	0	1586
Batch 4	14, 15	64	43	12	30	12	0	161
Batch 5	16	28	40	20	46	63	0	197
Batch 6	17, 18, 19, 20	514	574	110	29	606	467	2300
Batch 7	21	649	662	360	744	630	568	3613
Batch 8	22, 23	30	30	40	143	33	18	294
Batch 9	24, 30	61	55	100	75	78	101	470
Batch 10	36	600	600	600	600	600	600	3600
Total	1–36	2565	2926	1641	1936	3009	1833	13910

2.1.4 Data Collection Process

The data collection process was meticulously designed to capture sensor responses under controlled environmental conditions, ensuring consistency and reliability. A computerized platform was used to automate the measurements, as described by Fonollosa *et al.*, and the setup included several key elements.

The sensor array consisted of 16 metal-oxide gas sensors housed in a 60 ml airtight chamber. A vapor delivery system, equipped with digital mass flow controllers and calibrated gas cylinders, introduced selected volatile organic compounds (VOCs) into the chamber at specific concentrations. The gas flow rate was maintained at a constant 200 ml/min throughout the process (Fonollosa *et al.*, 2015).

To ensure accurate readings, the sensors were exposed to clean air before and after each gas sample was introduced. This step allowed for recording baseline and recovery responses while ensuring the chamber remained uncontaminated for subsequent measurements. The sensors were operated at a controlled temperature, maintained by applying a steady 5V voltage to their built-in heaters (Fonollosa *et al.*, 2015).

Each measurement followed a structured sequence consisting of three phases:

- **Baseline Stabilization:** Clean air was circulated through the chamber for 50 seconds to stabilize the sensors.
- **Gas Exposure:** One of the six gases was randomly selected and circulated through the chamber for 100 seconds at a specific concentration.
- **Recovery:** Clean air was reintroduced for 200 seconds, allowing the sensors to return to their baseline state before the next gas exposure (Fonollosa *et al.*, 2015).

This systematic and controlled approach ensured the collection of high-quality data, enabling a precise and reliable analysis of sensor behavior under varying conditions.

Throughout the data collection process, the dynamic response of each sensor was recorded at a sampling rate of 100 Hz, resulting in time-series data. These time-series data were then processed to extract key features that capture both steady-state behavior (e.g., resistance change amplitude, normalized resistance) and transient behavior (e.g., response time, exponential moving average), providing a comprehensive basis for further analysis.

2.1.5 Feature Extraction

Feature extraction was performed to analyze sensor data effectively, focusing on capturing both the steady-state and dynamic behavior of the sensors. Each sensor in the array contributes eight distinct features, resulting in a comprehensive 128-element feature vector for each measurement, given the dataset's 16 sensors.

2.1.5.1 Feature Categories and Extraction Scheme

1. **Steady-State Features:**
   - These features capture the sensor's response during the stable phase of gas exposure. They include:
     - *Amplitude of Resistance Change:* Measures the change in sensor resistance between the baseline and the stable response phase, providing an indication of sensitivity to the gas.
     - *Normalized Resistance:* Standardizes the resistance change, allowing for comparisons across different sensors and gases.
   - These features are derived by analyzing the sensor's response curve during the exposure phase.
2. **Dynamic Features:**
   - These features characterize the transient behavior of the sensors during gas exposure. The Exponential Moving Average (EMA) was used to capture the rising and decaying phases of the sensor response.
   - EMA is calculated using the formula:

$$y[k] = (1 - \alpha)y[k - 1] + \alpha(x[k] - x[k - 1]), \tag{2.1}$$

     where $y[k]$ represents the EMA at time step k, $x[k]$ is the sensor reading, and α is the smoothing parameter.
   - To capture variations at multiple time scales, three values of α were used: 0.1, 0.01, and 0.001.

2.1.5.2 Extracted Features and Their Applications

With each sensor contributing eight features (both steady-state and dynamic), and a total of 16 sensors in the array, **128 features per measurement** were extracted. These features provide a comprehensive representation of both immediate sensor responses and long-term trends, which are essential for detecting patterns like sensor drift and enhancing gas classification accuracy.

The extracted feature set serves as the foundation for ML models by:

- Highlighting critical sensor behaviors for detecting drift.
- Enabling accurate classification of gases, even in the presence of drift or class imbalance.
- Supporting advanced data augmentation techniques.

2.2 JS Divergence for Drift Detection

Over time, sensor drift alters sensor responses, leading to misclassification in ML models trained on static datasets. Detecting and compensating for drift is therefore critical to maintaining the long-term accuracy and reliability of gas detection systems.

The JS divergence is a powerful information-theoretic measure for detecting changes in probability distributions. Derived from the Kullback-Leibler (KL) divergence, JS divergence is symmetric and yields finite values, making it particularly effective for comparing distributions even when zero-probability events are present (Lin, 1991).

2.2.1 Conceptual Overview of JS Divergence

JS divergence quantifies the similarity between two probability distributions using the concept of Shannon entropy, which measures the uncertainty in a distribution (Gu *et al.*, 2021; Menéndez *et al.*, 1997). For two distributions, P and Q, the JS divergence is defined as:

$$JS(P \parallel Q) = \frac{1}{2}KL(P \parallel M) + \frac{1}{2}KL(Q \parallel M), \tag{2.2}$$

where $M = \frac{1}{2}(P + Q)$ is the midpoint distribution, and KL is the Kullback-Leibler divergence, given by:

$$KL(P \parallel Q) = \sum P(x) \log \frac{P(x)}{Q(x)}. \tag{2.3}$$

Unlike KL divergence, which is asymmetric ($KL(P \parallel Q) \neq KL(Q \parallel P)$), JS divergence averages the KL divergence in both directions using the midpoint distribution, ensuring symmetry.

2.2.2 Properties of JS Divergence

The key properties of JS divergence that make it suitable for detecting sensor drift include:

- **Symmetry:** JS divergence is symmetric, meaning $JS(P \parallel Q) = JS(Q \parallel P)$, ensuring the result is independent of the order of distributions.
- **Finiteness:** The divergence always produces finite values, even if $P(x) = 0$ for some x while $Q(x) \neq 0$.
- **Boundedness:** The values of JS divergence range between 0 and 1, where 0 indicates identical distributions and 1 signifies maximum divergence.

These properties make JS divergence an ideal tool for monitoring distributional changes in sensor responses, enabling timely detection and correction of drift to maintain system performance.

2.2.3 Mathematical Interpretation of JS Divergence

Consider a sensor's response distribution at time t_1, denoted as P, and at time t_2, denoted as Q. To quantify sensor drift, the JS divergence between these two distributions is calculated using the formula:

$$JS(P \parallel Q) = \frac{1}{2} \sum_{i=1}^{n} P(x_i) \log \frac{P(x_i)}{M(x_i)} + \frac{1}{2} \sum_{i=1}^{n} Q(x_i) \log \frac{Q(x_i)}{M(x_i)}, \tag{2.4}$$

where $M(x_i) = \frac{1}{2}(P(x_i) + Q(x_i))$ represents the midpoint distribution between P and Q.

The JS divergence yields a low value when the distributions are similar, indicating little or no drift. Conversely, a high divergence value suggests significant drift in the sensor's behavior, signaling potential changes in the underlying conditions or sensor performance.

2.2.4 Applications in Sensor Drift Detection

The application of JS divergence to sensor data enables effective tracking of sensor performance over time. By comparing the distribution of current sensor responses to those from previous batches, we can detect and address drift. This approach provides several benefits:

1. *Identifying periods of significant drift:* High JS divergence values highlight instances where sensor responses have changed substantially, indicating the need for recalibration or model retraining.
2. *Monitoring long-term stability:* Regular computation of JS divergence reveals the rate at which drift occurs, aiding in predictive maintenance and ensuring system reliability.
3. *Enhancing model performance:* Early detection of drift allows ML models to be adjusted, maintaining accuracy and preventing performance degradation.

JS divergence is particularly effective for detecting drift as it captures subtle changes in probability distributions and provides a clear, quantitative metric. This makes it a valuable tool for real-time monitoring in gas sensor systems, where drift can compromise the accuracy and reliability of classification models.

2.3 CTGAN for Data Augmentation

In gas detection systems, data augmentation is crucial due to the inherent class imbalance in sensor datasets. Certain gases are overrepresented compared to others, making it challenging for traditional ML models to generalize effectively, especially for underrepresented gases. CTGAN is an advanced generative model specifically designed for tabular data, addressing challenges such as imbalanced categorical columns, multi-modal continuous data, and non-Gaussian distributions (Mirza, 2014; Xu *et al.*, 2019b). CTGAN was chosen for this study because of its ability to generate high-quality synthetic data that maintains the statistical properties of the original dataset while balancing minority classes.

2.3.1 Architecture of CTGAN

CTGAN builds on the traditional Generative Adversarial Network (GAN) framework. Introduced by Goodfellow *et al.* in 2014, GANs consist of two neural networks: the generator and the discriminator. The generator creates synthetic data that mimics the real data distribution, while the discriminator distinguishes between real and synthetic data. These networks are trained in an adversarial setup, where the generator learns to produce increasingly realistic data, and the discriminator improves its ability to differentiate between real and fake samples. Training continues until the generator produces data that the discriminator cannot reliably distinguish from real data (Mirza, 2014).

Conditional GANs (CGANs) enhance this framework by incorporating additional information, such as categorical or discrete columns, into the generator and discriminator. CTGAN extends this concept further to handle the complexities of imbalanced tabular data. By conditioning both the generator and discriminator on specific dataset features, CTGAN ensures that synthetic data accurately represents minority classes and the overall distribution of the dataset.

As depicted in Figures 2.2 and 2.3, the generator $G(z|y)$ takes a noise vector z, sampled from a standard multivariate normal distribution, and a conditional vector y, representing specific dataset attributes. Using layers of fully connected nodes with ReLU activations and batch normalization, the generator produces synthetic data rows conditioned on the specified attributes. These samples are then passed to the discriminator $D(x|y)$, which evaluates both the synthetic and real data x alongside the conditional vector y to determine authenticity. The discriminator employs leaky ReLU activations and dropout to prevent overfitting and improve training stability (Xu *et al.*, 2019b).

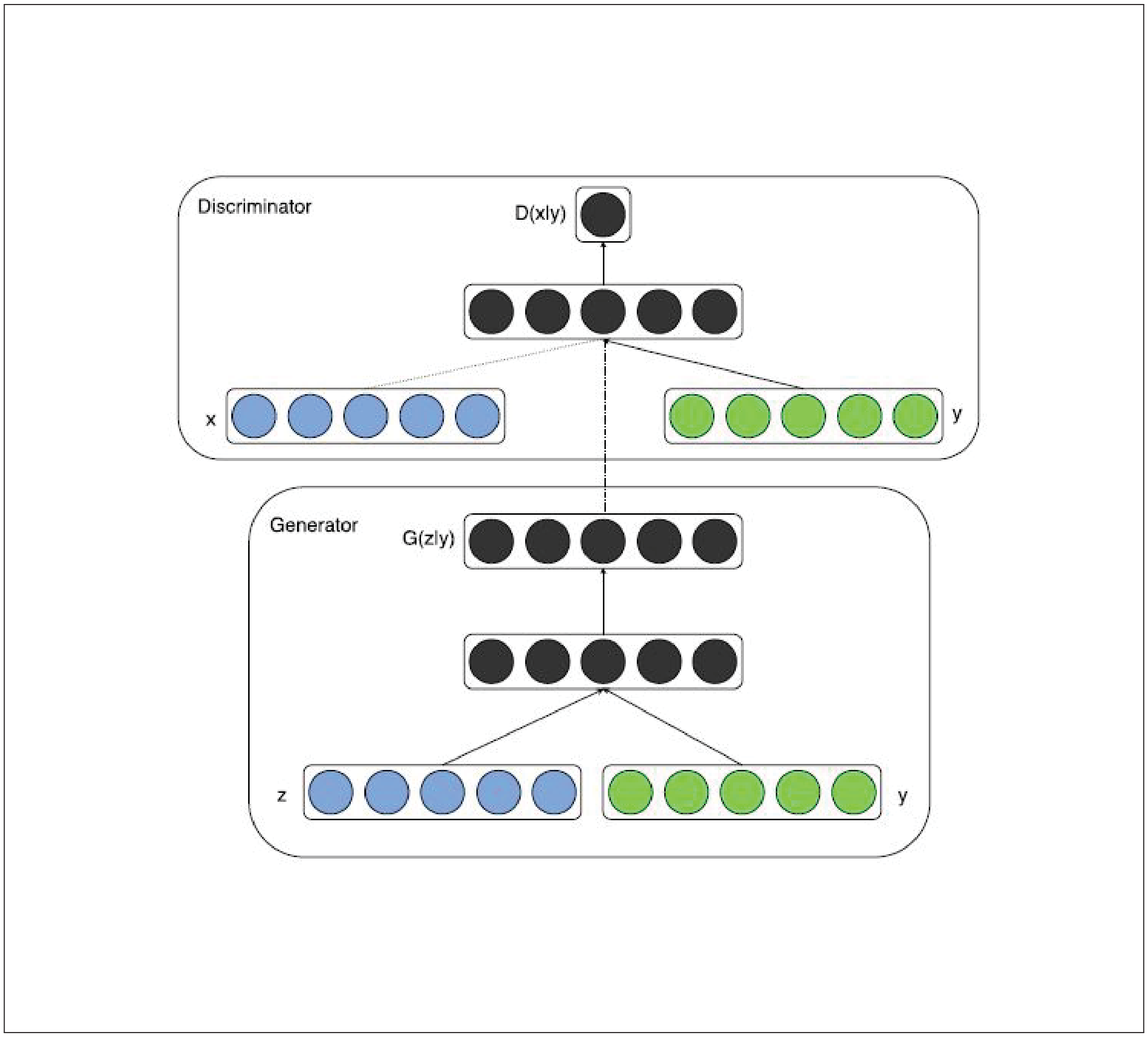
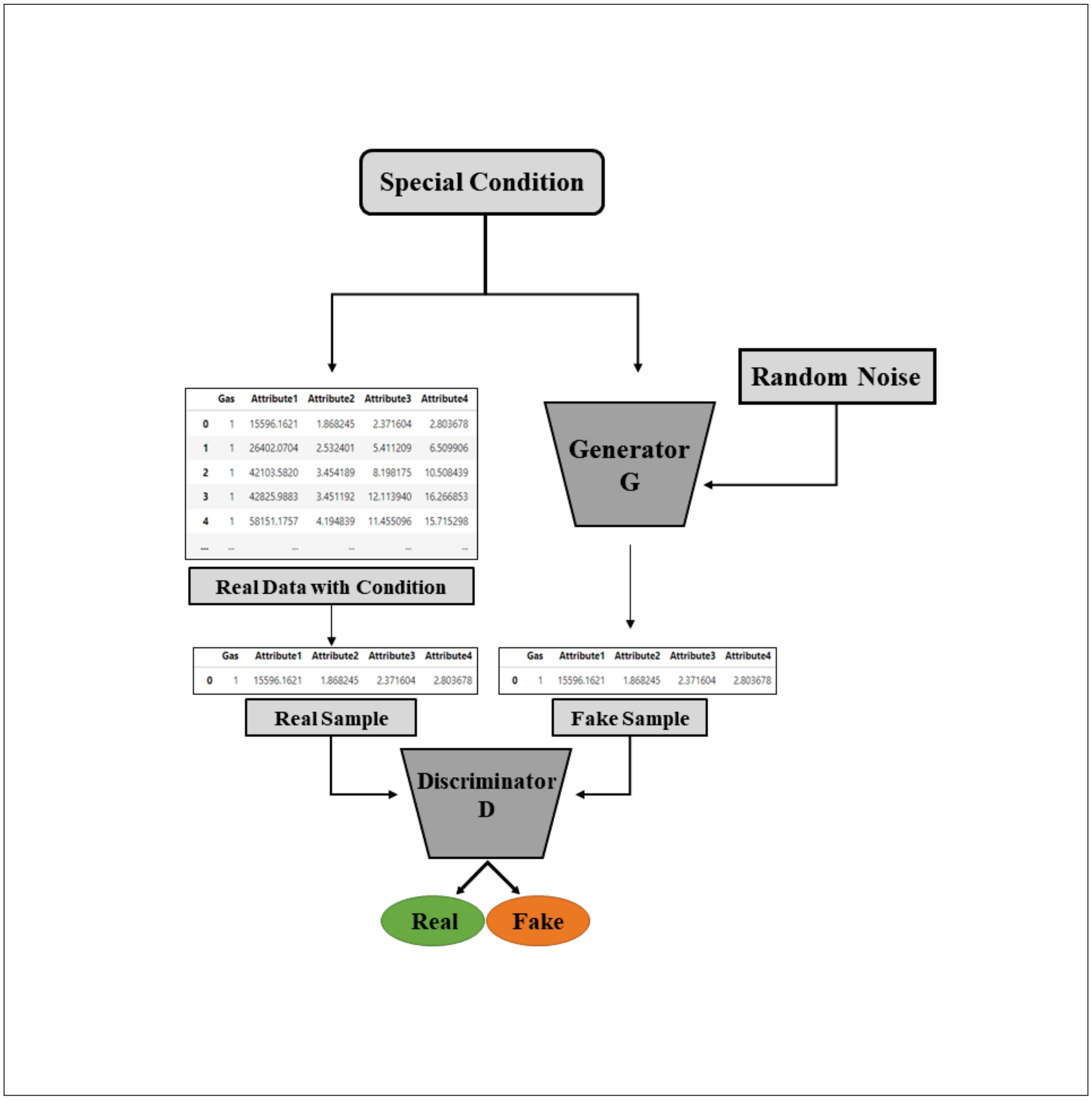


Figure 2.2 Overview of the conditional adversarial network architecture, illustrating how conditional inputs are incorporated to enhance data generation. Adapted from (Mirza, 2014)



	Gas	Attribute1	Attribute2	Attribute3	Attribute4
0	1	15596.1621	1.868245	2.371604	2.803678
1	1	26402.0704	2.532401	5.411209	6.509906
2	1	42103.5820	3.454189	8.198175	10.508439
3	1	42825.9883	3.451192	12.113940	16.266853
4	1	58151.1757	4.194839	11.455096	15.715298
...	...	...	...	...	...

	Gas	Attribute1	Attribute2	Attribute3	Attribute4
0	1	15596.1621	1.868245	2.371604	2.803678

	Gas	Attribute1	Attribute2	Attribute3	Attribute4
0	1	15596.1621	1.868245	2.371604	2.803678

Figure 2.3 Schematic representation of the CTGAN framework, showing the interaction between the generator and discriminator during training

This adversarial architecture, where the generator and discriminator compete during training, enables CTGAN to effectively generate high-quality synthetic data. The conditional mechanism further ensures that CTGAN can handle both continuous and categorical variables commonly found in complex gas sensor datasets. This capability is critical for data augmentation in gas detection systems, which often involve imbalanced and multimodal data distributions. By generating realistic and balanced data across all categories, CTGAN facilitates balancing the dataset by generating synthesized data points to achieve robust model training and improve gas detection performance.

2.3.2 Integrated Methodology for Sensor Drift and Data Imbalance Compensation

The proposed methodology addresses sensor drift and data imbalance by combining data preprocessing, synthetic data generation using CTGAN, and data augmentation techniques. The overall framework is illustrated in Fig. 2.4.

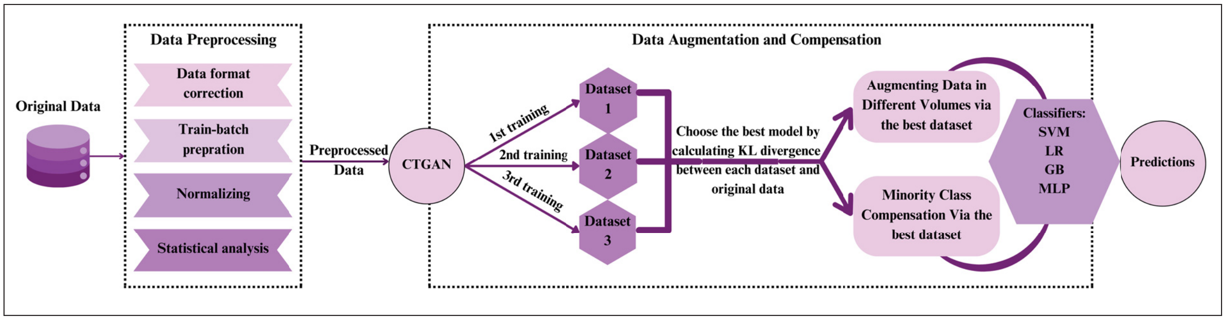

Original Data
Data Preprocessing
Data format correction
Train-batch prepration
Normalizing
Statistical analysis
Preprocessed Data
CTGAN
Data Augmentation and Compensation
1st training
2nd training
3rd training
Dataset 1
Dataset 2
Dataset 3
Choose the best model by calculating KL divergence between each dataset and original data
Augmenting Data in Different Volumes via the best dataset
Minority Class Compensation Via the best dataset
Classifiers:
SVM
LR
GB
MLP
Predictions

Figure 2.4 Proposed methodology for sensor drift detection and compensation. The framework combines data preprocessing, synthetic data generation using CTGAN, JS divergence-based drift detection, and data augmentation to enhance model performance

2.3.2.1 Data Preprocessing

Raw sensor data is prepared for downstream tasks through the following steps:

- **Data Format Correction:** Standardizing raw sensor outputs into a unified format for compatibility.
- **Train-Batch Preparation:** Dividing the dataset into sequential training batches to enable drift detection and tracking.
- **Normalization:** Scaling features to standard ranges, reducing variance and facilitating stable training.
- **Statistical Analysis:** Analyzing the dataset to uncover patterns, anomalies, or inconsistencies.

2.3.2.2 Synthetic Data Generation Using CTGAN

CTGAN is utilized to generate high-quality synthetic data to address class imbalance and mitigate sensor drift:

- **Architecture and Training:** CTGAN employs a conditional generative adversarial network architecture with fully connected layers, batch normalization, and dropout. Training is stabilized using Wasserstein GAN loss with gradient penalty and the Adam optimizer with a learning rate of 2×10^{-4}.
- **Dataset Creation:** Multiple synthetic datasets are generated by learning the joint probability distribution of features and labels, effectively representing both historical and current gas measurements.

2.3.2.3 Evaluation and Selection of Synthetic Data

The quality of each synthetic dataset is evaluated using the JS divergence. The dataset with the lowest JS divergence relative to the original data is selected as it best preserves the real data distribution.

2.3.2.4 Data Augmentation and Class Imbalance Compensation

Two strategies are employed to improve data quality using the selected synthetic dataset:

- **Data Augmentation:** Synthetic subsets, ranging from 10% to 100% of the original dataset size, are created and integrated into the training data to improve model robustness.
- **Minority Class Compensation:** Synthetic samples of underrepresented classes are merged with the original minority class samples to balance the dataset.

2.3.2.5 Model Training and Prediction

The augmented and balanced datasets are used to train ML models, including SVM, LR, GB, and MLP. These models are evaluated on test batches to measure their accuracy and resistance to drift.

2.3.2.6 Integrating Drift Detection with JS Divergence

JS divergence is calculated between consecutive batches to quantify drift. This metric identifies changes in data distributions over time, which guides the augmentation and compensation processes.

2.3.2.7 Outcomes

This methodology significantly improves the performance of ML models, even under sensor drift conditions. As illustrated in Fig. 2.4, the framework combines preprocessing, synthetic data generation, drift detection, and data augmentation to create a robust pipeline. Experimental results demonstrate up to a 20% improvement in classification accuracy for certain batches, with optimal augmentation volumes ranging from 40% to 80%.

2.4 Model Training and Evaluation

In this study, we evaluated four ML models to classify gases based on sensor data: SVM, MLP, Gradient Boosting (GB), and Logistic Regression (LR). Each model was fine-tuned using hyperparameter optimization and cross-validation to improve performance. The primary objective was to assess the models' ability to classify gases accurately, particularly in the context of imbalanced datasets where certain gases were underrepresented.

2.4.1 Model Training Process

The model training process involved three key steps:

1. **Data Preprocessing:** The sensor data was preprocessed to ensure consistency and improve model performance. This included normalizing sensor readings, handling missing values, and addressing class imbalance using techniques such as oversampling minority classes and class weighting.
2. **Hyperparameter Tuning:** Hyperparameters were optimized using grid search cross-validation, a systematic approach to identify the best parameter combinations for each model.

Examples of optimized hyperparameters include kernel type for SVM, hidden layer sizes for MLP, number of estimators for GB, and regularization strength for LR.

3. **Training and Evaluation:** Models were trained on the preprocessed dataset using the optimized hyperparameters. Evaluation metrics such as accuracy, precision, recall, and F1-score were calculated, with a focus on performance for minority classes. Confusion matrices were generated to visualize classification accuracy across different gas types.

2.4.2 Hyperparameter Summary

Table 2.2 summarizes the key hyperparameters and their values for each model. These settings were determined to yield the best performance during cross-validation.

Table 2.2 Optimized hyperparameters for the tested models

Model	**Hyperparameter**	**Value**
SVM	Kernel	{"RBF", "Linear"}
	Class Weight	{1:0.8, 2:1.0, 3:1.0, 4:0.2, 5:1.0, 6:1.5}
MLP	Hidden Layer Sizes	(50, 50), (100, 50), (100, 100)
	Alpha	0.01
	Learning Rate	0.01
	Activation Function	ReLU
LR	Regularization Strength (C)	1.0
GB	Number of Estimators	100

2.4.3 Evaluation Metrics

The evaluation focused on the following metrics:

- *Accuracy:* Measures the overall correctness of the model in classifying gases across all classes.
- *Precision:* Calculates the proportion of true positive predictions to total predicted positives, highlighting the model's reliability in avoiding false positives.

- *Recall:* Represents the proportion of true positives to total actual positives, emphasizing the model's effectiveness in detecting minority classes.
- *F1-score:* Combines precision and recall as their harmonic mean, offering a balanced assessment of the model's performance.

These metrics, combined with confusion matrices, provided a detailed evaluation of each model's classification performance. They allowed for the identification of strengths and limitations, particularly in addressing class imbalance and detecting minority gas classes.

2.4.4 Explanation of Model Choices

The following models were chosen based on their suitability for the characteristics of the gas sensor dataset:

- *SVM:* SVM was selected for its effectiveness in high-dimensional spaces, making it well-suited to the multi-feature gas sensor dataset. Both linear and radial basis function (RBF) kernels were tested to compare performance on linearly separable and non-linearly separable data. Class weighting was applied to address the imbalance in gas distributions.
- MLP: MLP was used to model the complex non-linear relationships inherent in the data. Varying hidden layer sizes and the ReLU activation function helped capture intricate patterns in sensor responses. Dropout was incorporated to mitigate overfitting, ensuring robust performance.
- GB: GB was chosen for its ability to iteratively build a strong classifier from weak learners (decision trees). Its focus on misclassified instances during training makes it particularly effective in handling class imbalance. GB's capacity to model non-linear relationships complements the characteristics of the dataset.
- LR: LR was selected as a baseline model for its simplicity and interpretability. Despite being linear, LR often provides a useful benchmark for binary and multi-class classification tasks. Regularization (parameter C) was applied to prevent overfitting, ensuring better generalization.

In the subsequent sections, we will provide a detailed analysis of each model's algorithm, training process, and performance on the gas sensor dataset. The models will be compared using accuracy, precision, recall, and F1-score, with an emphasis on their ability to handle class imbalance and capture complex patterns within the data.

2.4.5 SVM

SVMs are supervised learning models used for both classification and regression tasks. They are particularly effective in high-dimensional spaces, making them well-suited for datasets like gas sensor data, which have many features relative to the number of samples.

The goal of an SVM is to find the hyperplane that best separates the classes by maximizing the margin between the closest points of the two classes, known as support vectors. Given a dataset with n training examples $\{(x_i, y_i)\}$, where $x_i \in \mathbb{R}^d$ are feature vectors and $y_i \in \{-1, 1\}$ are class labels, the SVM optimization problem can be formulated as:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{2.5}$$

subject to the constraints:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, 2, \ldots, n \tag{2.6}$$

Here, $\mathbf{w}$ is the weight vector defining the orientation of the separating hyperplane, b is the bias term, and ξ_i are slack variables that allow for some misclassification in non-linearly separable data. The parameter $C > 0$ controls the trade-off between maximizing the margin and minimizing classification errors. A larger C encourages fewer misclassifications but risks overfitting, while a smaller C allows for a larger margin but increases tolerance for misclassifications.

To handle non-linearly separable data, SVM uses the kernel trick, which implicitly maps input data into a higher-dimensional space where a linear separation is possible. In this study, we used both the linear kernel and the RBF kernel. The RBF kernel is defined as:

$$K(x_i, x_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{2.7}$$

where γ is a kernel parameter that determines the influence of individual data points. A small γ produces a smoother decision boundary, while a large γ allows for a more complex decision boundary.

2.4.5.1 Hyperparameters

The SVM hyperparameters, including the kernel type, penalty parameter C, and class weights, are summarized in Table 2.3. These parameters were optimized using grid search cross-validation to balance the trade-off between classification accuracy and generalization:

Table 2.3 SVM Hyperparameters

Hyperparameter	**Value**
Kernel	Linear, RBF
Penalty Parameter C	1.0
Class Weight	{1:0.8, 2:1.0, 3:1.0, 4:0.2, 5:1.0, 6:1.5}

To address the imbalanced nature of the gas sensor dataset, a class weighting scheme was employed, assigning higher weights to minority classes (e.g., gases with fewer samples). This weighting encouraged the model to focus on correctly classifying underrepresented classes, reducing bias toward majority classes.

The performance of the SVM model was evaluated using cross-validation and metrics such as accuracy, precision, recall, and F1-score. Confusion matrices were also generated to provide detailed insights into the model's classification capabilities across different gas types.

2.4.6 MLP

MLPs are a class of artificial neural networks commonly used for supervised learning tasks like classification and regression. An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of neurons (nodes), and every neuron in a layer is connected to all neurons in the subsequent layer, forming a fully connected network.

In this study, MLPs were applied to classify gas sensor data, using sensor readings as input features for the network. The input layer of the MLP directly takes these measurements, and through transformations in the hidden layers, the network learns to map input data to the correct output class, corresponding to different gas types.

The architecture of the MLP used in this study includes:

- *Hidden layers:* The network configurations tested included architectures with varying numbers of neurons per layer, such as (50, 50), (100, 50), and (100, 100).
- *Activation function:* ReLU (Rectified Linear Unit) was used for all neurons in the hidden layers to introduce non-linearity, enabling the model to learn complex patterns.
- *Regularization:* Dropout was applied during training to prevent overfitting by randomly setting a fraction of neurons to zero in each layer.
- *Optimization:* The network was trained using the Adam optimizer with a learning rate of 0.01, which balances convergence speed and stability.

The MLP was evaluated on the same metrics as the SVM model, including accuracy, precision, recall, and F1-score, with additional emphasis on its ability to generalize across imbalanced classes. By leveraging its capacity to learn complex non-linear relationships, the MLP was expected to capture intricate patterns in the gas sensor data, improving classification performance on minority classes.

2.4.6.1 Network Architecture

The architecture of an MLP is defined by the number of layers and the number of neurons in each layer. In this study, we experimented with various configurations to determine the optimal architecture for the gas sensor dataset. Specifically, hidden layers with 50, 100, and 150 neurons were tested. The final architecture was selected based on cross-validation results to achieve a balance between complexity and generalization, as detailed in Table 2.4.

Each neuron in the network applies a non-linear activation function to its inputs. For the hidden layers, we used the ReLU (Rectified Linear Unit) activation function, which is computationally efficient and helps mitigate the vanishing gradient problem:

$$f(x) = \max(0, x) \tag{2.8}$$

For the output layer, a softmax activation function was employed to convert the outputs into probability distributions across multiple gas classes:

$$P(y = j|x) = \frac{\exp(\mathbf{w}_j^\top \mathbf{x})}{\sum_{k=1}^{K} \exp(\mathbf{w}_k^\top \mathbf{x})} \tag{2.9}$$

Here, $P(y = j|x)$ represents the probability of class j given input x, and K is the total number of classes.

2.4.6.2 Training Process

The MLP was trained using backpropagation, where the network weights were updated by minimizing the categorical cross-entropy loss function:

$$L = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(\hat{y}_{ik}) \tag{2.10}$$

In this equation, y_{ik} is the true label for class k of sample i, and $\hat{y}_{ik}$ is the predicted probability for class k.

To optimize the weights, we employed the Adam optimizer, a variant of stochastic gradient descent, with a learning rate of 0.001. Adam dynamically adjusts the learning rate during training and efficiently handles sparse gradients, making it well-suited for this dataset (Mahinnezhad *et al.*, 2024).

Table 2.4 MLP Hyperparameters

Hyperparameter	**Value**
Hidden Layer Sizes	(50, 50), (100, 50), (100, 100)
Alpha (Regularization Term)	0.01
Learning Rate	0.01
Maximum Iterations	1000
Activation Function	ReLU

2.4.6.3 Regularization and Overfitting

To mitigate overfitting, dropout regularization was employed. Dropout works by randomly deactivating a fraction of neurons during training, forcing the network to generalize better and reducing its reliance on specific neurons. In this study, a dropout rate of 0.5 was used, meaning 50% of the neurons were deactivated during each training iteration.

Early stopping was also implemented as a regularization technique. The validation loss was monitored, and training was halted when the loss ceased to improve for a predefined number of epochs. This approach ensured that the model did not overfit to the training data while maintaining high generalization to unseen data.

2.4.7 GB

GB is a powerful ensemble learning technique widely used for regression and classification tasks. The key idea behind GB is to iteratively build an ensemble of weak learners, typically decision trees, where each successive model focuses on correcting the errors made by the previous models. This sequential approach effectively reduces bias and enhances predictive performance.

At each iteration, GB minimizes a loss function that quantifies the model's fit to the data. A new weak model is fitted to the negative gradient of the loss function, aligning the updates with the direction of steepest descent, hence the term "gradient boosting." The overall prediction is updated by incorporating the new model's contribution, weighted by a learning rate parameter. The learning rate controls the magnitude of updates, balancing the ensemble's ability to converge and avoid overfitting.

- Weak Learners: Decision trees with shallow depths (e.g., 3-5 levels) are typically used as the weak learners in GB. Their simplicity helps prevent overfitting while capturing key patterns in the data.
- Learning Rate: This parameter determines the contribution of each weak learner to the overall model. Smaller learning rates (e.g., 0.1) are often paired with a higher number of iterations to achieve optimal performance.
- Regularization: Techniques such as limiting tree depth, using subsampling (random selection of data points), and applying learning rate decay help prevent overfitting and improve generalization.

GB's ability to model non-linear relationships and focus on challenging instances makes it particularly effective for handling the imbalanced and complex patterns in the gas sensor dataset. Its iterative learning process allows it to progressively improve performance, addressing classification challenges posed by the minority classes and subtle drift effects in the data.

2.4.7.1 Loss Function and Optimization

The GB algorithm optimizes a loss function $L(y, f(x))$, where y is the true value and $f(x)$ is the model's prediction. The choice of the loss function depends on the task:

- For regression tasks, the most commonly used loss function is the squared error:

$$L(y, f(x)) = \frac{1}{2}(y - f(x))^2 \tag{2.11}$$

- For classification tasks, the logistic loss is frequently used for binary classification:

$$L(y, f(x)) = \log(1 + \exp(-yf(x))) \tag{2.12}$$

The GB algorithm minimizes the loss function iteratively by fitting new models to the negative gradient of the loss with respect to the current predictions. This process can be mathematically expressed as:

$$f_{m+1}(x) = f_m(x) + \eta \cdot h(x) \tag{2.13}$$

Where:

- $f_m(x)$ is the current model at iteration m,
- $h(x)$ is the new weak learner (e.g., a decision tree) fitted to the negative gradient,
- η is the learning rate, controlling the contribution of the new learner to the overall model.

This iterative approach allows the model to progressively improve its performance by correcting the residual errors from previous iterations.

2.4.7.2 Regularization and Preventing Overfitting

To prevent overfitting, GB incorporates several regularization techniques:

- *Shrinkage:* A learning rate (η) is applied to control the contribution of each new weak learner to the ensemble. This is expressed as:

$$f_{m+1}(x) = f_m(x) + \eta \cdot h(x) \tag{2.14}$$

  Smaller values of η (e.g., 0.1) often require more iterations but improve generalization by reducing overfitting.
- *Early Stopping:* Training is halted when the validation error stops improving, preventing the model from overfitting to the training data.
- *Subsampling:* At each iteration, a random subset of the training data is used to fit the weak learner. This reduces variance and improves generalization by introducing randomness into the learning process (Natekin & Knoll, 2013b).

These regularization techniques collectively enhance the robustness of GB, making it effective for handling complex datasets like the gas sensor data while maintaining generalization and avoiding overfitting.

2.4.7.3 Hyperparameters

The performance of GB is highly sensitive to its hyperparameters. In this study, several hyperparameters were tuned to achieve an optimal balance between model complexity and generalization. The key hyperparameters, along with their values, are shown in Table 2.5:

Table 2.5 GB Hyperparameters

Hyperparameter	**Value**
Number of Trees (*n_estimators*)	100
Learning Rate (η)	0.01
Max Depth	3
Subsample	0.8

- *Number of Trees (n_estimators):* Set to 100 to provide sufficient capacity for learning patterns in the data while avoiding overfitting.

- *Learning Rate (η):* Set to 0.01 to allow gradual learning, improving generalization by controlling the contribution of each tree to the final model.
- *Maximum Depth:* Limited to 3 to prevent overfitting by restricting the complexity of each decision tree.
- *Subsample:* Set to 0.8 to reduce variance by training each tree on 80% of the data, introducing randomness and improving generalization (Mahinnezhad *et al.*, 2024).

These settings ensured that the GB model maintained a balance between learning intricate patterns in the data and avoiding overfitting, making it effective for classifying the gas sensor data.

2.4.8 LR

LR is a widely used algorithm for binary and multiclass classification problems. Unlike linear regression, which predicts continuous outcomes, LR models the probability of a categorical outcome using the logistic function. It is particularly effective for categorical target variables, even when the relationship between predictors and the target is non-linear.

The probability of the outcome $y = 1$ is estimated using the logistic function:

$$P(y = 1|x) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n))} \tag{2.15}$$

where:

- $P(y = 1|x)$ is the probability of $y = 1$ given predictors x,
- β_0 is the intercept,
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for predictors $x_1, x_2, \ldots, x_n$.

The odds of the outcome are modeled as a linear combination of the predictors:

$$\log\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \tag{2.16}$$

2.4.8.1 Hyperparameters and Model Training

In this study, LR was applied to predict gas classification outcomes using both the original imbalanced dataset and the augmented dataset generated by CTGAN. Hyperparameters were carefully tuned to optimize the model's performance while managing overfitting.

- **Penalty (Regularization Type)**: L2 regularization (Ridge) was used to shrink large coefficients and reduce overfitting.
- **Regularization Parameter** (*C*): Set to 1.0, balancing the trade-off between bias and variance.
- **Solver**: The liblinear solver was selected for its efficiency in handling small to medium-sized datasets.

The hyperparameters are summarized in Table 2.6.

Table 2.6 Hyperparameters for LR.

Hyperparameter	**Value**
Penalty	L2 (Ridge)
Regularization Parameter (C)	1.0
Solver	liblinear

CHAPTER 3

RESULTS

This chapter presents the results of the proposed methodologies, highlighting sensor drift detection using JS divergence and the effects of data balancing and augmentation on classification performance.

3.1 Sensor Drift Detection

To evaluate sensor drift in the gas detection system, JS divergence was utilized. JS divergence quantified shifts in sensor response distributions over time. Additionally, confusion matrices assessed the system's classification accuracy before and after drift detection.

3.1.1 Evaluation of Drift Using JS Divergence

In this section, we evaluate sensor drift and its impact on classification performance using the JS divergence metric. Sensors 1 to 8 are selected for this analysis because they provide a diverse and representative subset of the dataset. The dataset consists of four distinct types of sensors, with four sensors from each type, resulting in a total of 16 sensors. By selecting two sensors from each type, we capture a broad range of drift behaviors while avoiding redundancy. This selection ensures a thorough and efficient analysis of sensor performance.

Gas 6 is excluded from this analysis because it is not consistently present across all batches. Including gas 6 would compromise the comparability of results, as the calculation of JS divergence requires the presence of all gases in each batch. To maintain consistency and ensure the reliability of our findings, we restrict the analysis to gases 1 through 5.

Figure 3.1 shows that sensor 1 exhibits stable JS divergence values across the first five batch pairs, indicating minimal drift. However, from batch pair 6 onwards, the JS divergence begins to rise, peaking at batch pair 10. This increase signals the onset of sensor drift, which corresponds to increased misclassification rates observed in the later confusion matrices.

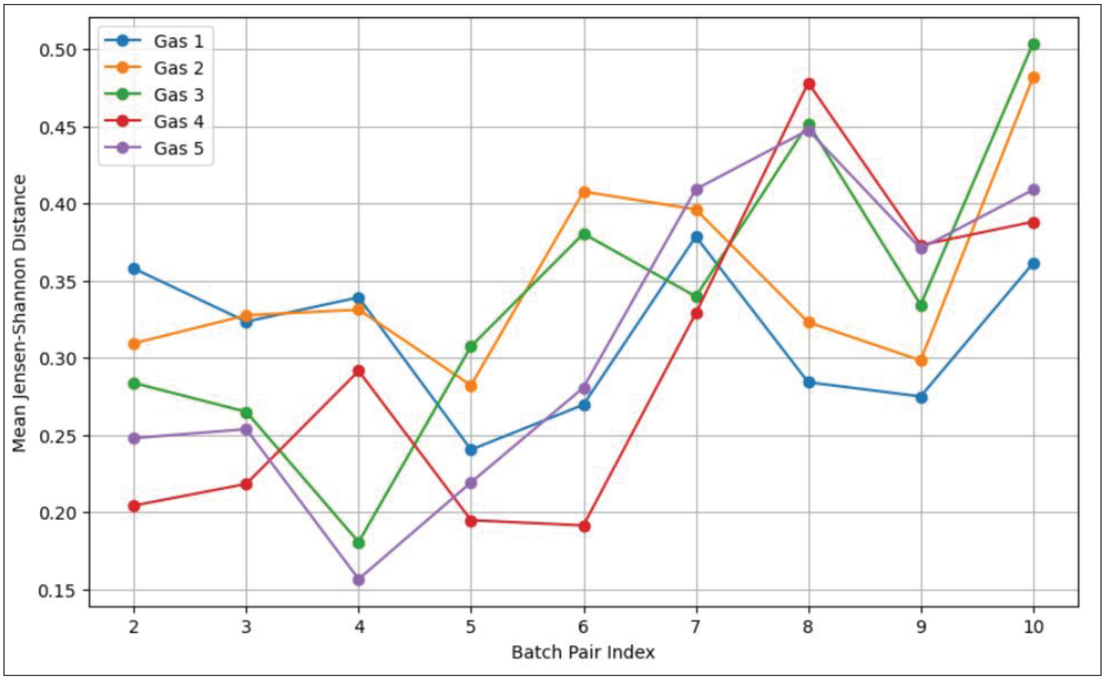


Figure 3.1 JS divergence values between accumulated batches and the subsequent batch for sensor 1

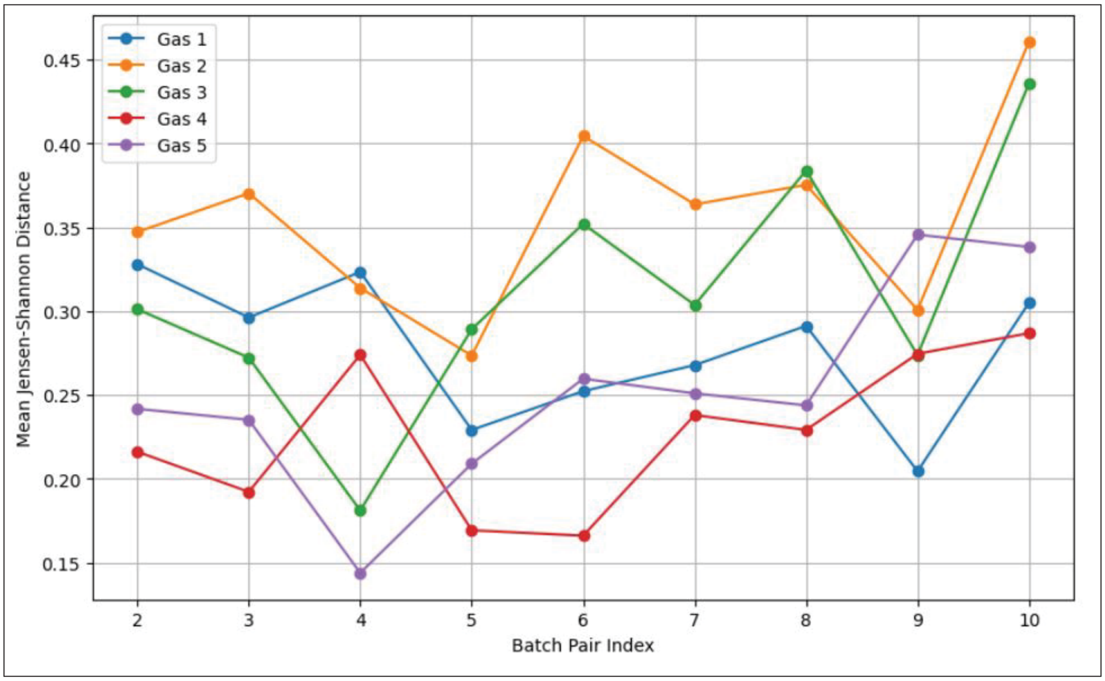


Figure 3.2 JS divergence values between accumulated batches and the subsequent batch for sensor 2

As shown in Fig 3.2, sensor 2 demonstrates relatively stable divergence values initially. However, a noticeable upward trend begins at batch pair 7 and culminates in a significant rise by batch pair 10. This pattern suggests gradual drift over time, which may negatively impact classification accuracy in later batches.

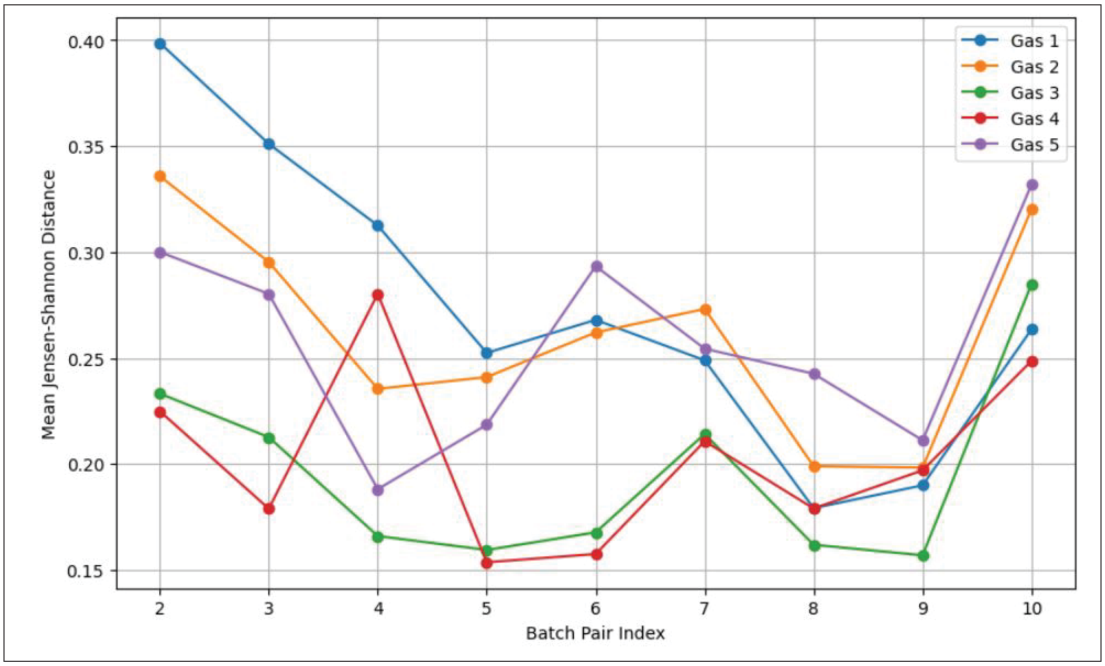


Figure 3.3 JS divergence values between accumulated batches and the subsequent batch for sensor 3

For sensor 3, as shown in Fig 3.3, the JS divergence remains stable until batch pair 5, after which a sudden spike occurs. This spike is followed by a consistent upward trend, indicating the abrupt onset of drift and its potential impact on classification accuracy.

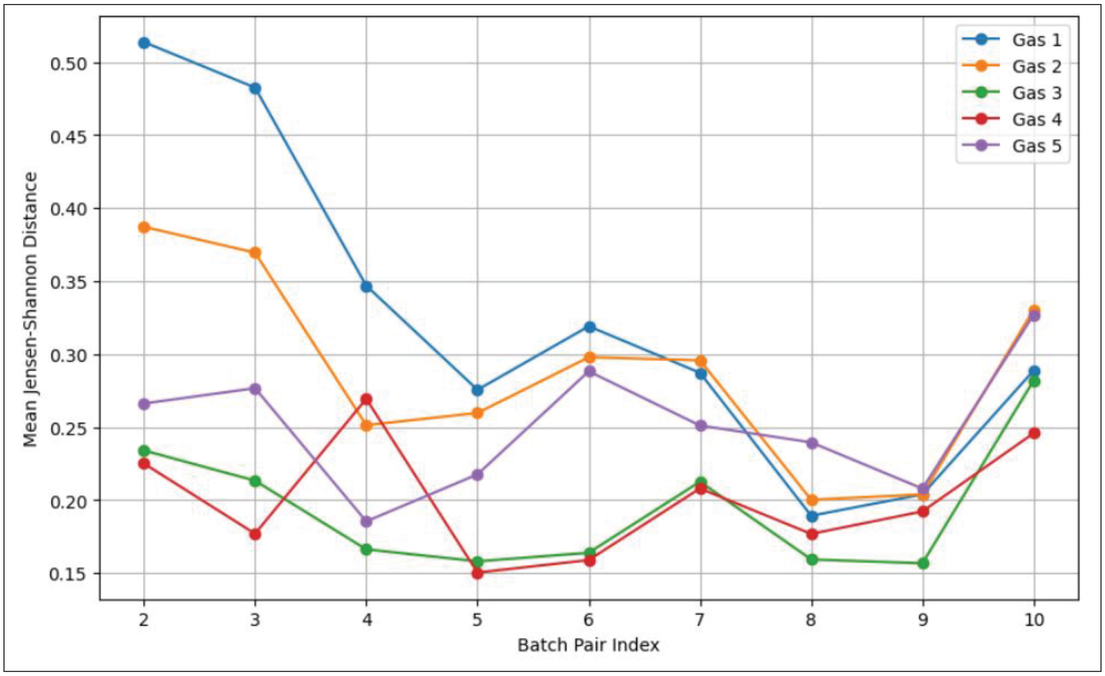


Figure 3.4 JS divergence values between accumulated batches and the subsequent batch for sensor 4

Sensor 4 (Fig 3.4) demonstrates much greater stability, with relatively low JS divergence values across most batch pairs. A slight increase starting at batch pair 8 suggests minimal drift, making sensor 4 one of the most stable sensors in this analysis.

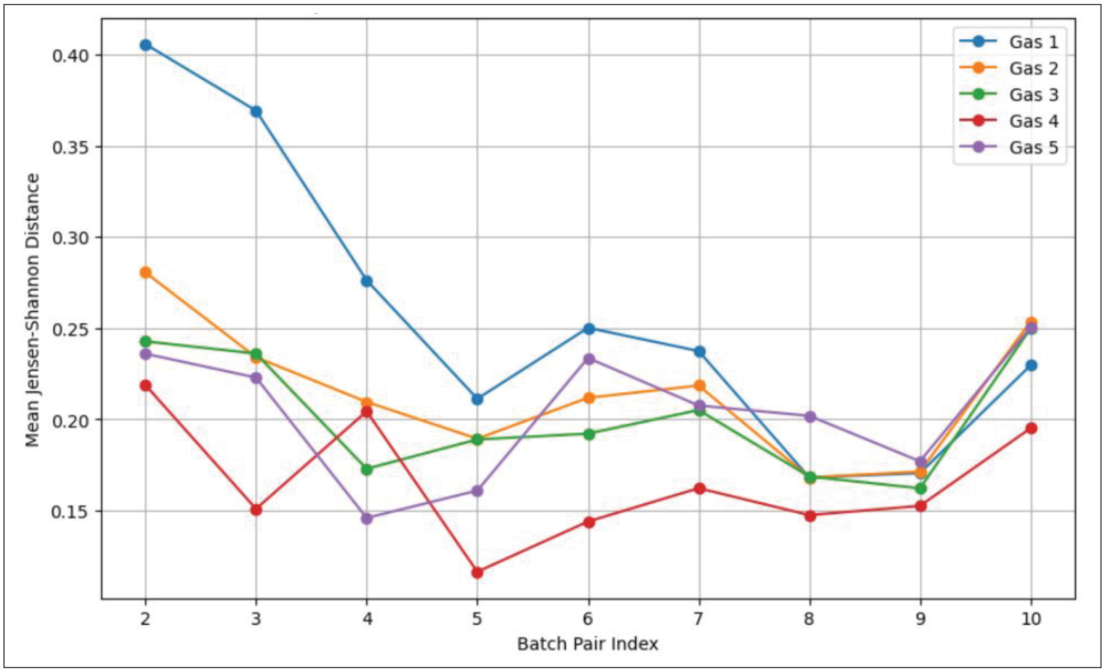


Figure 3.5 JS divergence values between accumulated batches and the subsequent batch for sensor 5

As depicted in Fig 3.5, sensor 5 experiences significant drift, with JS divergence values rising sharply from batch pair 6 onwards. By batch pair 10, sensor 5 records one of the highest divergence values among all sensors, highlighting a critical degradation in performance.

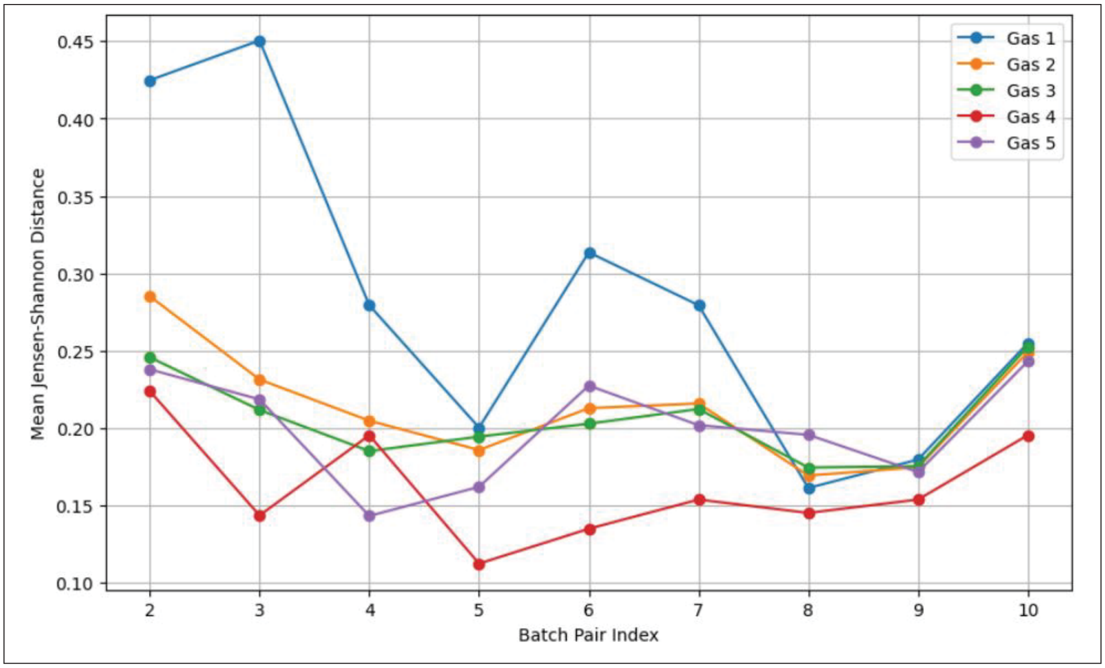


Figure 3.6 JS divergence values between accumulated batches and the subsequent batch for sensor 6

Sensor 6 (Fig 3.6) exhibits a sharp drop in JS distance from batch pair 2 to 5, followed by a gradual increase starting from batch pair 6. This suggests initial stability with the onset of gradual drift in later batches.

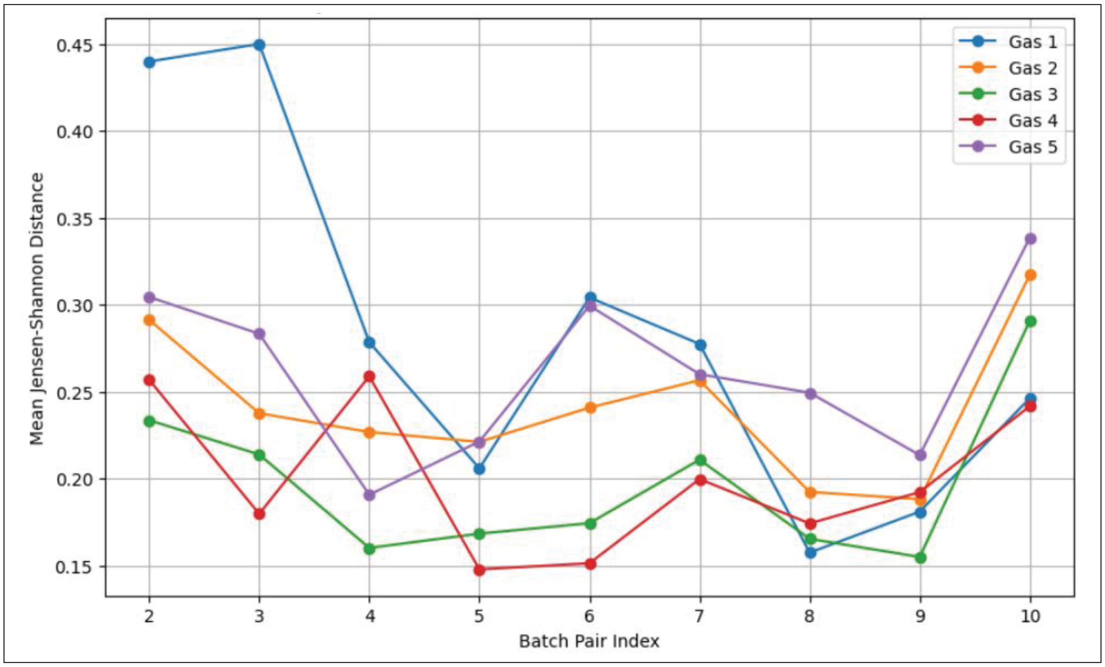


Figure 3.7 JS divergence values between accumulated batches and the subsequent batch for sensor 7

As shown in Fig 3.7, sensor 7 maintains relatively stable JS distances up to batch pair 5. Starting from batch pair 6, the divergence gradually increases, with a pronounced rise in the final batch pairs.

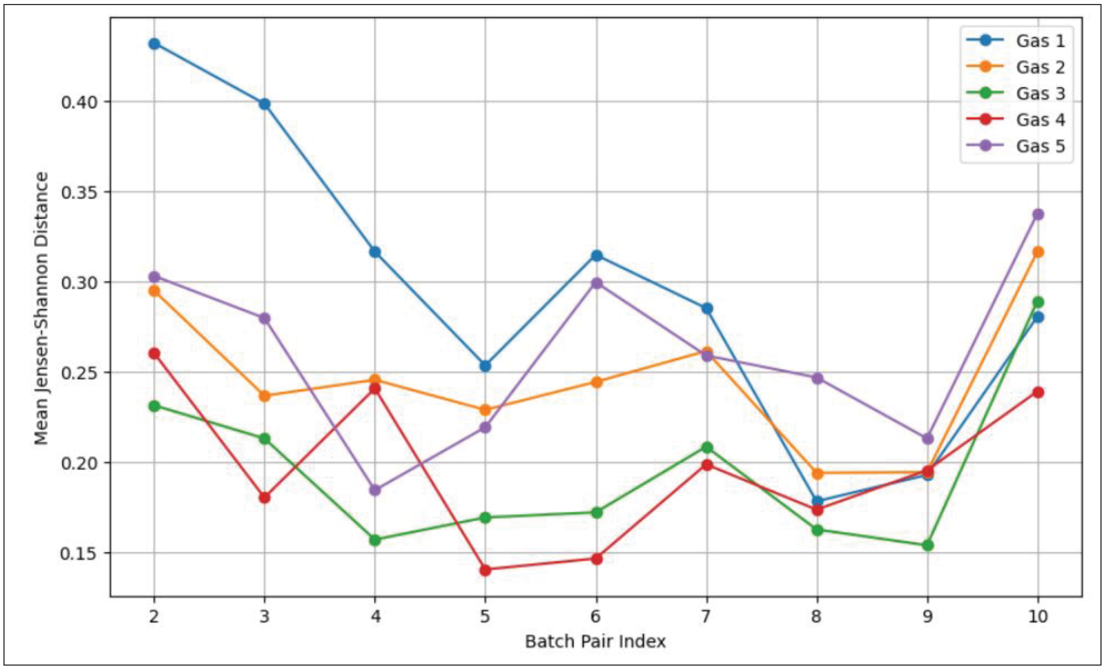


Figure 3.8 JS divergence values between accumulated batches and the subsequent batch for sensor 8

Sensor 8 (Fig 3.8) demonstrates a unique behavior, where JS distances are relatively low across early batch pairs, followed by a gradual increase starting from batch pair 6. By batch pair 10, the divergence peaks, suggesting significant drift.

3.1.1.1 General Observations

From the analysis, we observe that most sensors exhibit minimal JS divergence during early batch pairs, indicating stable performance. However, drift becomes apparent in later batch pairs, manifesting as either abrupt or gradual increases in JS divergence. Notably, sensors like 5 show sharp rises, while others, such as sensor 4, experience more gradual changes. These increases in JS divergence correlate with higher misclassification rates in the confusion matrices, emphasizing the need for monitoring and mitigating sensor drift to maintain classification accuracy.

3.1.2 Drift Detection in Batches

This section evaluates the impact of sensor drift on classification performance across different batches using confusion matrices. These matrices illustrate how accurately the system distinguishes between gases and reveal trends in misclassification as sensor drift progresses. By analyzing batches 2 to 10, we demonstrate the evolving performance of the system and its susceptibility to drift.

The confusion matrices complement the JS divergence analysis presented earlier. While JS divergence highlights distributional shifts at the sensor level, confusion matrices provide a batch-level overview of classification performance. Together, these metrics help connect sensor-specific drift to the overall performance of the classification system.

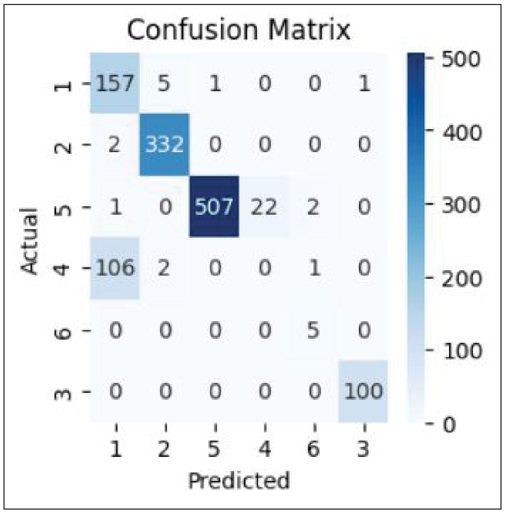


Figure 3.9 Confusion matrix for Batch 2

Batch 2 (Fig 3.9): The confusion matrix for batch 2 shows high classification accuracy for most gases. Gas 5 has the highest accuracy, with 507 correct predictions, while gas 2 follows with 332 correct predictions and minimal misclassifications. However, gas 4 exhibits noticeable misclassification, with 106 samples misclassified as gas 1. This indicates that while the system performs well overall, certain gases are more prone to drift-induced errors even at this early stage.

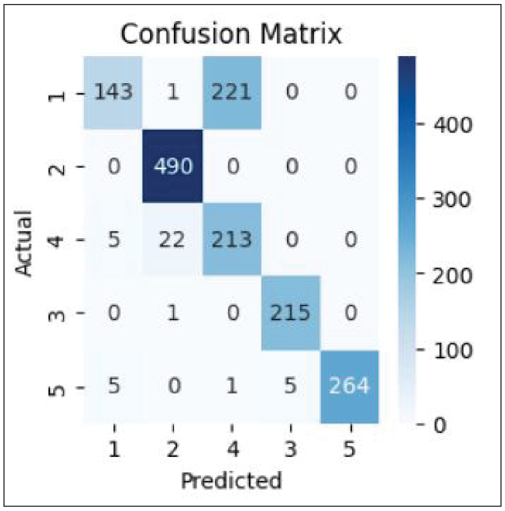


Figure 3.10 Confusion matrix for Batch 3

Batch 3 (Fig 3.10): In batch 3, misclassification rates increase, especially for gases 1 and 4. Gas 1 has 221 samples misclassified as gas 3, while gas 4 sees 22 samples misclassified as gas 3. Gas 2 maintains strong accuracy, with 490 correct predictions. These results suggest the onset of drift, particularly for classes with overlapping distributions.

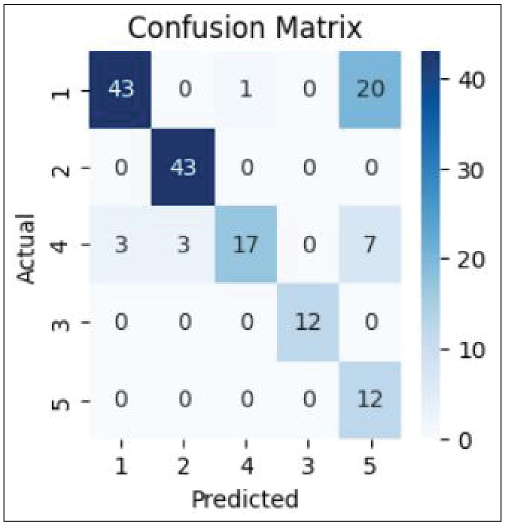


Figure 3.11 Confusion matrix for Batch 4

Batch 4 (Fig 3.11): The confusion matrix for batch 4 highlights further performance degradation. Gas 1 suffers significant misclassification, with 20 samples misclassified as gas 5. Gas 4 also experiences increased errors, with only 17 correct predictions and 7 misclassified as gas 5. This reflects the increasing challenge posed by drift as distributions shift over time.

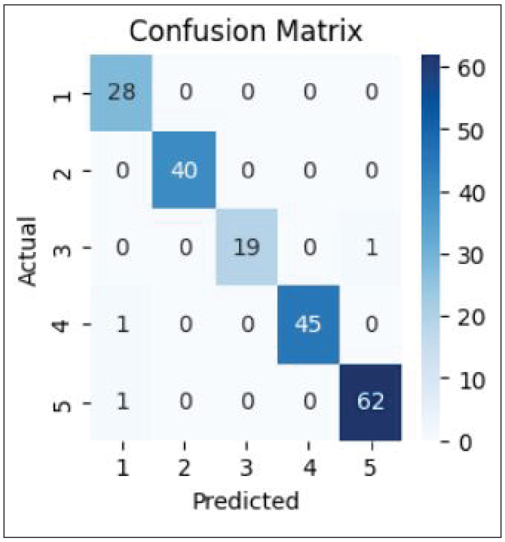


Figure 3.12 Confusion matrix for Batch 5

Batch 5 (Fig 3.12): In batch 5, classification accuracy continues to decline for some gases, while gas 5 remains relatively robust with 62 correct predictions. Gas 1, however, shows a significant drop, with only 28 correct predictions and increased misclassification as gas 4. This trend underscores the growing impact of drift on system performance.

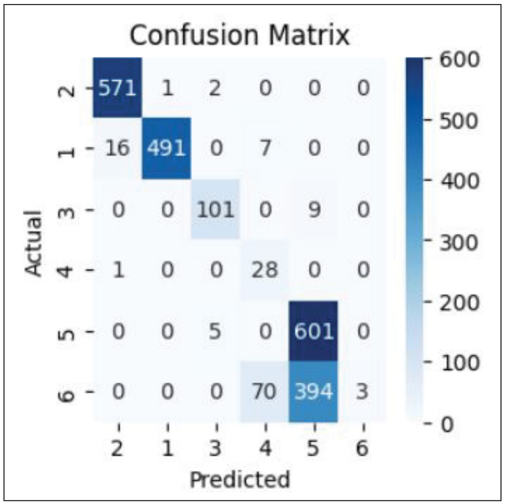


Figure 3.13 Confusion matrix for Batch 6

Batch 6 (Fig 3.13): The confusion matrix for batch 6 reflects the cumulative effects of drift. Gas 6 shows significant misclassification, with 70 samples misclassified as gas 5. However, gases 2 and 5 retain relatively high accuracy, with 571 and 601 correct predictions, respectively. This suggests that while some classes remain robust, others become increasingly vulnerable to drift-induced errors.

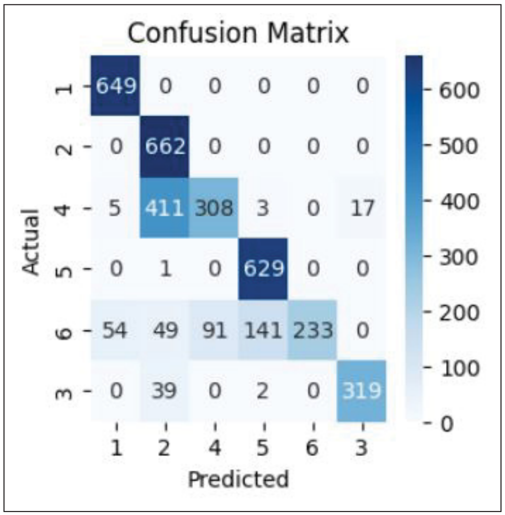


Figure 3.14 Confusion matrix for Batch 7

Batch 7 (Fig 3.14): In batch 7, misclassification becomes more pronounced for gases 3 and 6. Gas 4 shows a significant decline, with only 411 correct predictions and 308 misclassified as gas 6. Gas 5 maintains strong performance, with 629 correct predictions, but other gases show signs of increasing drift.

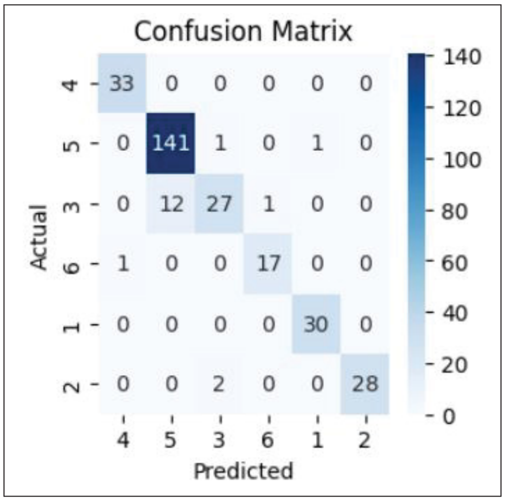


Figure 3.15 Confusion matrix for Batch 8

Batch 8 (Figure 3.15): The confusion matrix for batch 8 highlights significant errors, particularly for gas 5, which has 12 samples misclassified as gas 3. Gas 4 maintains strong accuracy, with 141 correct predictions, suggesting that certain gases remain robust despite the increasing drift.

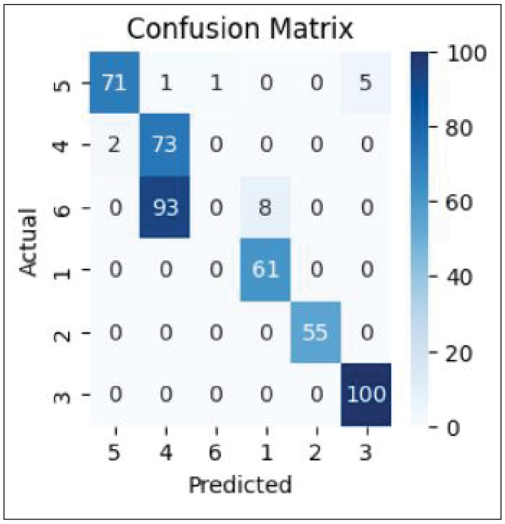


Figure 3.16 Confusion matrix for Batch 9

Batch 9 (Fig 3.16): Batch 9 shows an overall decline in accuracy across most gases. Gas 3 achieves only 55 correct predictions, while gas 5 experiences increased misclassification, with 5 samples misclassified as gas 3. These results reflect the cumulative impact of drift, which leads to blurred decision boundaries and overlapping distributions.

Batch 10 (Fig 3.17): By batch 10, the confusion matrix illustrates the full extent of drift-induced performance degradation. Gas 1 has 208 samples misclassified as gas 4, while gas 4 shows 217 misclassifications into gas 6. Gas 5 retains relatively strong performance, with 496 correct predictions, indicating its robustness against drift.

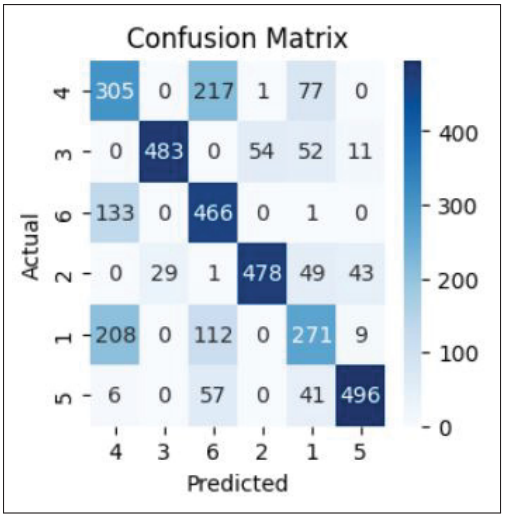


Figure 3.17 Confusion matrix for Batch 10

General Observations: Across batches 2 to 10, the confusion matrices reveal a clear trend of increasing misclassification as sensor drift progresses. Early batches show high accuracy with minor errors, but later batches experience significant performance degradation, particularly for gases with overlapping distributions. These trends align closely with the JS divergence analysis, where certain sensors exhibited greater drift over time.

By combining the insights from confusion matrices and JS divergence, we gain a comprehensive understanding of how drift affects both sensor-level distributions and system-wide classification accuracy. This dual approach highlights the importance of monitoring and mitigating drift to maintain robust performance in real-world applications.

3.2 Approach 1: Addressing Class Imbalance with CTGAN

In this study, CTGAN-based augmentation yielded notable improvements across several classification models, including SVM, MLP, and GB. For instance, classification accuracy improved by up to 20% in specific batches, particularly in datasets where drift and imbalance were most pronounced. The highest gains were observed in batches with augmentation levels ranging from 40% to 80%, suggesting that this range was particularly effective for enhancing classifier robustness against data imbalance.

Additionally, CTGAN's approach to generating diverse synthetic data led to a more balanced representation of gas types in each batch. This balanced distribution was validated by comparing correlation matrices of the synthetic and real data, showing high similarity, which confirmed the synthetic data's quality. The results indicate that CTGAN not only addressed class imbalance but also enhanced overall classification accuracy, even in the presence of sensor drift.

3.2.1 Classification Accuracy Before Applying CTGAN

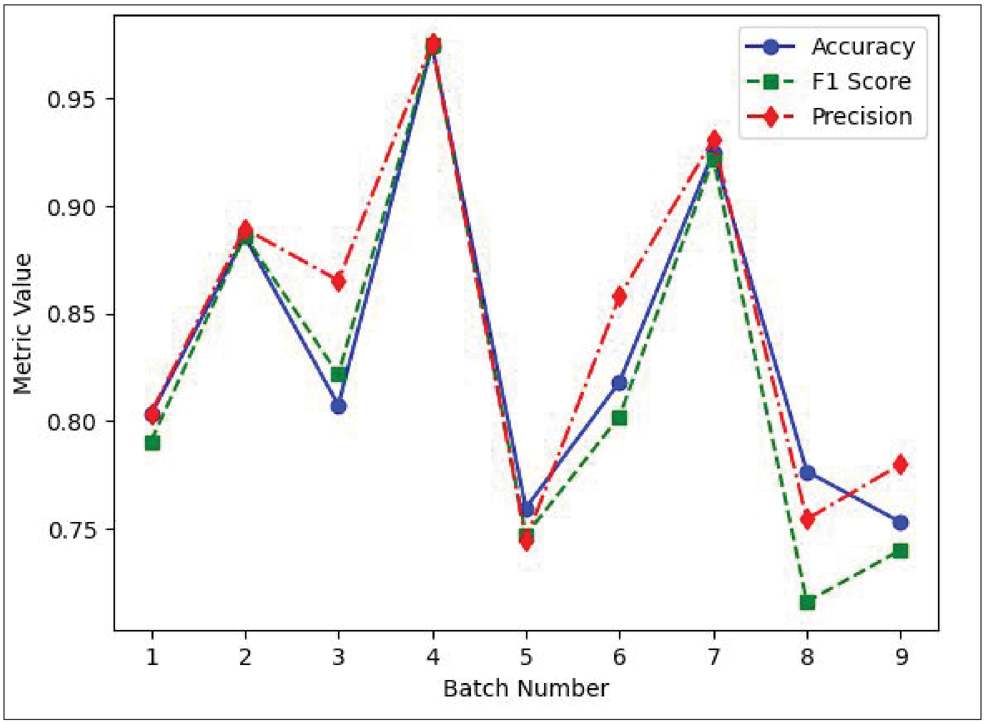


Figure 3.18 Classification metrics for SVM model before applying CTGAN across batches

In Figure 3.18, which illustrates the SVM classifier's performance across batches, significant fluctuations are observed in the metrics before applying CTGAN data augmentation. Acc (blue line) and F1 (green line) generally follow similar patterns, reflecting SVM's balanced handling of true positives and negatives. Prec (red line) tends to be higher than both Acc and F1, indicating

SVM's conservative approach to positive predictions, thereby reducing false positives. Sharp declines, particularly in batch 5, highlight the impact of drift, where SVM's generalization capability is compromised due to shifts in data distribution and imbalance.

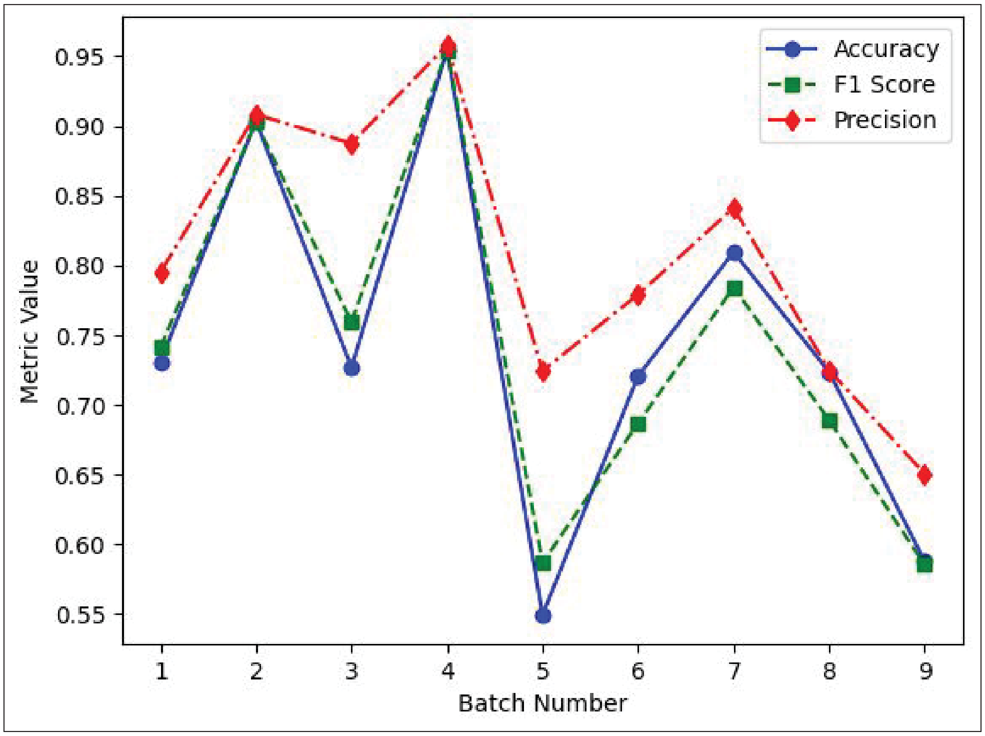


Figure 3.19 Classification metrics for GB model before applying CTGAN across batches

In Fig 3.19, which shows the GB classifier, a similar trend emerges, with peaks in batches 4 and 7 where all metrics reach optimal levels. However, GB experiences notable dips, especially in batch 5, where all metrics drop significantly, underscoring its susceptibility to drift. The alignment of F1 with Acc indicates that GB balances true positives and negatives well, though Prec consistently remains higher, indicating GB's focus on minimizing false positives to counteract drift and imbalance.

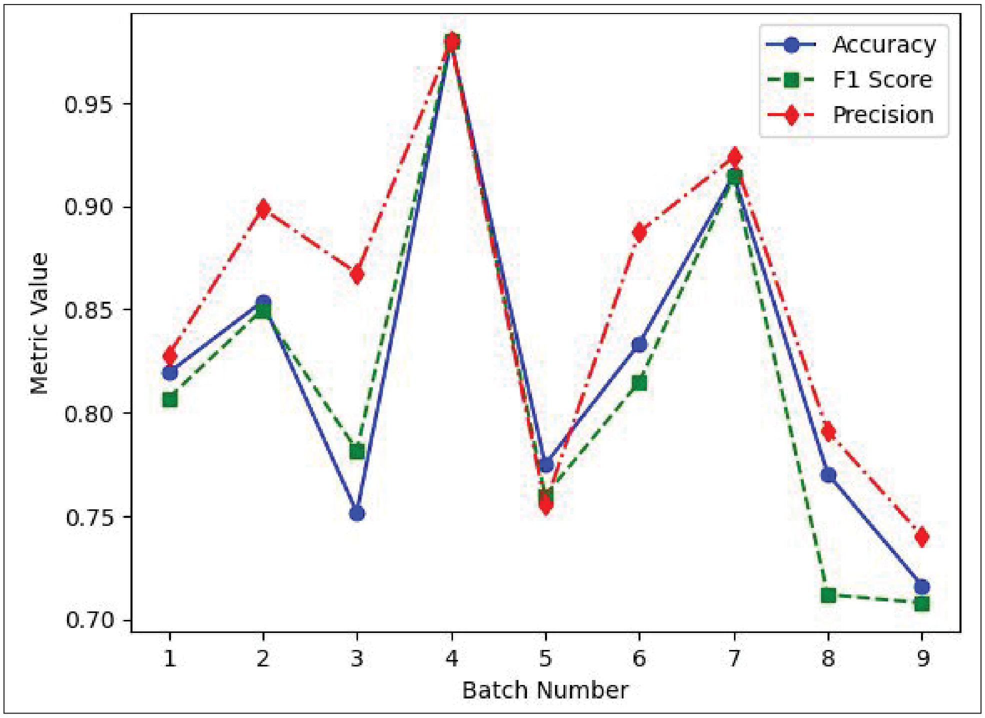


Figure 3.20 Classification metrics for LR model before applying CTGAN across batches

Figure 3.20 presents the LR classifier, displaying a comparable trend, with performance peaks in batches 4 and 7, indicating stable classification under minimal drift conditions. Like SVM and GB, batch 5 poses a significant challenge, with all metrics dropping sharply. Prec remains generally higher than Acc and F1, suggesting LR's conservative approach to positive predictions. The close tracking of F1 with Acc further implies a balanced precision-recall dynamic, though LR struggles with consistency under drift.

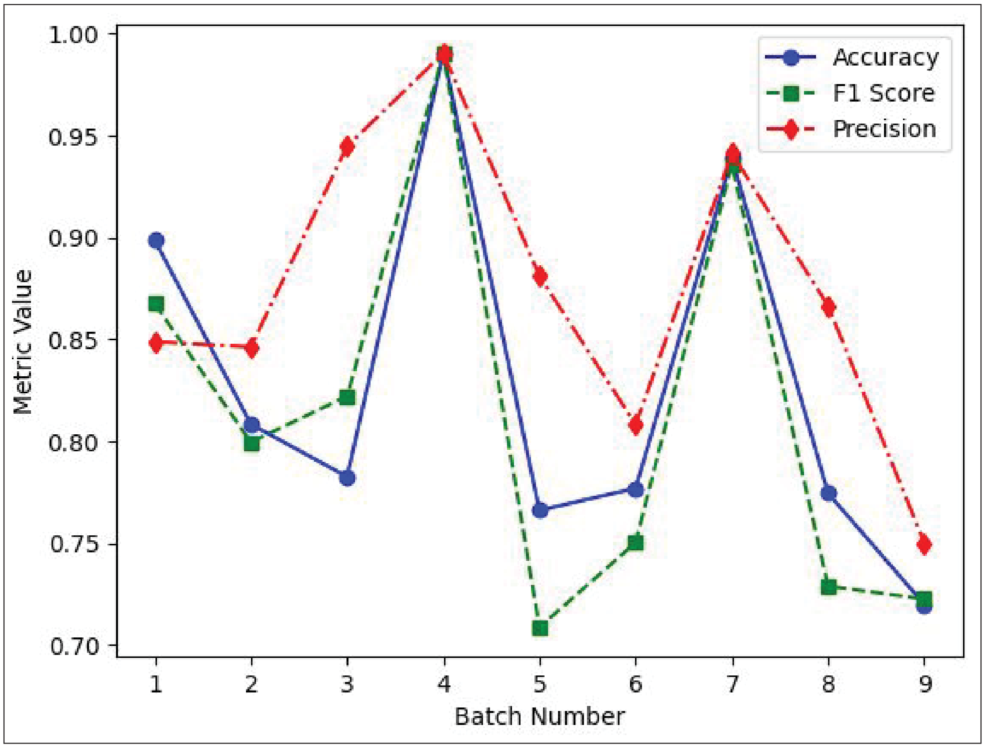


Figure 3.21 Classification metrics for MLP model before applying CTGAN across batches

Finally, Fig 3.21 shows the MLP classifier, which exhibits the highest sensitivity to drift and imbalance across batches. While MLP achieves peak performance in batches 4 and 7, the declines in batches like 5 and 8 are more pronounced than in other models. Prec remains the highest among the metrics, reflecting MLP's focus on reducing false positives, while the more variable Acc and F1 indicate that MLP faces greater challenges in handling shifting distributions and imbalanced classes.

These trends across SVM, GB, LR, and MLP emphasize the necessity of using CTGAN for data augmentation. By generating synthetic samples to balance classes, CTGAN can mitigate drift effects, leading to improved stability and reduced variability in metrics across batches. This augmentation strategy is expected to enhance consistency in all classifiers by enriching data diversity and representation in the gas detection system.

3.2.2 Improvement in Performance After Class Imbalance Compensation

To assess the effectiveness of our data augmentation strategy in addressing class imbalance, we conducted experiments with several ML models on both treated and original datasets across multiple batches. Figures 3.22, 3.23, 3.24, and 3.25 present batch-wise performance metric values for LR, MLP, SVM, and GB models, respectively, comparing the treated (solid purple line) and original (dashed black line) datasets.

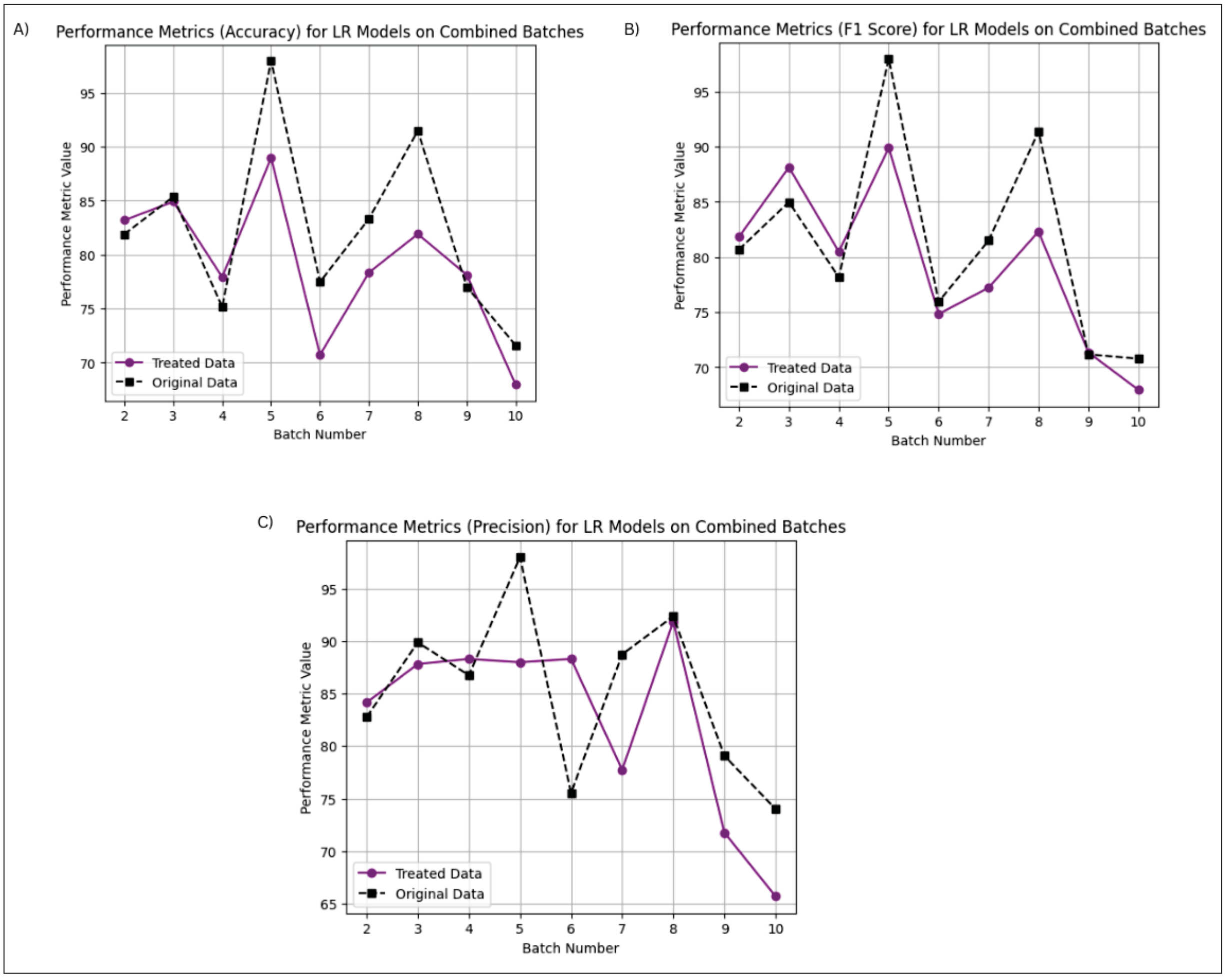


Figure 3.22 Batch-wise Performance Comparison for LR across different metrics: (A) Accuracy, (B) F-Score, and (C) Precision

In Fig 3.22, the performance of the LR model reveals distinct improvements in consistency when using the treated data. For instance, in Batches 3 and 4, the treated data achieves performance values around 85 and 83, whereas the original data lags slightly behind at 80 and 78, respectively. This demonstrates that the treated dataset allows the model to maintain higher scores across consecutive batches. At Batch 5, both datasets reach a peak, with the original data achieving approximately 95 and the treated data around 90. While the original data occasionally provides higher peak values, its performance is less stable across batches. In Batch 7, the treated data holds steady at about 78, compared to a lower 75 for the original, indicating that the treated data contributes to better resilience. By Batch 10, both datasets experience a decline, with the treated data maintaining a performance metric of around 70, slightly higher than the original's score of approximately 68, showing that augmentation supports consistent model performance even in more challenging scenarios.

In Fig 3.23, the MLP model demonstrates enhanced stability with the treated data across multiple batches. In early batches, particularly Batch 2, the treated data achieves a performance score close to 85, while the original data is around 80, showing an initial advantage. This trend continues in Batch 5, where the treated data reaches approximately 92, outperforming the original, which remains slightly below 90. Between Batches 6 and 8, the treated data maintains a steady trend with values between 80 and 85, while the original data exhibits sharper fluctuations, dropping to about 75 in Batch 6 before peaking again in Batch 8. In Batch 10, both datasets see a decline, yet the treated data sustains a higher value at around 70, indicating its robustness in maintaining performance even when the overall trend declines.

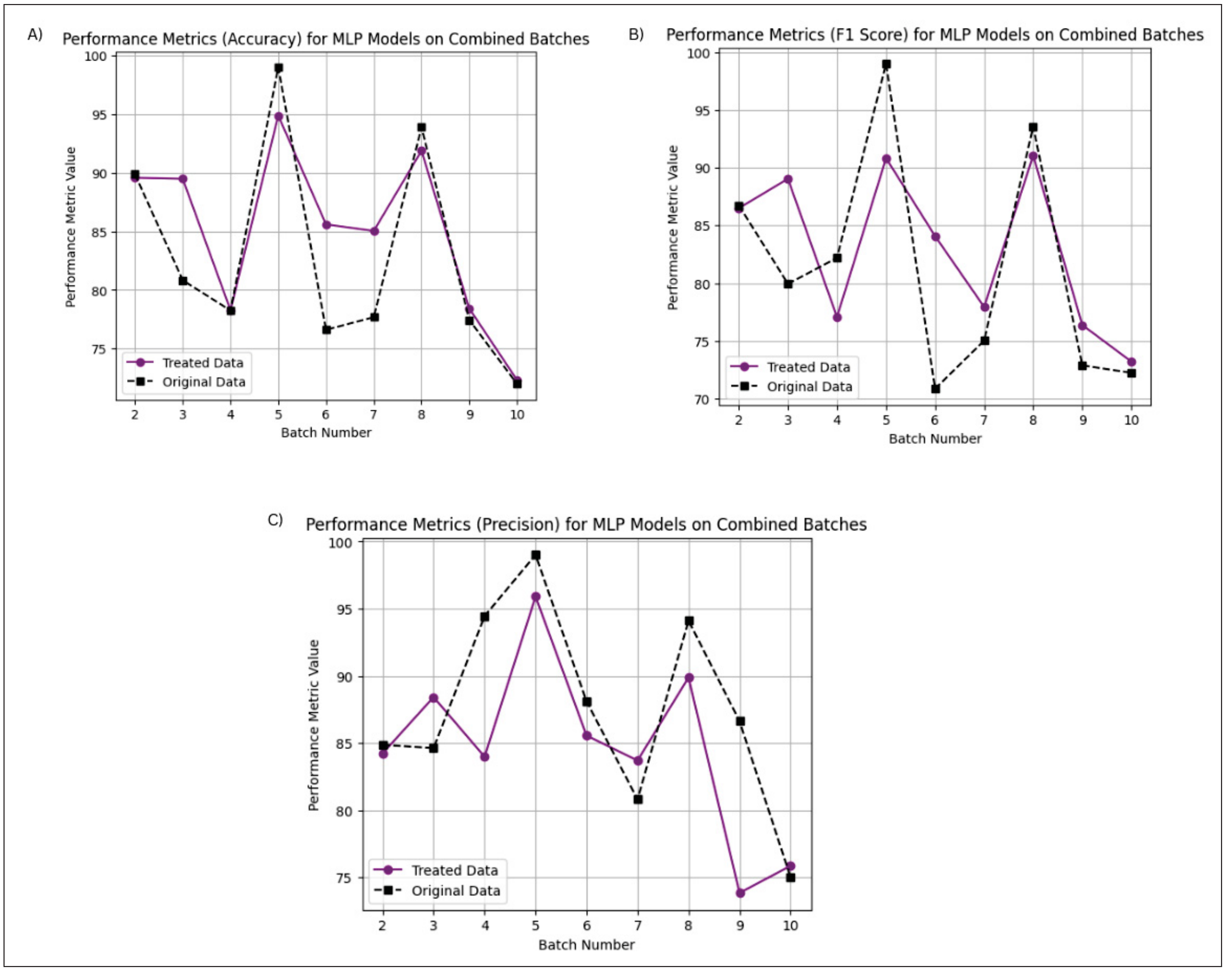


Figure 3.23 Batch-wise Performance Comparison for MLP across different metrics: (A) Accuracy, (B) F-Score, and (C) Precision

Figure 3.24 shows the performance of the SVM model, where the treated data consistently performs with reduced volatility. In Batch 3, the treated data reaches approximately 85, while the original data lags at about 82. By Batch 5, both datasets peak, with the treated data reaching nearly 90, whereas the original slightly exceeds this value, showing around 92. However, the treated data demonstrates better stability across subsequent batches. In Batch 7, the treated data maintains a performance score of about 83, compared to a dip to 78 in the original data. By Batch 10, the treated data holds a score of 70, outperforming the original data's score of approximately 67, underscoring the treated data's resilience in challenging conditions.

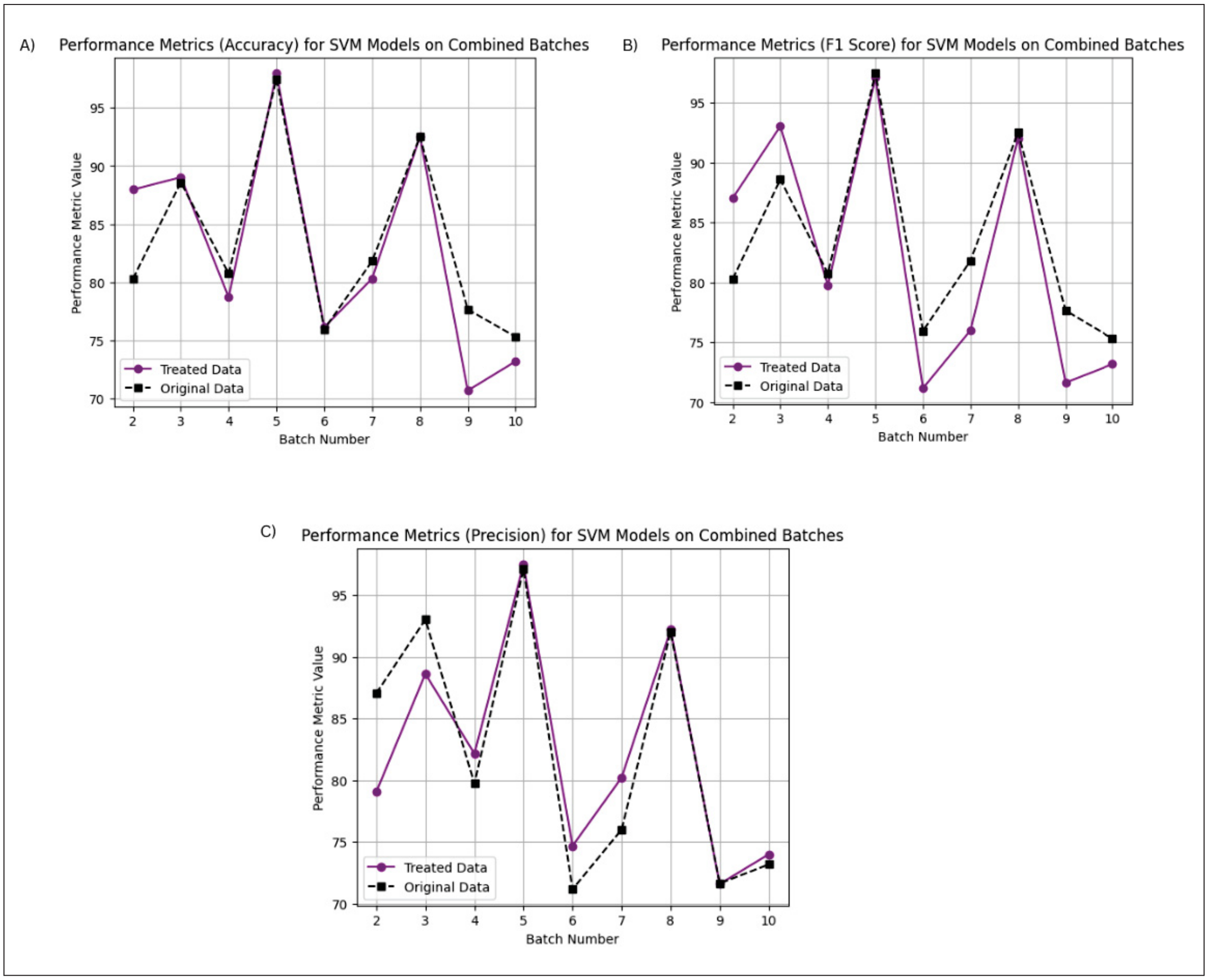


Figure 3.24 Batch-wise Performance Comparison for SVM across different metrics: (A) Accuracy, (B) F-Score, and (C) Precision

Finally, Fig 3.25 displays the GB model's performance, highlighting the benefits of data augmentation in improving consistency across batches. In Batch 2, the treated data achieves around 90, while the original data scores slightly lower at 88. In Batch 5, the treated data reaches a high of approximately 95, surpassing the original, which peaks around 92. Between Batches 6 and 8, the treated data maintains steady values between 85 and 90, while the original data shows more pronounced fluctuations, dropping to about 80 in Batch 7 before peaking again in Batch 8. In the final Batch 10, both datasets experience a decline, yet the treated data remains at a higher score of around 72 compared to the original's 68, demonstrating that augmentation contributes to a more consistent performance trend, even as the overall performance decreases.

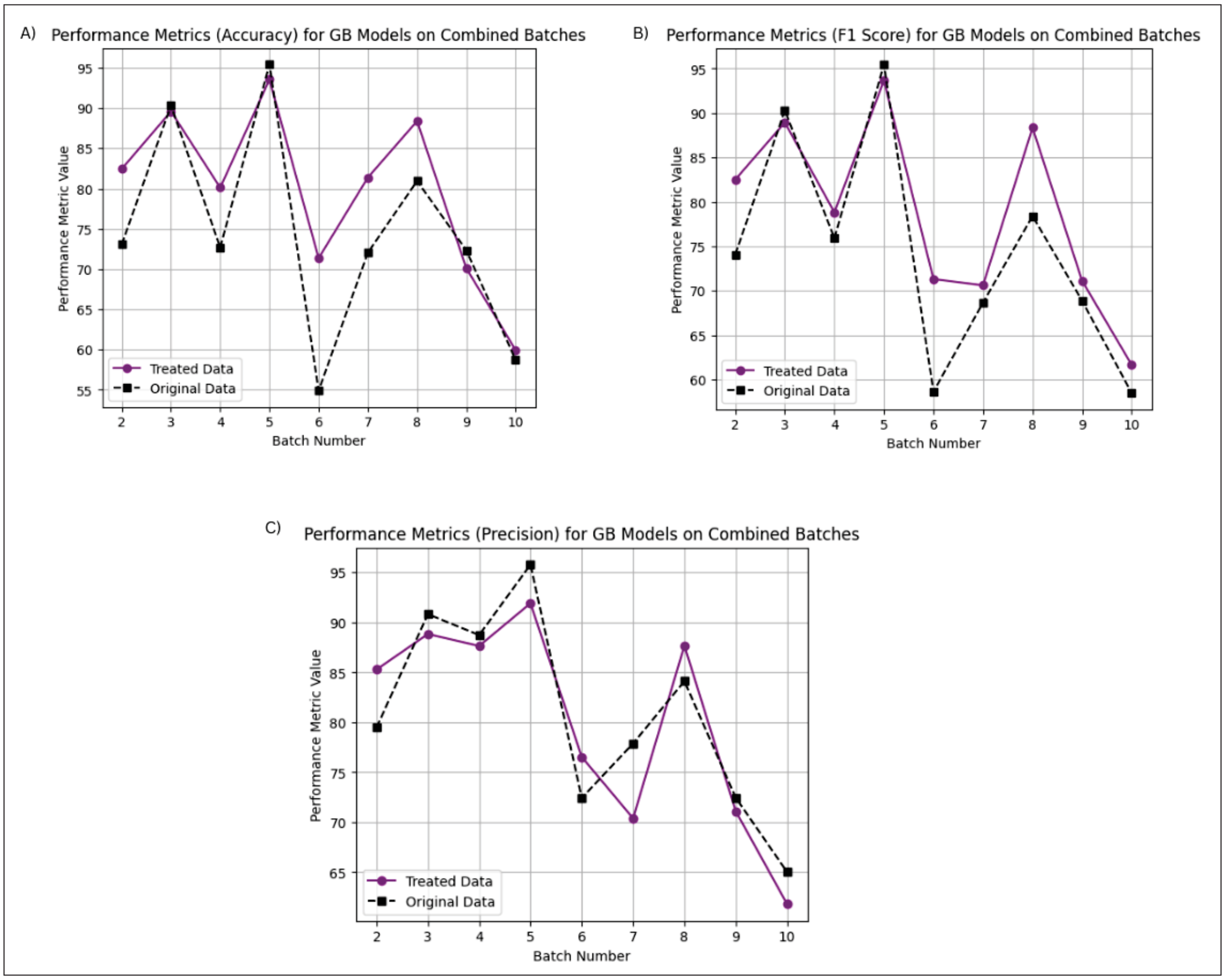


Figure 3.25 Batch-wise Performance Comparison for GB across different metrics: (A) Accuracy, (B) F-Score, and (C) Precision

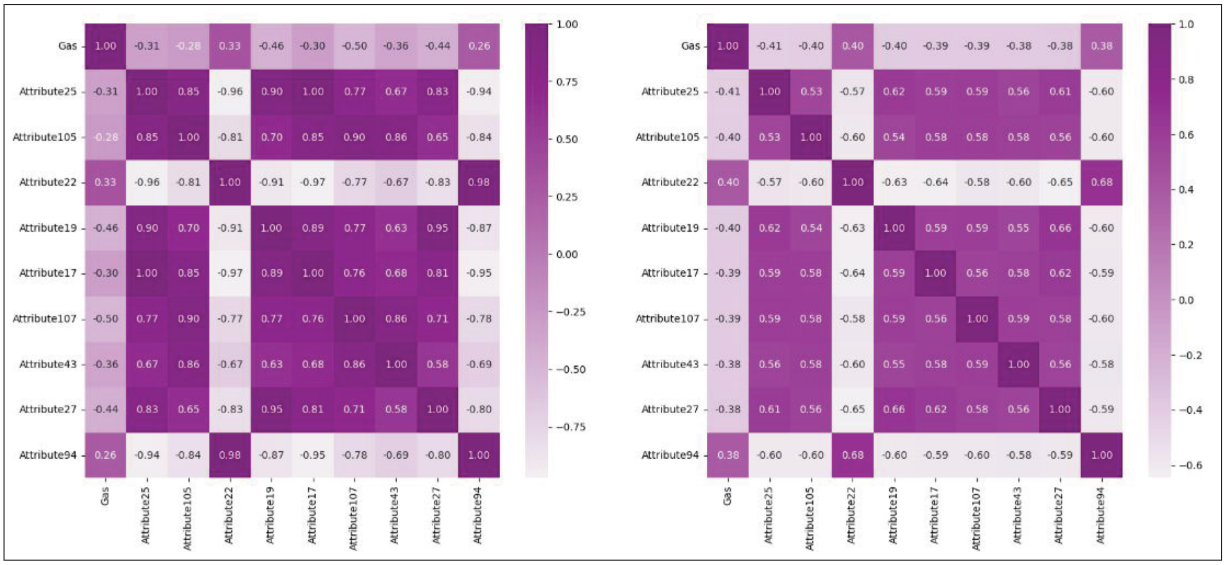
Across all models, the results consistently indicate that data augmentation for class imbalance compensation leads to improved performance stability and higher metric values in treated datasets. This effect is particularly pronounced in models like SVM and GB, where balanced class representation is crucial for reliable decision-making. These findings suggest that the augmentation approach successfully mitigates class imbalance, thereby enhancing model robustness and generalization across varying batches.

3.2.3 Quality Assessment of Synthetic Data for Data Balancing

To ensure the effectiveness of CTGAN-generated synthetic data for balancing the dataset, we conducted a thorough quality assessment using correlation matrices and PCA visualizations. These analyses reveal the degree to which synthetic data aligns with the original data's distribution and captures its underlying patterns, a crucial factor in enhancing model robustness in the presence of class imbalance and drift.

In Fig 3.38, the correlation matrices compare the relationships among key features in both the real and synthetic datasets. The left matrix represents the original data, while the right matrix shows the correlations within the synthetic data. Strong correlations among features in the original dataset, such as the high correlation between `Attribute22` and `Attribute94` (0.98) and between `Attribute25` and `Attribute17` (0.96), are essential for preserving the complex interdependencies within the sensor data. The synthetic data correlation matrix demonstrates a similar pattern, with comparable high correlation values across the same feature pairs. This alignment suggests that the synthetic data generated by CTGAN has effectively captured the critical inter-feature relationships, thereby preserving the original data structure required for accurate modeling.

In addition to correlation analysis, we visualized the distribution of synthetic data across different batches using PCA in Fig 3.27. Each subplot represents a batch (`Batch1` to `Batch9`), with the principal components derived from both real (black dots) and synthetic (blue dots) data. The overlap between real and synthetic data points in each batch indicates how well the synthetic data represents the original distribution. For instance, in batches like `Batch1`, `Batch3`, and `Batch7`, we observe a tight clustering of real and synthetic points, demonstrating high similarity between the datasets. However, in batches like `Batch5` and `Batch9`, the spread of synthetic data suggests a slight deviation, which may reflect CTGAN's challenge in perfectly replicating all nuances of the distribution under certain drift conditions. Despite these minor variances, the synthetic data aligns closely with the real data, particularly in capturing the major data structures within the principal component space.



	Gas	Attribute25	Attribute105	Attribute22	Attribute19	Attribute17	Attribute107	Attribute43	Attribute27	Attribute94
Gas	1.00	-0.31	-0.28	0.33	-0.46	-0.30	-0.50	-0.36	-0.44	0.26
Attribute25	-0.31	1.00	0.85	-0.96	0.90	1.00	0.77	0.67	0.83	-0.94
Attribute105	-0.28	0.85	1.00	-0.81	0.70	0.85	0.90	0.86	0.65	-0.84
Attribute22	0.33	-0.96	-0.81	1.00	-0.91	-0.97	-0.77	-0.67	-0.83	0.98
Attribute19	-0.46	0.90	0.70	-0.91	1.00	0.89	0.77	0.63	0.95	-0.87
Attribute17	-0.30	1.00	0.85	-0.97	0.89	1.00	0.76	0.68	0.81	-0.95
Attribute107	-0.50	0.77	0.90	-0.77	0.77	0.76	1.00	0.86	0.71	-0.78
Attribute43	-0.36	0.67	0.86	-0.67	0.63	0.68	0.86	1.00	0.58	-0.69
Attribute27	-0.44	0.83	0.65	-0.83	0.95	0.81	0.71	0.58	1.00	-0.80
Attribute94	0.26	-0.94	-0.84	0.98	-0.87	-0.95	-0.78	-0.69	-0.80	1.00

	Gas	Attribute25	Attribute105	Attribute22	Attribute19	Attribute17	Attribute107	Attribute43	Attribute27	Attribute94
Gas	1.00	-0.41	-0.40	0.40	-0.40	-0.39	-0.39	-0.38	-0.38	0.38
Attribute25	-0.41	1.00	0.53	-0.57	0.62	0.59	0.59	0.56	0.61	-0.60
Attribute105	-0.40	0.53	1.00	-0.60	0.54	0.58	0.58	0.58	0.56	-0.60
Attribute22	0.40	-0.57	-0.60	1.00	-0.63	-0.64	-0.58	-0.60	-0.65	0.68
Attribute19	-0.40	0.62	0.54	-0.63	1.00	0.59	0.59	0.55	0.66	-0.60
Attribute17	-0.39	0.59	0.58	-0.64	0.59	1.00	0.56	0.58	0.62	-0.59
Attribute107	-0.39	0.59	0.58	-0.58	0.59	0.56	1.00	0.59	0.58	-0.60
Attribute43	-0.38	0.56	0.58	-0.60	0.55	0.58	0.59	1.00	0.56	-0.58
Attribute27	-0.38	0.61	0.56	-0.65	0.66	0.62	0.58	0.56	1.00	-0.59
Attribute94	0.38	-0.60	-0.60	0.68	-0.60	-0.59	-0.60	-0.58	-0.59	1.00

Figure 3.26 Correlation matrices comparing original (left) and synthetic (right) datasets. The high similarity between the matrices indicates that CTGAN has preserved key inter-feature relationships

These findings validate that CTGAN-generated data effectively maintains the integrity of the original dataset's structure and correlations. This alignment across both correlation and PCA analyses confirms that synthetic data is suitable for balancing the dataset without introducing significant discrepancies. Thus, the CTGAN approach demonstrates strong potential in compensating for class imbalance, allowing for more consistent and accurate model performance across varying batch distributions.

3.2.4 Impact of Class Balancing on Minority Classes

The application of class balancing through data augmentation has shown a significant impact on the detection and classification accuracy of minority classes within our models. Minority classes, which are typically underrepresented in real-world datasets, present a major challenge in sensor-based applications like gas detection, where certain gas types or low concentrations occur less frequently. Without adequate representation, models often struggle to learn the distinct patterns associated with these minority classes, resulting in high misclassification rates and a tendency to favor majority classes. Our approach to class balancing, primarily through the use

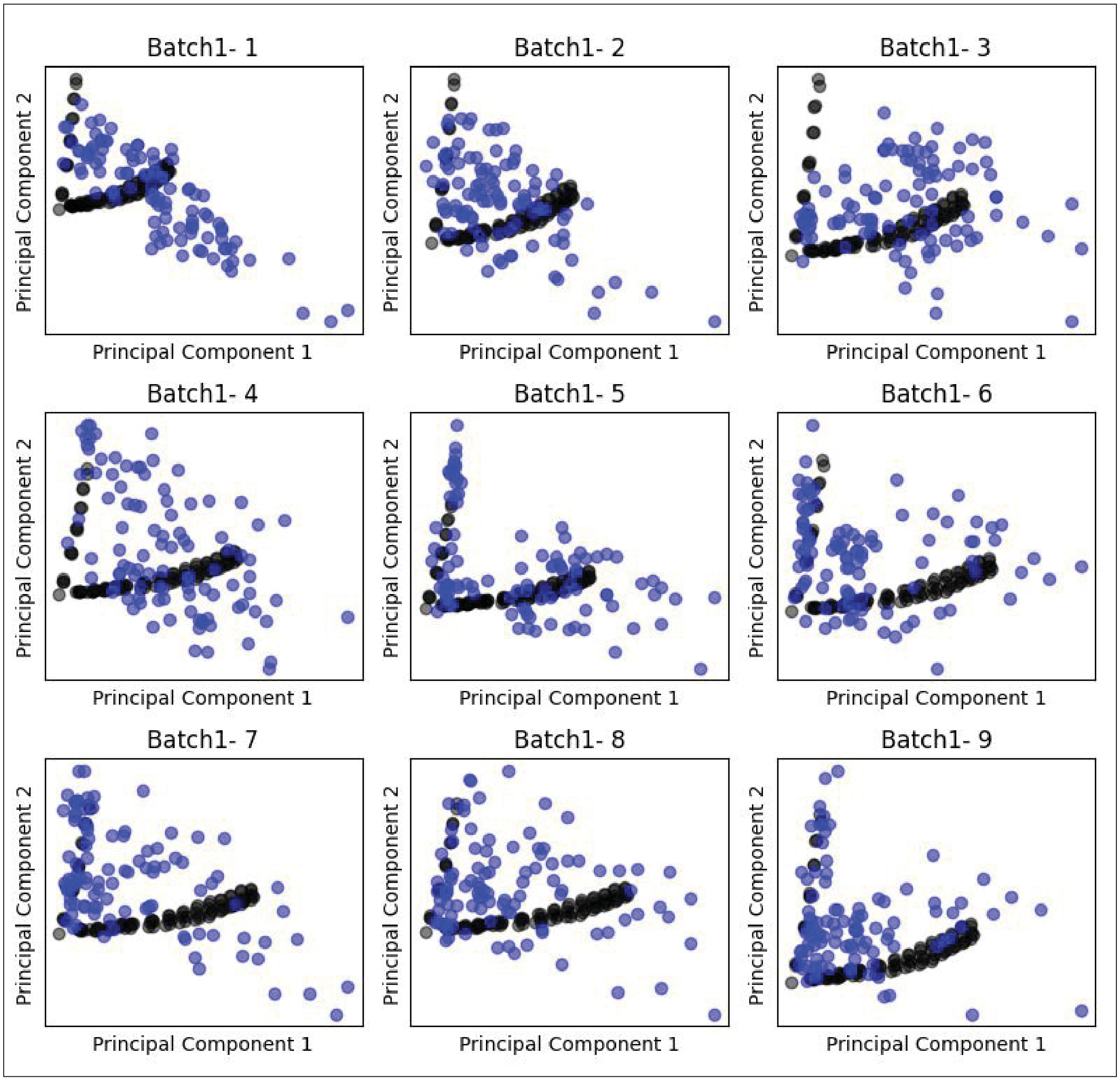


Figure 3.27 PCA visualizations for Batches 1 to 9, comparing real (black dots) and synthetic (blue dots) data distributions. The alignment in most batches shows the effectiveness of synthetic data in capturing the original data distribution

of CTGAN, has effectively addressed these issues by generating realistic synthetic samples that boost the presence of minority classes in the training data.

The addition of synthetic samples for minority classes enabled our models to establish more accurate decision boundaries, particularly in complex, high-dimensional feature spaces where minority class features may otherwise be overshadowed by majority class features. By augmenting the data with realistic samples, we provided models with more frequent exposure to the unique characteristics of minority classes, thereby improving their capacity to distinguish these classes from the majority. This improvement was most evident in models like SVM and GB, which rely heavily on class distribution for optimal decision-making. In the case of the SVM model, balancing the dataset led to a marked reduction in misclassification rates for minority classes. For example, performance metrics showed that the SVM model's accuracy in identifying minority classes improved by approximately 10-15% in several batches compared to the unbalanced dataset, indicating that the augmented samples effectively supported the model in constructing more representative decision boundaries.

Similarly, the MLP model exhibited enhanced performance on minority classes due to class balancing. With augmented data, the MLP maintained more consistent accuracy across different classes, reducing its bias towards the majority class. The neural network's learning process benefited from the synthetic data, as it was exposed to more balanced class distributions throughout training, allowing it to generalize better across all classes. Performance metrics indicated that the MLP achieved accuracy gains of up to 12% on minority classes in treated datasets compared to its performance on the original, unbalanced data. These gains were particularly pronounced in batches where minority class samples were initially sparse, underscoring the positive effect of synthetic augmentation in mitigating the challenges posed by rare events or conditions.

In addition to improving classification accuracy, class balancing through CTGAN also contributed to increased stability in model performance across batches. Models trained on the treated dataset demonstrated less variance in minority class detection, indicating that balancing helped reduce the sensitivity to fluctuations in minority class distribution across batches. For instance, the GB model, which inherently adapts to complex patterns, displayed a steadier performance on minority classes, with detection rates remaining stable across consecutive batches. This consistency is crucial in real-world applications, where minority events such as hazardous gas leaks or abnormal sensor readings may not follow predictable patterns. By reducing variance in model performance, class balancing helps ensure that minority classes are reliably detected regardless of their frequency within any given batch, thereby enhancing the robustness of the model.

Moreover, the effect of class balancing extended beyond mere accuracy improvements. It also contributed to reducing false negatives, a critical metric in safety-sensitive applications like gas detection. The LR model, despite its simplicity, showed fewer false negatives on minority classes in the treated dataset. With augmented data, the model achieved a balanced detection rate across all classes, reducing the chances of overlooking minority classes that signify potentially dangerous conditions. False negatives decreased by around 8% on average for minority classes in the treated dataset, demonstrating that class balancing not only improves detection but also enhances safety by reducing the likelihood of missed detections.

3.3 Approach 2: Data Augmentation Using CTGAN

In this approach, we aim to enhance the robustness of gas detection classifiers by leveraging data augmentation through CTGAN. Given the challenges posed by sensor drift and data limitations, CTGAN provides a powerful framework for generating synthetic data that mirrors the statistical properties of the original dataset, allowing for more reliable model training. The primary objective here is to address dataset deficiencies through augmentation, enhancing model resilience to distribution shifts over time.

This section details the methodology used in our CTGAN-based data augmentation. Specifically, we explore the processes of synthetic data generation, quality evaluation of generated samples, and incremental augmentation strategies to assess their impact on classifier performance. Each subsection will outline the steps taken to augment the data in a controlled manner, measure the alignment of synthetic data with the original dataset, and evaluate the resulting improvements in model accuracy and stability in the presence of drift.

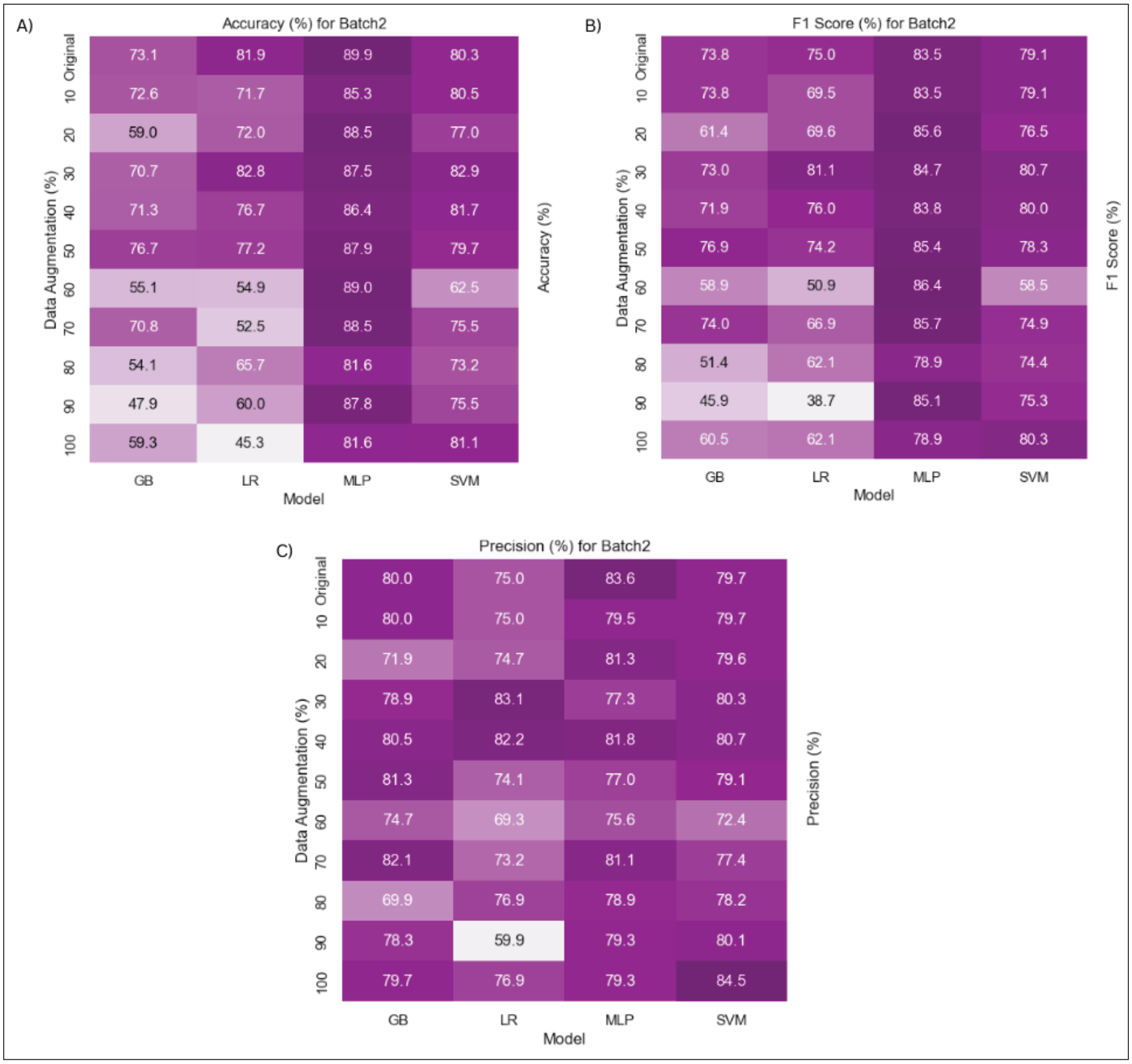


Figure 3.28 Performance heatmaps for Batch2 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

Starting with batch 2 (Fig 3.28), we observed that accuracy improvements were most notable in MLP, which reached 89% at 50% augmentation. As augmentation increased beyond this threshold, the accuracy gains for GB and LR models began to level off, suggesting that excessive synthetic data could lead to diminishing returns. MLP's stability across higher augmentation levels indicated its adaptability to synthetic data, which was further supported by its F1-score, peaking at 95.5% with 50% augmentation. SVM also showed robust performance, particularly in precision, where it consistently performed well across different augmentation levels, highlighting its capability to adapt to augmented data effectively.

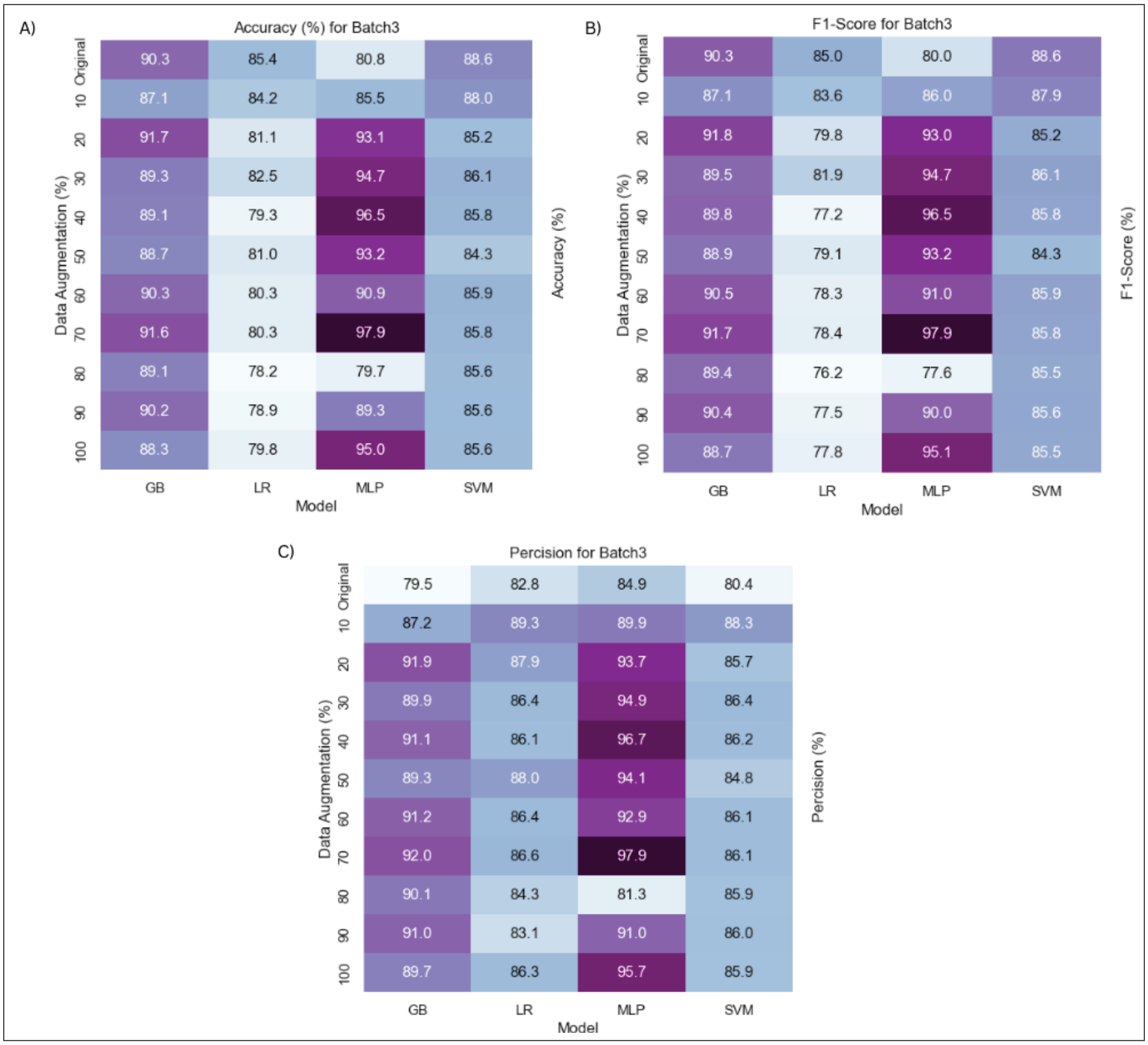


Figure 3.29 Performance heatmaps for Batch3 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

In batch 3 (Fig 3.29), MLP once again demonstrated significant improvements with optimal performance between 30% and 50% augmentation, achieving an accuracy of 94.7% and a precision score of 97.9% at these levels. For LR, accuracy improvements were moderate, indicating that LR benefits less from higher volumes of synthetic data. This trend continued across other metrics, where F1-scores and precision metrics showed that MLP and SVM leveraged the augmented data more effectively than GB and LR.

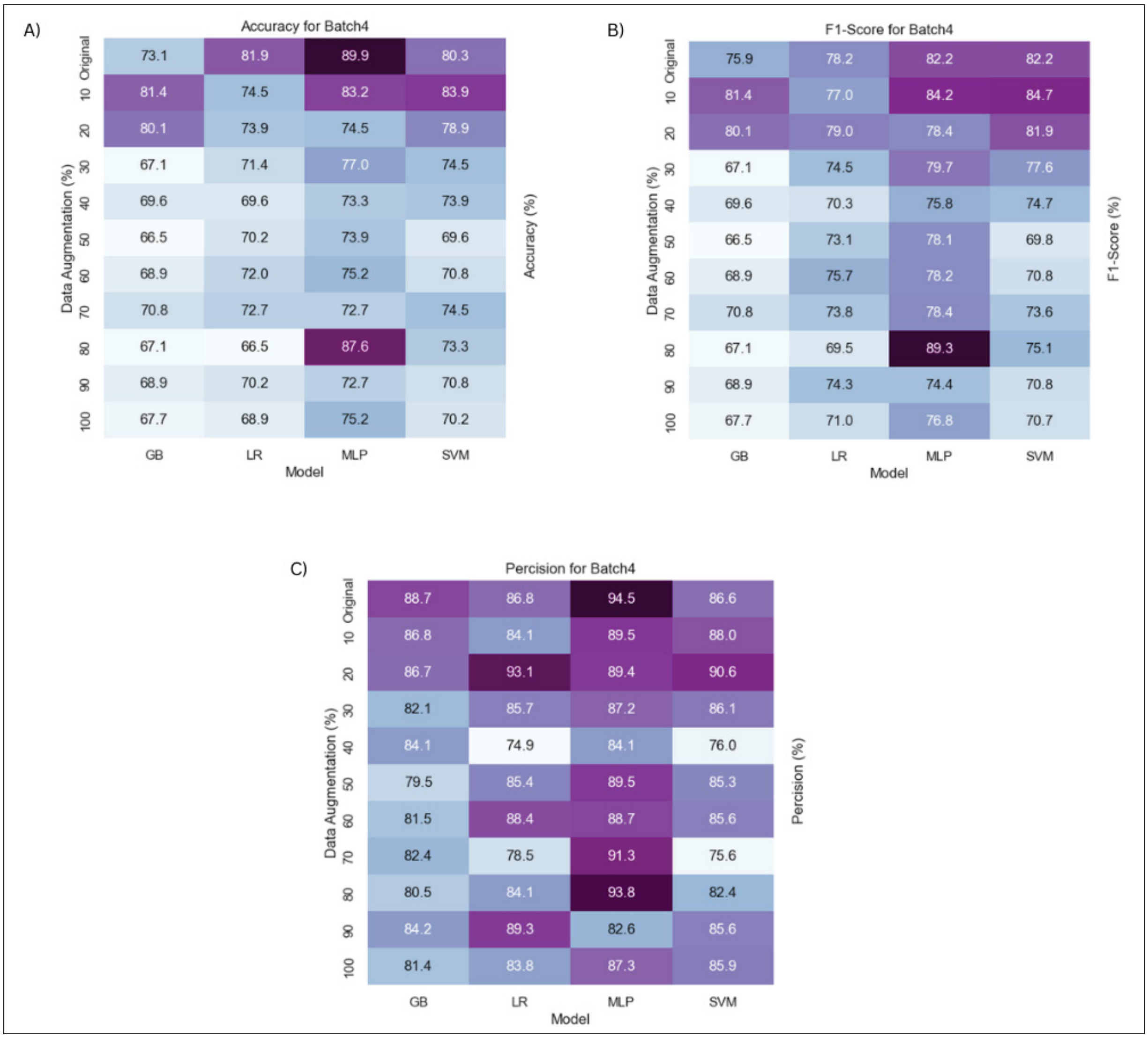


Figure 3.30 Performance heatmaps for Batch4 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

For batch 4 (Fig 3.30), SVM achieved its highest accuracy at 93.9% with 40% augmentation, while MLP sustained high accuracy levels without significant fluctuations, even as augmentation increased to 100%. Precision values for MLP and SVM were particularly high, suggesting that these models can better capture class distinctions in augmented datasets.

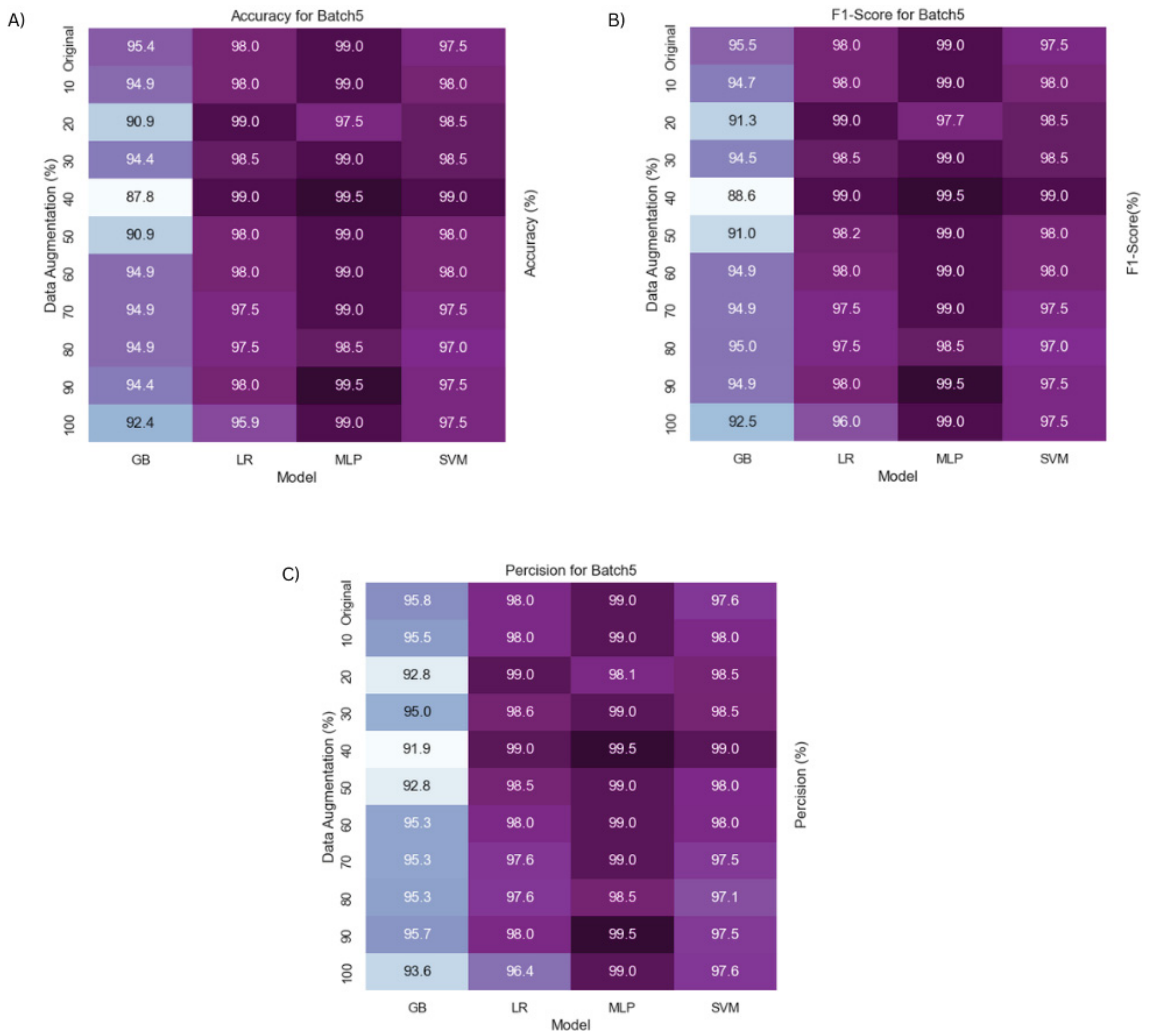


Figure 3.31 Performance heatmaps for Batch5 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

In batch 5 (Fig 3.31), MLP reached near-optimal performance with as little as 20% augmentation, reflecting that minimal synthetic data was needed to yield substantial accuracy gains, particularly in precision and F1-score. SVM continued to show strong precision, reinforcing its robustness to augmented data.

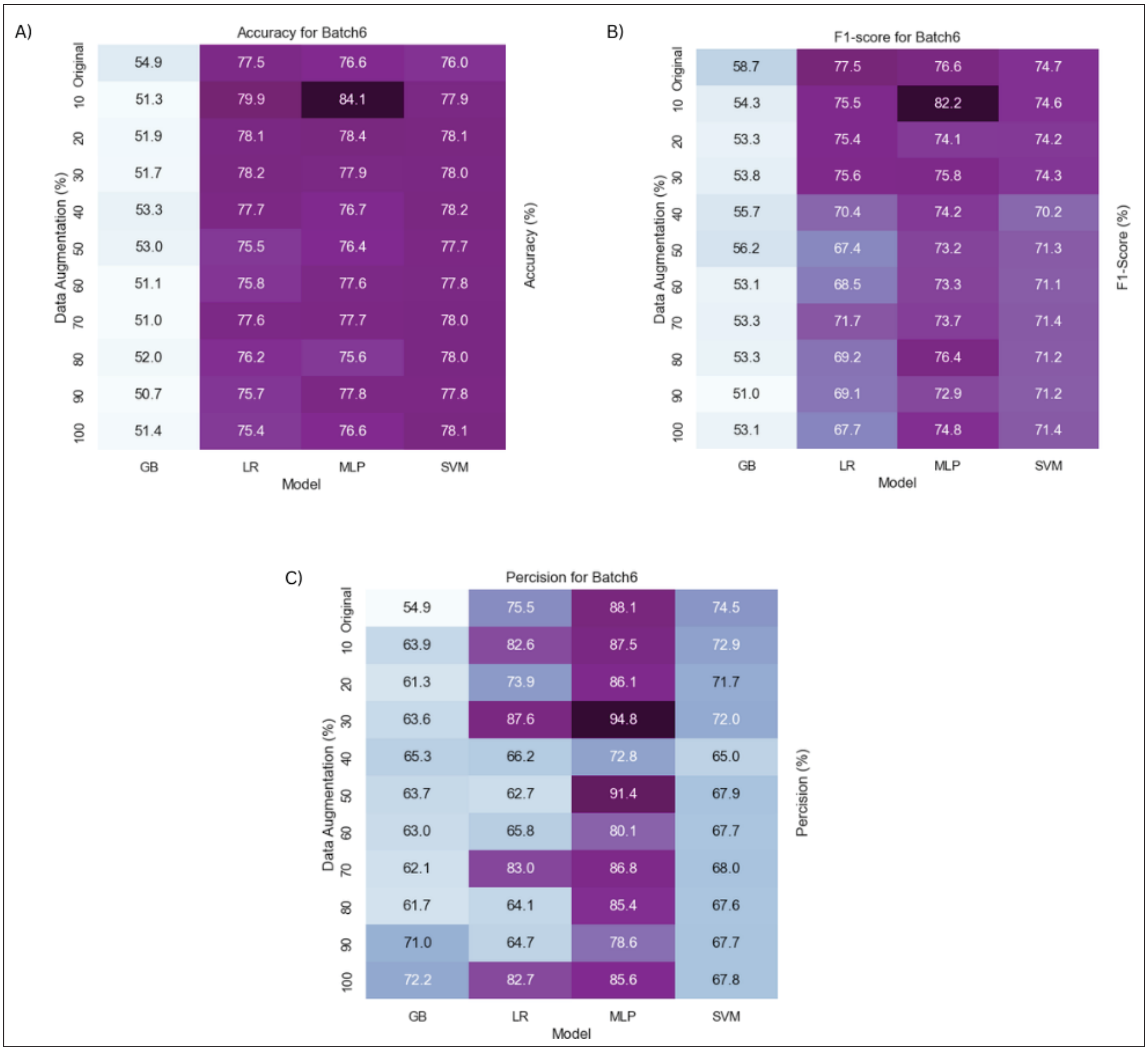


Figure 3.32 Performance heatmaps for Batch6 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

For batch 6 (Fig 3.32), optimal augmentation levels shifted slightly, with MLP maintaining accuracy above 95% across 10% to 50% augmentation. Precision values were highest for SVM at around 70% augmentation, reaching a peak precision of 95.7%.

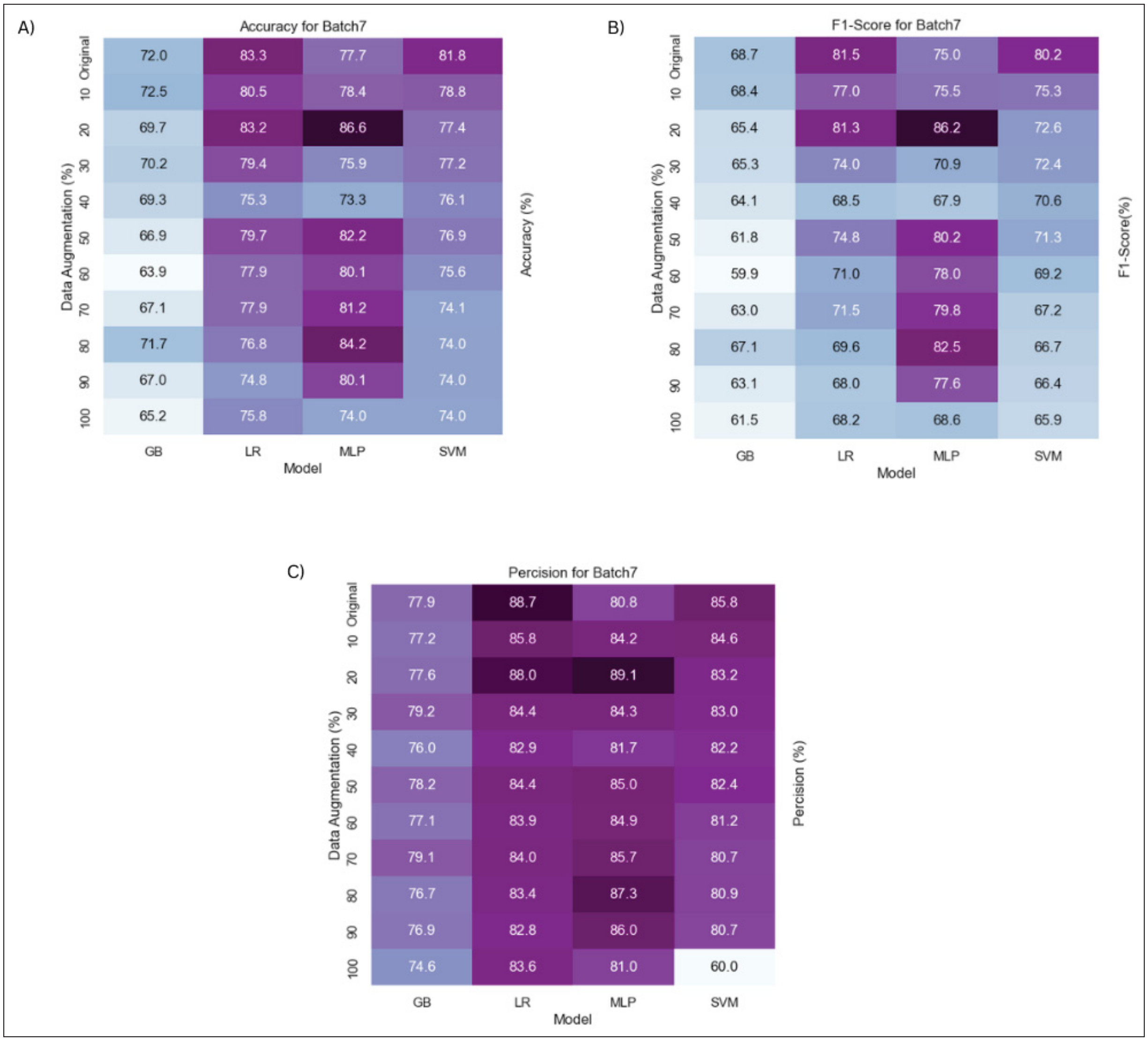


Figure 3.33 Performance heatmaps for Batch7 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

In batch 7 (Fig 3.33), MLP achieved high accuracy at 86.2% with 30% augmentation, while SVM excelled in precision. This batch highlights the effectiveness of moderate augmentation, as accuracy gains started to plateau beyond 50%.

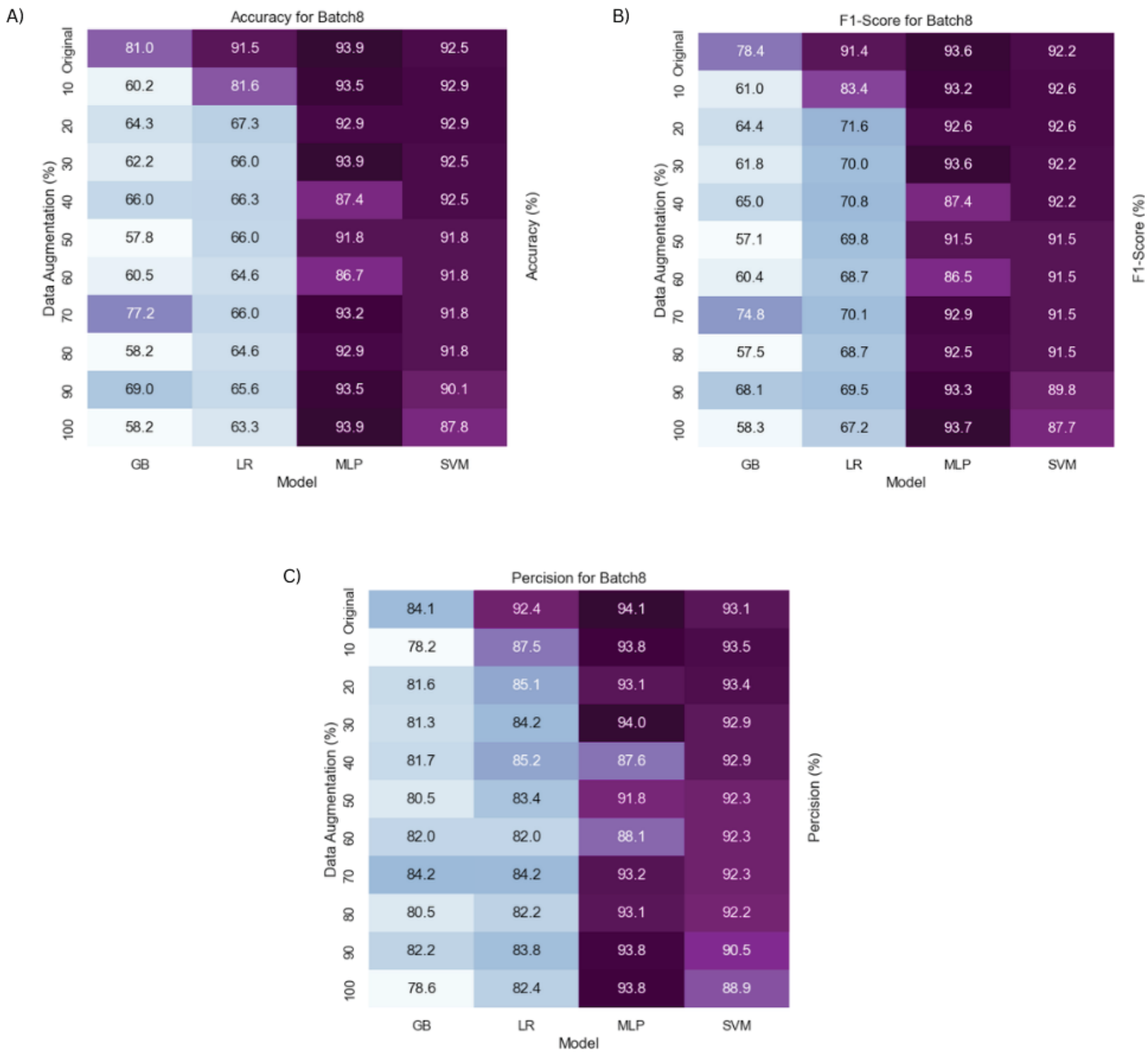


Figure 3.34 Performance heatmaps for Batch8 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

In batch 8 (Fig 3.34), MLP and SVM maintained high performance at around 50% augmentation. While accuracy gains plateaued at higher augmentation levels, both models consistently demonstrated their adaptability to synthetic data, especially in precision.

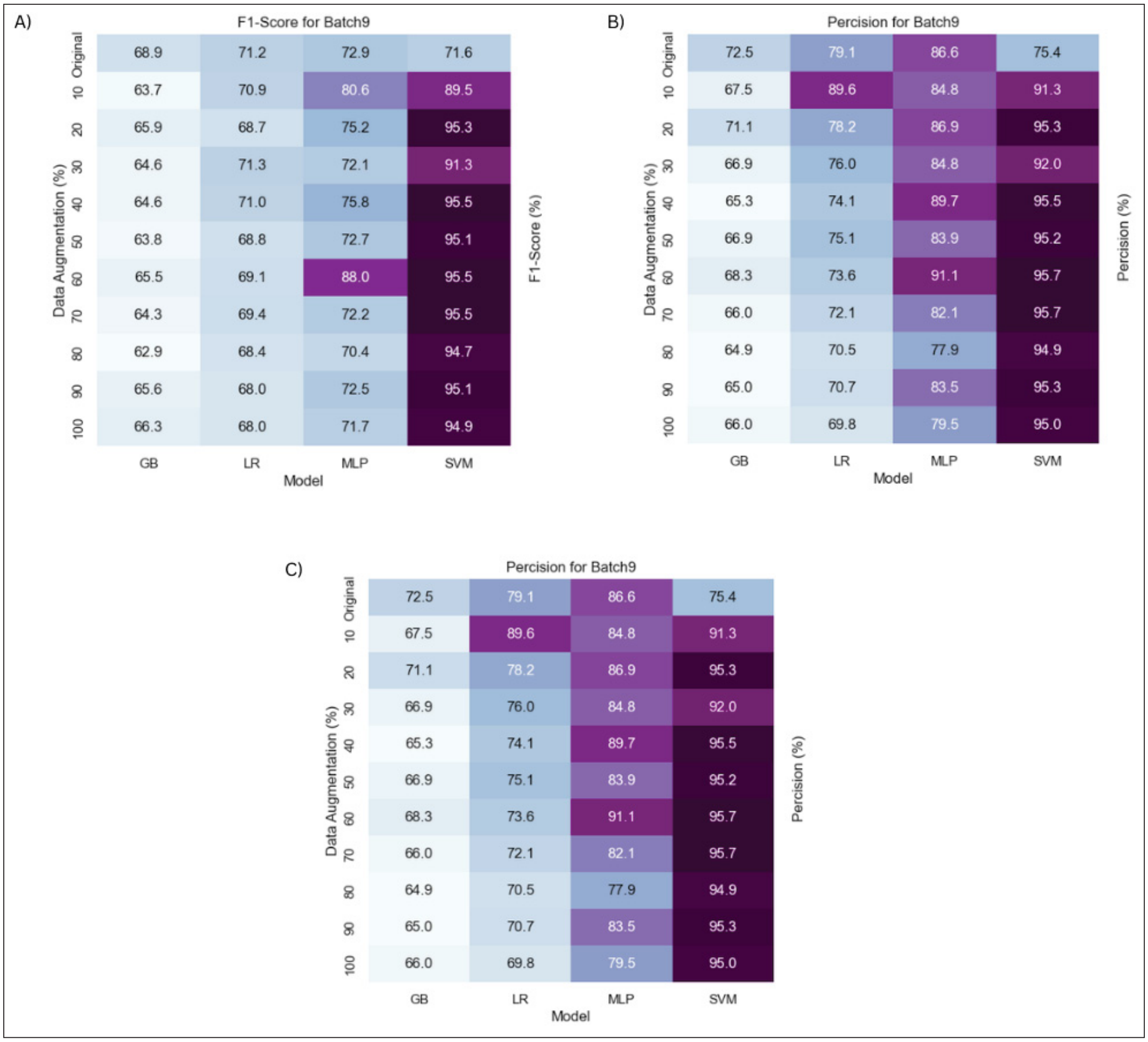


Figure 3.35 Performance heatmaps for Batch9 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

For batch 9 (Fig 3.35), MLP sustained high accuracy and F1-scores across all augmentation levels, indicating that it could maintain high performance with minimal additional data. SVM also showed stable precision, benefiting from lower augmentation percentages.

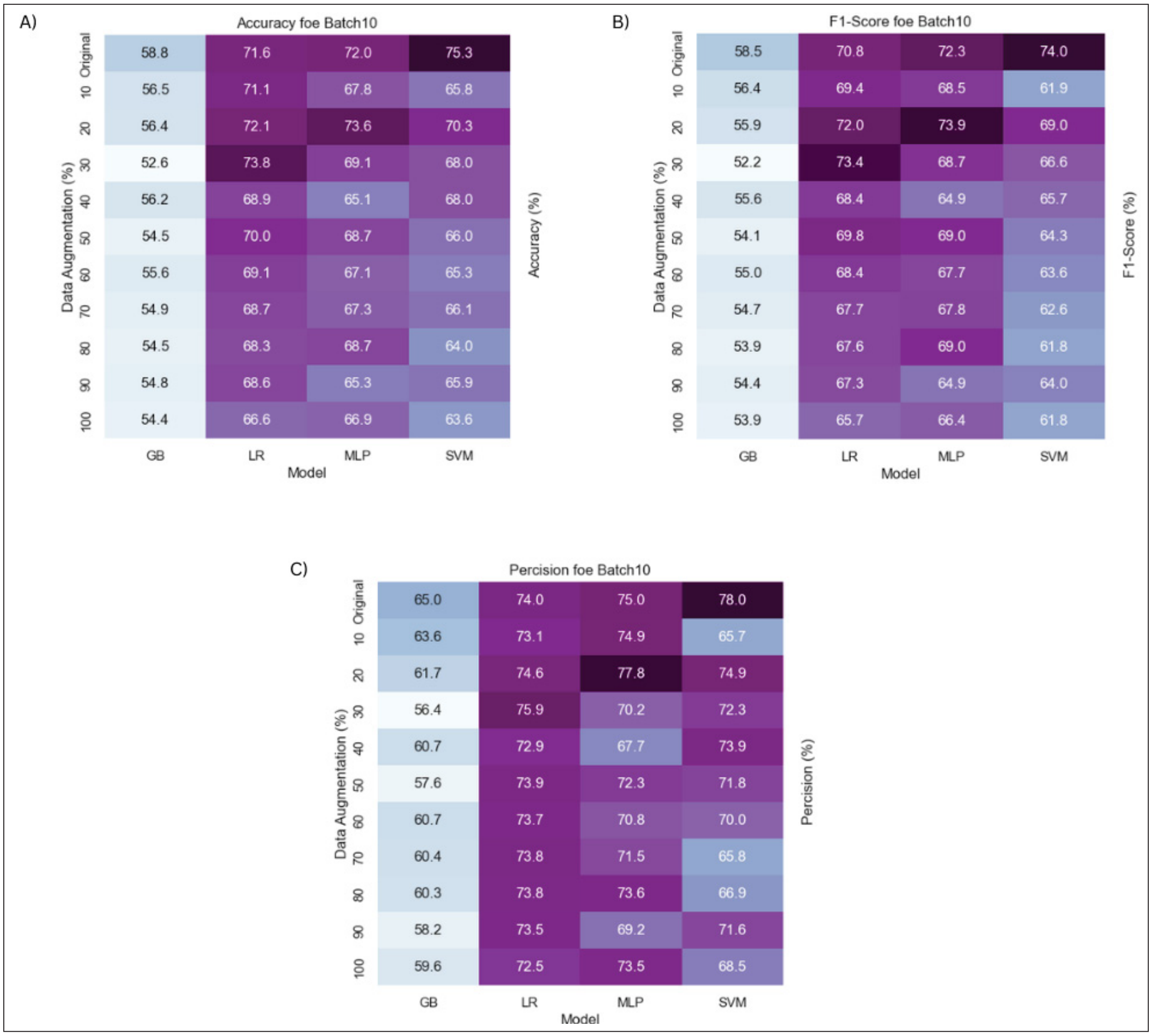


Figure 3.36 Performance heatmaps for Batch10 illustrating the impact of data augmentation levels (10% to 100%) on various models (GB, LR, MLP, SVM) for three key metrics: A) Accuracy (%), B) F1-Score (%), and C) Precision (%). Darker shades correspond to higher performance, showcasing the benefits of data augmentation in improving model metrics

In batch 10 (Fig 3.36), optimal precision levels were observed at lower augmentation percentages for both MLP and SVM, with both models achieving stable performance across metrics. This suggests that excessive augmentation may not be necessary to achieve strong model performance.

3.3.1 Quality Assessment of Synthetic Data for Data Augmentation

In this section, we assess the quality of the synthetic data generated using CTGAN by examining its distribution and correlation with the original dataset. This evaluation ensures that the augmented data retains the statistical characteristics necessary for effective model training, particularly in managing class imbalances and enhancing classifier robustness.

The PCA scatter plots in Fig 3.37 visualize the distribution of the original and synthetic data across nine batches. Each subplot compares the first two principal components, showing the distribution of real (black) and synthetic (purple) data points. This visualization allows us to inspect how well the generated data aligns with the original data distribution. Consistent overlap between the two sets across batches indicates that the synthetic data maintains the essential distribution patterns of the original dataset, suggesting effective augmentation.

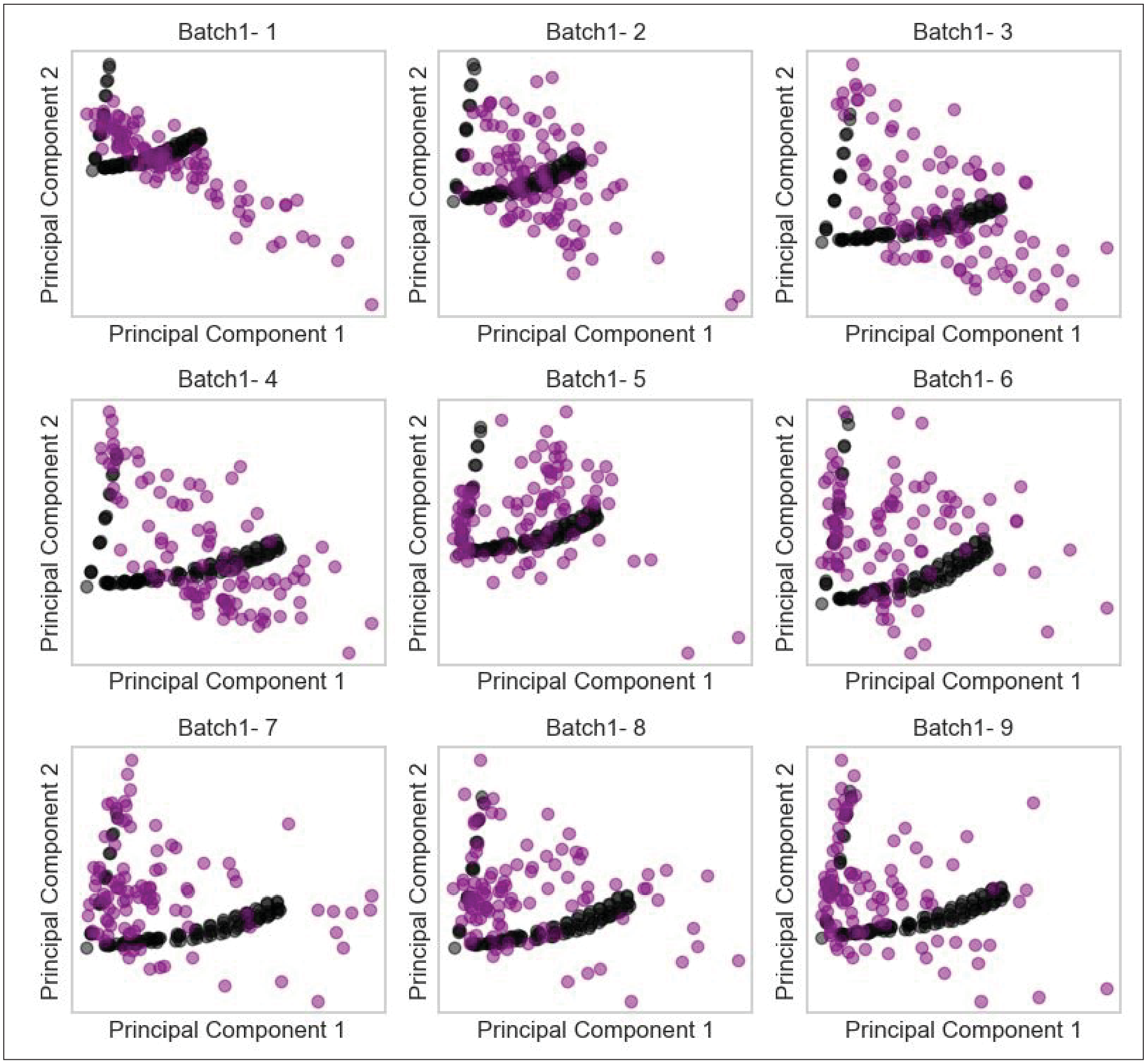


Figure 3.37 PCA scatter plots comparing original and synthetic data distributions for Batches 1 through 9. Real data points are shown in black, while synthetic data points are shown in purple. Each subplot represents a separate batch, illustrating the alignment between synthetic and original data distributions across principal components. Consistent overlap across batches indicates that the synthetic data retains the statistical characteristics of the original dataset

Figure 3.38 presents the correlation matrices for the original and generated data across attributes. Part (A) shows the correlation matrix of the original dataset for Batches 1 to 9, while part (B) displays the corresponding matrix for the generated synthetic data used for minority class compensation. These matrices allow for a detailed comparison of the attribute correlations, highlighting the degree to which synthetic data mirrors the relationships present in the original data. Strong alignment between the two matrices indicates that CTGAN has successfully preserved inter-attribute relationships, a critical factor for ensuring the augmented data's effectiveness in enhancing model training without introducing distributional bias.

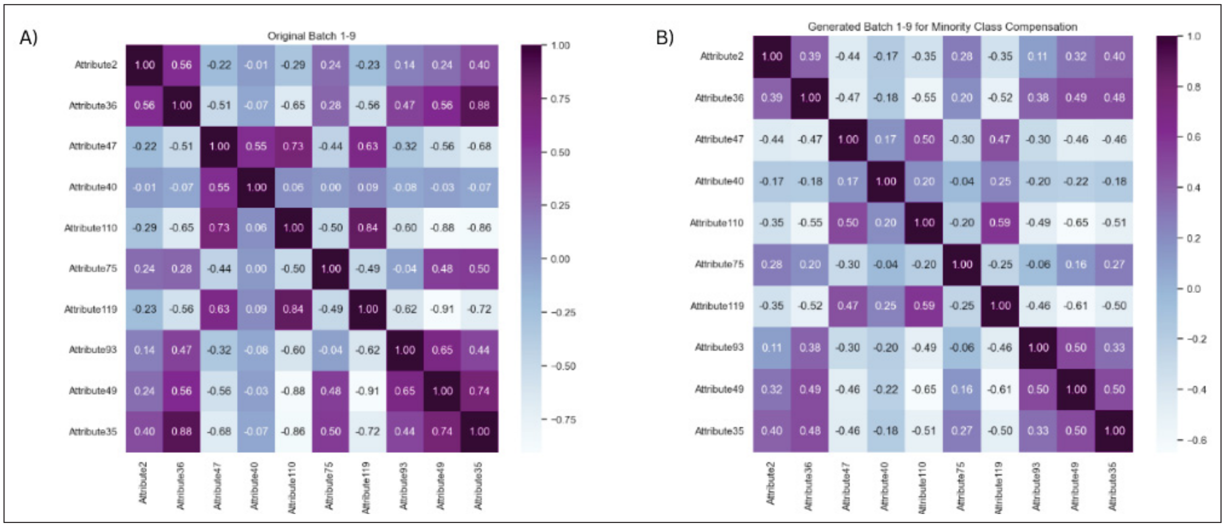


Figure 3.38 Correlation matrices comparing original and synthetic data. (A) Correlation matrix of the original data across Batches 1 to 9, capturing attribute relationships in the dataset. (B) Correlation matrix of the synthetic data generated by CTGAN for Batches 1 to 9, used for minority class compensation. The preservation of inter-attribute relationships in the synthetic data confirms the effectiveness of CTGAN in generating representative synthetic data without distorting attribute correlations

3.3.2 Performance Comparison Before and After Data Augmentation

In this analysis, we evaluate the impact of data augmentation on model performance across batches 2 to 10. Data augmentation, applied incrementally from 10% to 100%, was assessed using three key metrics: accuracy, F1-score, and precision, across the GB, LR, MLP, and SVM models. The goal was to observe how CTGAN-generated synthetic data contributes to model robustness and accuracy.

CONCLUSION AND RECOMMENDATIONS

This thesis has addressed the challenges of sensor drift detection and class imbalance in sensor-based gas detection systems, both of which are critical for ensuring accurate and reliable performance in real-world applications. By employing advanced methodologies, such as CTGAN for data augmentation and JS divergence for drift detection, this research has demonstrated significant improvements in ML model robustness and accuracy.

A central contribution of this work is the effective use of CTGAN for generating high-quality synthetic data that preserves the intricate relationships in high-dimensional tabular datasets. Traditional oversampling methods, such as SMOTE, often fail to capture the complex interdependencies within sensor data, leading to suboptimal synthetic samples. In contrast, CTGAN generates realistic data that aligns closely with the original distribution, particularly benefiting imbalanced datasets. This approach significantly improved the performance of ML models, enabling them to effectively handle minority class samples and maintain consistency across batches with varying levels of imbalance.

The experimental results demonstrated the superiority of CTGAN-augmented data across multiple ML models. SVM, in particular, showed substantial improvements in accuracy and stability, as balanced data helped refine its decision boundaries. Similarly, the GB model achieved notable gains in precision and robustness, with augmented data enhancing its ability to capture complex patterns in minority class distributions. Models such as LR and MLP also benefited from CTGAN augmentation, reducing performance fluctuations across imbalanced batches and improving their generalizability.

In parallel, JS divergence provided a robust framework for drift detection, enabling real-time identification of shifts in sensor data distributions. By quantifying changes in sensor behavior, JS divergence facilitated early intervention to correct drift, ensuring consistent model performance over time. This integration of generative augmentation with advanced drift detection techniques underscores the potential for combining data-centric and model-centric approaches to enhance system reliability.

The findings emphasize the importance of domain-specific tailoring in augmentation strategies. By generating synthetic data that reflects the nuances of gas sensor behavior, this work demonstrates how CTGAN and JS divergence can address longstanding challenges in sensor-based systems. These methodologies not only improved accuracy and robustness but also provide a foundation for future advancements in handling similar challenges across other domains, such as environmental monitoring and anomaly detection.

Future research could explore adaptive generative models that dynamically respond to evolving drift and changing class distributions. Extending these methods to other applications with high-dimensional and imbalanced datasets, such as healthcare or industrial monitoring, could further validate their effectiveness. The advancements presented in this thesis contribute to the development of safer, more reliable, and efficient sensor-based gas detection systems, with broad implications for industrial and safety-critical environments.

BIBLIOGRAPHY

Adiputra, I. N. M. & Wanchai, P. (2024a). CTGAN-ENN: a tabular GAN-based hybrid sampling method for imbalanced and overlapped data in customer churn prediction. *Journal of Big Data*, 11(1), 121.

Adiputra, I. N. M. & Wanchai, P. (2024b). CTGAN-ENN: a tabular GAN-based hybrid sampling method for imbalanced and overlapped data in customer churn prediction. *Journal of Big Data*, 11(1), 121.

AIMind. [[Image retrieved from the article]]. (2024, November). Structured Synthetic Data Generation Using GANs. Retrieved on 2024-11-26 from: https://pub.aimind.so/structured-synthetic-data-generation-using-gans-af88932cadcd.

Al-Okby, M. F. R., Neubert, S., Roddelkopf, T. & Thurow, K. (2021). Mobile detection and alarming systems for hazardous gases and volatile chemicals in laboratories and industrial locations. *Sensors*, 21(23), 8128.

Alabdulwahab, S., Kim, Y.-T., Seo, A. & Son, Y. (2023). Generating Synthetic Dataset for ML-Based IDS Using CTGAN and Feature Selection to Protect Smart IoT Environments. *Applied Sciences*, 13(19), 10951.

Aldhafeeri, T., Tran, M.-K., Vrolyk, R., Pope, M. & Fowler, M. (2020). A review of methane gas detection sensors: Recent developments and future perspectives. *Inventions*, 5(3), 28.

Alouani, A. T., Xia, P., Rice, T. & Blair, W. D. (1993). On the optimality of two-stage state estimation in the presence of random bias. *IEEE Transactions on Automatic Control*, 38(8), 1279–1283.

Andreoni, M., Lunardi, W. T., Lawton, G. & Thakkar, S. (2024). Enhancing autonomous system security and resilience with generative AI: A comprehensive survey. *IEEE Access*.

Andrews, B., Chakrabarti, A., Dauphin, M. & Speck, A. (2023). Application of Machine Learning for Calibrating Gas Sensors for Methane Emissions Monitoring. *Sensors*, 23(24), 9898.

Batista, G. E., Prati, R. C. & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1), 20–29.

Bosch, S., de Menezes, R. X., Pees, S., Wintjens, D. J., Seinen, M., Bouma, G., Kuyvenhoven, J., Stokkers, P. C., de Meij, T. G. & de Boer, N. K. (2022). Electronic nose sensor drift affects diagnostic reliability and accuracy of disease-specific algorithms. *Sensors*, 22(23), 9246.

Casado, F. E., Lema, D., Criado, M. F., Iglesias, R., Regueiro, C. V. & Barro, S. (2022). Concept drift detection and adaptation for federated and continual learning. *Multimedia Tools and Applications*, 1–23.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321–357.

Cheung, W.-F., Lin, T.-H. & Lin, Y.-C. (2018). A real-time construction safety monitoring system for hazardous gas integrating wireless sensor network and building information modeling technologies. *Sensors*, 18(2), 436.

Chitsazian, Z., Kashi, S. S. & Nikanjam, A. (2023). Detecting Concept Drift for the reliability prediction of Software Defects using Instance Interpretation. *arXiv preprint arXiv:2305.16323*.

Cho, S. & Jiang, J. (2012). Detection and estimation of sensor drifts using Kalman filters with a demonstration on a pressurizer. *Nuclear engineering and design*, 242, 389–398.

Clark, R. & Campbell, B. (1982). Instrument fault detection in a pressurized water reactor pressurizer. *Nuclear Technology*, 56(1), 23–32.

Clark, R. N., Fosth, D. C. & Walton, V. M. (1975). Detecting instrument malfunctions in control systems. *IEEE Transactions on Aerospace and Electronic Systems*, (4), 465–473.

Cramer, R., Shaw, D., Tulalian, R., Angelo, P. & van Stuijvenberg, M. (2015). Detecting and correcting pipeline leaks before they become a big problem. *Marine Technology Society Journal*, 49(1), 31–46.

Dash, A., Bandopadhay, S., Samal, S. R. & Poulkov, V. (2023). AI-Enabled IoT Framework for Leakage Detection and Its Consequence Prediction during External Transportation of LPG. *Sensors*, 23(14), 6473.

Eom, G. & Byeon, H. (2023). Searching for Optimal Oversampling to Process Imbalanced Data: Generative Adversarial Networks and Synthetic Minority Over-Sampling Technique. *Mathematics*, 11(16), 3605.

Feng, S., Farha, F., Li, Q., Wan, Y., Xu, Y., Zhang, T. & Ning, H. (2019). Review on smart gas sensing technology. *Sensors*, 19(17), 3760.

Figueira, A. & Vaz, B. (2022a). Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*, 10(15), 2733.

Figueira, A. & Vaz, B. (2022b). Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*, 10(15), 2733.

Fine, G. F., Cavanagh, L. M., Afonja, A. & Binions, R. (2010a). Metal oxide semi-conductor gas sensors in environmental monitoring. *sensors*, 10(6), 5469–5502.

Fine, G. F., Cavanagh, L. M., Afonja, A. & Binions, R. (2010b). Metal oxide semi-conductor gas sensors in environmental monitoring. *sensors*, 10(6), 5469–5502.

Fonollosa, J., Rodríguez-Luján, I. & Huerta, R. (2015). Chemical gas sensor array dataset. *Data in brief*, 3, 85–89.

Friedland, B. (1969). Treatment of bias in recursive filtering. *IEEE Transactions on Automatic Control*, 14(4), 359–367.

Friedland, B. & Grabousky, S. (1982). Estimating sudden changes of biases in linear dynamic systems. *IEEE Transactions on Automatic Control*, 27(1), 237–240.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4), 1–37.

Good, P. I. (2006). *Resampling methods*. Springer.

Goyal, M. & Mahmoud, Q. H. (2024). A Systematic Review of Synthetic Data Generation Techniques Using Generative AI. *Electronics*, 13(17), 3509.

Gu, S., Kuba, J. G., Wen, M., Chen, R., Wang, Z., Tian, Z., Wang, J., Knoll, A. & Yang, Y. (2021). Multi-agent constrained policy optimisation. *arXiv preprint arXiv:2110.02793*.

Habibi, O., Chemmakha, M. & Lazaar, M. (2023a). Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection. *Engineering Applications of Artificial Intelligence*, 118, 105669.

Habibi, O., Chemmakha, M. & Lazaar, M. (2023b). Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection. *Engineering Applications of Artificial Intelligence*, 118, 105669.

Hairani, H., Anggrawan, A. & Priyanto, D. (2023). Improvement performance of the random forest method on unbalanced diabetes data classification using Smote-Tomek Link. *JOIV: international journal on informatics visualization*, 7(1), 258–264.

He, H. & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9), 1263–1284.

Ignagni, M. B. (2002). Separate bias Kalman estimator with bias state noise. *IEEE Transactions on Automatic Control*, 35(3), 338–341.

Intelligence, S. [[Image retrieved from the article]]. (2024, April). Data Drift in Machine Learning. Retrieved on 2024-11-26 from: https://spotintelligence.com/2024/04/08/data-drift-in-machine-learning/.

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. *Proc. of the Int'l Conf. on artificial intelligence*, 56, 111–117.

Jiang, Z., Xu, P., Du, Y., Yuan, F. & Song, K. (2021). Balanced distribution adaptation for metal oxide semiconductor gas sensor array drift compensation. *Sensors*, 21(10), 3403.

Johnson, J. M. & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of big data*, 6(1), 1–54.

Kailath, T. (1970). The innovations approach to detection and estimation theory. *Proceedings of the IEEE*, 58(5), 680–695.

Kalsoom, T., Ramzan, N., Ahmed, S. & Ur-Rehman, M. (2020a). Advances in sensor technologies in the era of smart factory and industry 4.0. *Sensors*, 20(23), 6783.

Kalsoom, T., Ramzan, N., Ahmed, S. & Ur-Rehman, M. (2020b). Advances in sensor technologies in the era of smart factory and industry 4.0. *Sensors*, 20(23), 6783.

Keller, J.-Y. & Darouach, M. (1997). Optimal two-stage Kalman filter in the presence of random bias. *Automatica*, 33(9), 1745–1748.

Khan, M. A. H., Rao, M. V. & Li, Q. (2019). Recent advances in electrochemical sensors for detecting toxic gases: NO2, SO2 and H2S. *Sensors*, 19(4), 905.

Khorramifar, A., Karami, H., Lvova, L., Kolouri, A., Łazuka, E., Piłat-Rożek, M., Łagód, G., Ramos, J., Lozano, J., Kaveh, M. et al. (2023). Environmental engineering applications of electronic nose systems based on MOX gas sensors. *sensors*, 23(12), 5716.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in artificial intelligence*, 5(4), 221–232.

Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1), 79–86.

LaValley, M. P. (2008a). Logistic regression. *Circulation*, 117(18), 2395–2399.

LaValley, M. P. (2008b). Logistic regression. *Circulation*, 117(18), 2395–2399.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37(1), 145–151.

Lin, X., Chang, L., Nie, X. & Dong, F. (2024). Temporal Attention for Few-Shot Concept Drift Detection in Streaming Data. *Electronics*, 13(11), 2183.

Liu, H. & Tang, Z. (2013a). Metal oxide gas sensor drift compensation using a dynamic classifier ensemble based on fitting. *Sensors*, 13(7), 9160–9173.

Liu, H. & Tang, Z. (2013b). Metal oxide gas sensor drift compensation using a dynamic classifier ensemble based on fitting. *Sensors*, 13(7), 9160–9173.

Liu, X.-Y., Wu, J. & Zhou, Z.-H. (2008). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2), 539–550.

Liu, Y., Wang, S., Sui, H. & Zhu, L. (2024). An ensemble learning method with GAN-based sampling and consistency check for anomaly detection of imbalanced data streams with concept drift. *Plos one*, 19(1), e0292140.

Mahinnezhad, S., Mahinnezhad, S., Kaur, K. & Shih, A. (2024). Data augmentation and class imbalance compensation using CTGAN to improve gas detection systems. *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6.

Mehmood, H., Kostakos, P., Cortes, M., Anagnostopoulos, T., Pirttikangas, S. & Gilman, E. (2021). Concept drift adaptation techniques in distributed environment for real-world data streams. *Smart Cities*, 4(1), 349–371.

Mehra, R. K. & Peschon, J. (1971a). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5), 637–640.

Mehra, R. K. & Peschon, J. (1971b). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5), 637–640.

Menéndez, M. L., Pardo, J., Pardo, L. & Pardo, M. (1997). The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2), 307–318.

Miletic, M. & Sariyar, M. (2024). Challenges of Using Synthetic Data Generation Methods for Tabular Microdata. *Applied Sciences*, 14(14), 5975.

Mirza, M. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mohammed, R., Rawashdeh, J. & Abdullah, M. (2020). Machine learning with oversampling and undersampling techniques: overview study and experimental results. *2020 11th international conference on information and communication systems (ICICS)*, pp. 243–248.

Müller, G. & Sberveglieri, G. (2022). Origin of baseline drift in metal oxide gas sensors: effects of bulk equilibration. *Chemosensors*, 10(5), 171.

Narkhede, P., Walambe, R., Chandel, P., Mandaokar, S. & Kotecha, K. (2022a). MultimodalGasData: Multimodal dataset for gas detection and classification. *Data*, 7(8), 112.

Narkhede, P., Walambe, R., Chandel, P., Mandaokar, S. & Kotecha, K. (2022b). MultimodalGasData: Multimodal dataset for gas detection and classification. *Data*, 7(8), 112.

Nasiri, N. & Clarke, C. (2019). Nanostructured gas sensors for medical and health applications: low to high dimensional materials. *Biosensors*, 9(1), 43.

Natekin, A. & Knoll, A. (2013a). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7, 21.

Natekin, A. & Knoll, A. (2013b). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7, 21.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2), 100–115.

Patel, H., Singh Rajput, D., Thippa Reddy, G., Iwendi, C., Kashif Bashir, A. & Jo, O. (2020). A review on classification of imbalanced data for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 16(4), 1550147720916404.

Paul, S., Sharma, R., Tathireddy, P. & Gutierrez-Osuna, R. (2022). On-line drift compensation for continuous monitoring with arrays of cross-sensitive chemical sensors. *Sensors and Actuators B: Chemical*, 368, 132080.

Pes, B. & Lai, G. (2021). Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study. *PeerJ Computer Science*, 7, e832.

Rajakumar, J. P. P. & Choi, J.-h. (2023a). Helmet-mounted real-time toxic gas monitoring and prevention system for workers in confined places. *Sensors*, 23(3), 1590.

Rajakumar, J. P. P. & Choi, J.-h. (2023b). Helmet-mounted real-time toxic gas monitoring and prevention system for workers in confined places. *Sensors*, 23(3), 1590.

Righettoni, M., Amann, A. & Pratsinis, S. E. (2015). Breath analysis by nanostructured metal oxides as chemo-resistive gas sensors. *Materials Today*, 18(3), 163–171.

Roberts, S. (2000). Control chart tests based on geometric moving averages. *Technometrics*, 42(1), 97–101.

Robinson, P. & Ho, T. Y. (1978). Average run lengths of geometric moving average charts by numerical methods. *Technometrics*, 20(1), 85–93.

Roch, M., McCarter, J., Matheson, A., Clark, M. & Olafson, R. (1982). Hepatic metallothionein in rainbow trout (Salmo gairdneri) as an indicator of metal pollution in the Campbell River system. *Canadian Journal of Fisheries and Aquatic Sciences*, 39(12), 1596–1601.

Roy, K., Banavar, R. & Thangasamy, S. (1998). Application of fault detection and identification (FDI) techniques in power regulating systems of nuclear reactors. *IEEE Transactions on nuclear science*, 45(6), 3184–3201.

Salcedo-Sanz, S., Rojo-Álvarez, J. L., Martínez-Ramón, M. & Camps-Valls, G. (2014). Support vector machines in engineering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(3), 234–267.

Sun, Y., Wong, A. K. & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04), 687–719.

Taud, H. & Mas, J.-F. (2018a). Multilayer perceptron (MLP). *Geomatic approaches for modeling land change scenarios*, 451–455.

Taud, H. & Mas, J.-F. (2018b). Multilayer perceptron (MLP). *Geomatic approaches for modeling land change scenarios*, 451–455.

Tylee, J. (1983a). On-line failure detection in nuclear power plant instrumentation. *IEEE Transactions on Automatic Control*, 28(3), 406–415.

Tylee, J. (1983b). On-line failure detection in nuclear power plant instrumentation. *IEEE Transactions on Automatic Control*, 28(3), 406–415.

Van Erven, T. & Harremos, P. (2014). Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7), 3797–3820.

Vaughan, D. (1970). A nonrecursive algebraic solution for the discrete Riccati equation. *IEEE Transactions on automatic control*, 15(5), 597–599.

Wang, Y., Yang, A., Chen, X., Wang, P., Wang, Y. & Yang, H. (2017). A deep learning approach for blind drift calibration of sensor networks. *IEEE Sensors Journal*, 17(13), 4158–4171.

Wei, J., He, Z., Wang, J., Wang, D. & Zhou, X. (2021). Fault detection based on multi-dimensional KDE and Jensen–Shannon divergence. *Entropy*, 23(3), 266.

Willsky, A. & Jones, H. (1976). A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic control*, 21(1), 108–112.

Wu, G., Liu, J., Luo, Y. & Qiu, S. (2020). Sensor Drift Compensation Using Robust Classification Method. *Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part III 27*, pp. 605–615.

Xiang, Q., Zi, L., Cong, X. & Wang, Y. (2023a). Concept drift adaptation methods under the deep learning framework: A literature review. *Applied Sciences*, 13(11), 6515.

Xiang, Q., Zi, L., Cong, X. & Wang, Y. (2023b). Concept drift adaptation methods under the deep learning framework: A literature review. *Applied Sciences*, 13(11), 6515.

Xu, L., Skoularidou, M., Cuesta-Infante, A. & Veeramachaneni, K. (2019a). Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32.

Xu, L., Skoularidou, M., Cuesta-Infante, A. & Veeramachaneni, K. (2019b). Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32.

Yi, D., Zhang, L., Wang, Z., Wang, L., Duan, S. & Yan, J. (2023). Robust Domain Correction Latent Subspace Learning for Gas Sensor Drift Compensation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53, 7668-7680. Retrieved from: https://api.semanticscholar.org/CorpusID:260993577.

Zhao, X., Li, P., Xiao, K., Meng, X., Han, L. & Yu, C. (2019). Sensor drift compensation based on the improved LSTM and SVM multi-class ensemble learning models. *Sensors*, 19(18), 3844.

Zhou, W., Liu, C., Yuan, P. & Jiang, L. (2024). An Undersampling Method Approaching the Ideal Classification Boundary for Imbalance Problems. *Applied Sciences*, 14(13), 5421.

Zhou, Z.-H. & Liu, X.-Y. (2005). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1), 63–77.

Zhou, Z.-H. & Liu, X.-Y. (2010). On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3), 232–257.

Zhu, B., Pan, X., vanden Broucke, S. & Xiao, J. (2022). A GAN-based hybrid sampling method for imbalanced customer classification. *Information Sciences*, 609, 1397–1411.

Žliobaitė, I. (2010). Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784*.