Dissertation for the Degree of Master of Engineering in Information System Security

Optimizing Model Convergence and Accuracy in Time Series Anomaly Detection using Synthetic Data Integration and Rolling Window Stratified Cross-Validation

Jahidul Arafat
Tirtha Mondal
A.S.M. Abdur Rob
Faruq Ahamed

Department of Information and Communication Technology
Faculty of Science and Technology
Bangladesh University of Professionals

July, 2024

Optimizing Model Convergence and Accuracy in Time Series Anomaly Detection using Synthetic Data Integration and Rolling Window Stratified Cross-Validation

by

Jahidul Arafat
Tirtha Mondal
A.S.M. Abdur Rob
Faruq Ahamed

Supervised by

Prof. Md. Ahsan Habib, Ph.D.

Submitted to the Department of Information and Communication Technology of the Faculty of Science and Technology in Bangladesh University of Professionals for partial fulfillment of the requirements of the degree of Master of Engineering

This dissertation titled "Optimizing Model Convergence and Accuracy in Time Series Anomaly Detection using Syntheic Data Integration and Rolling Window Stratified Cross-Validation" submitted by Jahidul Arafat, Student ID: 2254911003, Tirtha Mondal, Student ID: 2254911005, A.S.M Abdur Rob, Student ID: 2254911002 and Faruq Ahmed, Roll No: 2254911043, has been accepted as satisfactory in partial fulfilment of the requirement for the degree of Master of Engineering in Information System Security on 12 July, 2024.

Dissertation Committee

1. (signature)
   Dr. Md. Ahsan Habib, PhD — Chairman
   Professor, Dept of Computer Science & Engineering
   Green University of Bangladesh

2. (signature)
   Dr. Masud Rana, PhD
   Member
   Chairman, Dept of ICT — (Ex-official)
   Bangladesh University of Professionals

3. (signature)
   Afrina Khatun
   Assistant Professor, Dept of ICT — Member
   Bangladesh University of Professionals
   Member

4. (signature)
   Dr. Fazlul Hasan Siddiqui, PhD — External Member
   Professor, Dept of Computer Science and Engineering
   Dhaka University of Engineering and Technology

Declaration

We declare that this dissertation titled "Optimizing Model Convergence and Accuracy in Time Series Anomaly Detection using Syntheic Data Integration and Rolling Window Stratified Cross-Validation" and the works presented in it are our own. We confirm that:

- The full part of the work is done during master's research study in Bangladesh University of Professionals, Bangladesh.
- This dissertation or any portion of it has not been previously submitted for a degree or other qualification at this University of any other institution.
- We have cited the published works of others with appropriate references.
- This research work is done entirely by us and our contribution and enhancements from other works are clearly stated.

Singned: _____

Candidates:

Jahidul Arafat

Tirtha Mondal

A.S.M. Abdur Rob

Faruq Ahamed

Countersigned: _____

Supervisor: Dr. Md. Ahsan Habib

Abstract

Detecting anomalies in time series data is critical for applications across various sectors, including but not limited to financial fraud detection, predictive maintenance, healthcare monitoring, and cybersecurity. The latest learning-based anomaly detection models have shown significant improvements over their statistical-based counterparts in inference accuracy and model interoperability, owing to their task-specific training on extensive multivariate corpora. However, their current performance requires enhancement for scalable practical deployment. Particularly when handling high-dimensional, high-velocity, and high-volume complex time series data; present challenges like data sparsity, imbalance, variability, and temporal inconsistencies of real data during model training. The absence of window focused stratefied cross-validation in model training for time series data could further lead to suboptimal model performance and unreliable evaluations. To address these challenges and trade-off between model convergence and generalizability, this paper proposes a comprehensive framework namely irsRSk, leveraging high quality synthetic data generated through pTimeGAN along with real datasets and Rolling Window Time Series Stratified k-Fold Cross-Validation during model training and validation. Empirical experiments were conducted on 6 models for generalization, 5 synthetic data generation and cross-valdiaiton techniques with 3 opensource datasets. These are rigorously tested and cross-validated against proposed irsRSk. The proposed framework achieves substantial improvements in computational accuracy and lower prediction errors across EM Acc and CA, which are further cross-validated with auxiliary PRF1.

Acknowledgement

This research would not have been possible without the support and guidance of many individuals and institutions. We would like to express my deepest gratitude to our advisor, Prof. Dr. Ahsan Habib, for his invaluable insights, constant encouragement, and unwavering support throughout this journey. His expertise and mentorship were instrumental in shaping the direction and quality of this work. We are also grateful to my committee members, Dr. Masud Rana and Dr. Fazlul Hasan Siddiqui, for their constructive feedback and guidance, which greatly enhanced the depth and rigor of this research.

Special thanks go to our colleagues and fellow Security and Machine Learning researchers at bKash for their collaborative spirit and intellectual discussions that enriched my understanding of the subject. We would also like to acknowledge the support from Prof. Dr. Mahbubur Rahman, Dept of CSE, MIST for providing the necessary resources and a conducive research environment.

Lastly, We are profoundly thankful to our family and friends for their unwavering support, patience, and encouragement, which gave us the strength to persevere through the challenges of this research. This work is dedicated to all who believed in us and supported us along the way.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we overview the key motivation to develop irsRSk backbone, which uses synthetic data and Rolling Window Time Series Stratified k-Fold Cross-Validation to improve model convergence, generalization, and prediction accuracy while reducing false positives and negatives.

1.1 Introduction

The field of anomaly detection has witnessed substantial advancements with contemporary machine learning models, aimed at enhancing detection accuracy and efficiency. With the rapid growth of industries, time series data has become increasingly prevalent, characterized by high volume, velocity, and intricate patterns. Advanced models, such as deep learning and Generative Adversarial Networks (GANs), excel in identifying anomalies across various domains. The rise of big data has driven the need for sophisticated methods to detect complex behaviors and trends within time series data [1],[2]. Consequently, as industries generate vast amounts of data, the demand for effective anomaly detection models has surged.

Broadly, time series anomaly detection models fall into three clusters [2-4]: statistical, machine learning, and deep learning models. Statistical models like control charts rely on predefined thresholds, offering simplicity but struggling with high-dimensional data. Machine learning models such as support vector machines (SVMs) and decision trees learn from labeled data, providing more flexibility and up to 20% improved accuracy over statistical methods in sectors like healthcare [3]. However, they require extensive labeled datasets and face challenges with imbalanced data.

Deep learning models, including neural networks and GANs, leverage complex architectures to capture intricate data patterns, achieving up to 30% higher accuracy in detecting network intrusions. Despite their superior performance, deep learning models demand significant computational resources and large training datasets [2],[4]. Thus, while statistical models are simple, machine learning models offer better adaptability, and deep learning models excel in capturing complex patterns, though at a higher computational cost.

1.1.1 Exponential Growth of Time Series Data

Since the volume and complexity of time series data are growing alongside the large adoption of sensors, networking, security, and monitoring devices across various sectors such as finance, healthcare, manufacturing, and cybersecurity, the challenges associated with anomaly detection have intensified. Fast growth in edge devices, with worldwide connected devices projected to reach 30.9 billion by 2025 from 13.8 billion units in 2021, driven by IoT, 5G, and cloud adoption, contributes significantly to this trend. Businesses generate high volumes, velocity, and variety of time series data that cannot be effectively monitored via traditional dashboards. Additionally, the global anomaly detection market size is projected to reach US$ 3835.9 million by 2032, up from US$ 2077.9 million in 2021, at a CAGR of 8.7% during 2022-2032 [5]. This exponential growth in time series data and the corresponding increase in anomalies present significant challenges in terms of data storage, processing, and real-time analysis, necessitating the development of more sophisticated detection models.

1.1.2 Evolution from Statistical to Advanced Models for Time Series Anomaly Detection

In the early stages of development, statistical-based solutions were the primary approach to anomaly detection, relying heavily on predefined thresholds and statistical assumptions to identify outliers. Methods such as Shewhart control charts, introduced in the 1920s, and hypothesis testing, like the t-test and chi-square test, were effective for simple datasets with low dimensionality and well-understood distributions [4]. However, as data complexity increased, these traditional methods began to show limitations. Control charts struggled with high-dimensional data and non-linear relationships, and the assumption of normality in many statistical tests often did not hold in real-world datasets, leading to inaccurate anomaly detection. According to [3], statistical models' reliance on fixed thresholds and assumptions made them inadequate for dynamic environments where data patterns evolve over time. In sectors like manufacturing, simple statistical process control methods failed to predict equipment failures due to their inability to adapt to changing conditions and complex interdependencies between variables. Similarly, in the banking sector, statistical models struggled with the evolving tactics of fraudsters, as highlighted by [7], who noted that traditional methods missed sophisticated fraud patterns not fitting predefined assumptions. These limitations of statistical-based solutions in handling high-dimensional and complex data led researchers to explore alternative approaches, paving the way for machine learning and deep learning techniques that offered more accurate and adaptive solutions for anomaly detection in modern, dynamic environments.

1.1.3 Challenges within State-of-the-art works

To address the limitations of statistical models in handling complex time series data, recent research has focused on neural networks, Generative AI (GANs), and deep learning. These advanced techniques offer greater flexibility and accuracy by learning complex data patterns without relying on predefined thresholds. Neural networks, such as RNNs and LSTMs, have significantly improved fraud detection rates in financial transactions, while GANs generate synthetic data to augment real datasets, addressing data scarcity and imbalance [6]. Deep learning models, particularly CNNs and autoencoders, have been effective in cybersecurity, achieving up to 35% higher accuracy in detecting network intrusions. However, these models require large labeled datasets, extensive computational resources, and face challenges like overfitting and lack of interpretability. Despite these issues, the shift towards neural networks, GANs, and deep learning represents a substantial advancement in anomaly detection, offering more accurate and adaptive solutions compared to traditional methods.

In the state-of-the-art work, time series data analysis can be categorized into three major areas: predictive modeling, pattern recognition, and anomaly detection. Predictive modeling involves forecasting future data points based on historical data, a technique extensively applied in finance for stock price prediction and in weather forecasting. Studies [11],[14],[15],[38],[40] have shown that LSTM networks significantly outperform traditional ARIMA models in predictive accuracy. Pattern recognition, on the other hand, focuses on identifying recurring patterns within the data. This is crucial in domains such as healthcare, where recognizing patterns in patient vitals can indicate the onset of diseases. According to a study [13], convolutional neural networks (CNNs) have demonstrated superior performance in identifying complex patterns compared to traditional methods. Anomaly detection aims at identifying outliers that deviate from expected behavior, which is critical in

areas like cybersecurity and fraud detection. Researchers have correlated these categories with earlier works, highlighting significant advancements through the application of machine learning and deep learning techniques. Deep learning models, in particular, have provided more effective and efficient solutions across these categories by leveraging their ability to model complex, non-linear relationships within the data. For instance, a study [13] found that autoencoders achieved up to 30% higher anomaly detection rates in network security applications compared to traditional statistical methods. Overall, the integration of advanced techniques in predictive modeling, pattern recognition, and anomaly detection has significantly contributed to the progress in the field of time series data analysis, offering more robust and accurate models.

However, these strategies invite specific problems [10-15]. Neural networks and GANs face high computational costs, extensive data requirements, and are prone to overfitting, particularly with imbalanced datasets. This overfitting leads to poor generalization, resulting in inaccurate predictions and increased false positives and negatives. Additionally, the interpretability of these complex models is limited, posing challenges in critical sectors like healthcare and finance. Real-time data processing requirements further exacerbate these issues. Real data often contains noise, missing values, and inconsistencies, hindering model convergence and generalization. To address these challenges, synthetic data generation is used to create balanced datasets that maintain the statistical properties of real data in a controlled environment. Therefore, this dissertation focuses on leveraging synthetic data and Rolling Window Stratified k-Fold Cross-Validation (TSK-Fold) to optimize model convergence and generalization, enhancing prediction accuracy while reducing false positives and negatives. This approach aims to develop more interpretable, efficient, and robust anomaly detection frameworks.

The rest of this chapter is organized as follows. Section 1.2 provides an overview of synthetic data generation strategies. Next, we review several time series anomaly detection models, focusing on different challenges in Section 1.3. The effectiveness of using synthetic data alongside real data in model training, highlighting its benefits in improving model performance for time series anomaly deteciton in section 1.4. Section 1.5 delves into the motivation behind this research, aiming to alleviate issues faced by these models in practical applications for time series anomaly detection. The problem statement, and the solution methodologies proposed in this research is mentioned in Section 1.6. The contributions of these research efforts are discussed in Section 1.7. Finally, Section 1.8 outlines the overall organization of the dissertation, providing a roadmap for the subsequent chapters.

1.2 Overview of Synthetic Data Generation Strategies

Synthetic data generation has emerged as a pivotal technique in enhancing the performance of machine learning models, particularly for anomaly detection in time series data. Various strategies have been developed to generate synthetic data [11-17], each with its own strengths and challenges. Generative Adversarial Networks (GANs) and their variants, such as Conditional GANs (CGANs) , Wasserstein GANs (WGANs), WGAN with Gradient Penalty (WGAN-GP), and DRAGAN, have been widely studied for their ability to produce high-fidelity synthetic data. GANs operate by training two neural networks—the generator and the discriminator—in tandem, leading to the creation of synthetic data that closely mimics the distribution of real data[12]. Advanced models like TimeGAN and DoppelGANger have specifically targeted the generation of sequential and time series data, capturing temporal dynamics effectively [15]. Additionally, CTGAN and Gaussian Mixture Models offer alternative approaches for generating high-quality synthetic data suitable for complex

datasets [16]. Despite their promise, these methods face challenges such as the computational intensity of generating synthetic data that maintains intricate dependencies and patterns present in real-world time series data, and the need for rigorous validation to ensure the synthetic data retains the statistical properties and variability of the original data. Comparative studies [13],[16],[17] have shown that models trained with synthetic data often achieve better generalization and robustness, addressing the limitations of training solely on real data. As these techniques evolve, they hold the potential to significantly improve the capabilities of anomaly detection models in diverse and complex time series environments.

1.3 Overview of Time Series Anomaly Detection Models

Time series anomaly detection encompasses a range of models, each designed to address specific challenges inherent in time series data. Among statistical models, ARIMA (AutoRegressive Integrated Moving Average) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) are widely used. ARIMA is particularly effective for short-term forecasting and capturing linear relationships but struggles with non-linear patterns and requires stationary data [13]. GARCH models excel at modeling and predicting volatility, making them useful in financial applications, though they can be complex to implement and necessitate careful parameter tuning [9].

In the realm of deep learning, LSTM-Autoencoders and GANs have shown significant promise. LSTM-Autoencoders combine the strengths of LSTM networks, which are adept at handling sequential data, with the dimensionality reduction capabilities of autoencoders, making them effective for detecting anomalies in highly variable time series data. However, they are computationally intensive and can overfit without sufficient data [11]. GANs, originally developed for generating synthetic data,

have been adapted for anomaly detection by training the generator to produce normal data and the discriminator to identify anomalies. This approach excels in capturing complex data distributions but also faces challenges like mode collapse and high computational requirements [13].

Generative AI models such as Isolation Forest and Prophet also play a crucial role. Isolation Forest is a robust, tree-based model that isolates anomalies by randomly partitioning the data, offering efficiency and scalability, although it can struggle with high-dimensional data [12]. Prophet, developed by Facebook, is designed for forecasting time series data with strong seasonal effects and missing data, providing simplicity and interpretability but may not perform well with non-linear anomalies [13]. Each of these models presents unique strengths and limitations, underscoring the importance of selecting the appropriate technique based on the specific characteristics and requirements of the time series data in question.

1.4 Effectiveness of Synthetic Data with Real Data in Model Training

The integration of synthetic data with real data in model training has shown significant promise in enhancing model performance and addressing challenges inherent in real-world datasets. Synthetic data can mitigate issues such as data scarcity, imbalance, and noise, which are prevalent in real datasets and can hinder the training and generalization of machine learning models. Studies [13],[4],[17],[5] have demonstrated that Generative Adversarial Networks (GANs) can create high-quality synthetic data that closely resembles real data, thereby enriching the training process. Moreover, combining synthetic data with real data can improve model robustness, as it provides

a more diverse training set that captures a wider range of data variations. This is particularly important in anomaly detection, where anomalies are often rare and varied. The use of synthetic data has also been shown to reduce the risk of overfitting, as models trained on augmented datasets can better generalize to unseen data. However, challenges remain, such as ensuring the quality and relevance of synthetic data, and the computational cost associated with generating large volumes of synthetic data [17]. Despite these challenges, the effectiveness of integrating synthetic data with real data in model training is evident, as it enhances prediction accuracy and model reliability, ultimately leading to more robust anomaly detection systems.

1.5 Motivation for This Research

Earlier studies [11],[13],[17] justify that integrating synthetic data with real data is critical for model training. If real data alone is used, the model is likely to face challenges such as data scarcity, imbalance, and noise, which can significantly impact model convergence, prediction accuracy, and generalization. Real data often contains inconsistencies and missing values, leading to higher prediction errors and poor model performance. Synthetic data helps alleviate these issues by providing a more balanced and comprehensive training set, which in turn improves the robustness and reliability of the model.

In the literature [11-17], we have found a good number of synthetic data generation strategies. Notable methods include Generative Adversarial Networks (GANs), Conditional GANs (CGANs), Wasserstein GANs (WGANs), WGAN with Gradient Penalty (WGAN-GP), DRAGAN, CameraGAN, CWGAN-GP, CTGAN, Gaussian Mixture Models, Sequential Data Generation, TimeGAN, and

DoppelGANger. Each of these techniques offers unique advantages in generating high-quality synthetic data that closely resembles real data, enhancing the model's training process. However, these models also face significant challenges. GANs and their variants, while powerful, often suffer from issues like mode collapse and require extensive computational resources. Ensuring the quality and relevance of the synthetic data generated is another major challenge. The generated data must accurately reflect the statistical properties and variability of the real data to be effective. Moreover, the integration of synthetic data with real data must be done carefully to avoid introducing biases or inconsistencies that could degrade model performance.

Therefore, in time series anomaly detection, it is essential to integrate synthetic data with real data through rigorous cross-validation to minimize these issues. This necessity has driven us to generate synthetic data using TimeGAN, which is designed specifically for time series data. By integrating this synthetic data with real data using Time Series Rolling Window k-Fold Cross-Validation, we can preserve trends, seasonality, and temporal dependencies in the data. This approach guides the model towards better training convergence, improved prediction accuracy, and enhanced generalization, ultimately leading to more robust and effective anomaly detection frameworks.

1.6 Problem Description and Solution Methods

In this section, we provide a brief overview of the problem addressed in this dissertation along with the solution methodologies to address it.

1.6.1 Problem Description

Earlier research in time series anomaly detection has made significant strides using statistical models, machine learning models, and deep learning models. However, these approaches have notable shortcomings. Statistical models like ARIMA and GARCH struggle with high-dimensional and non-linear data patterns, limiting their effectiveness in complex environments. Machine learning models such as SVMs and decision trees require extensive labeled datasets and face challenges with imbalanced data, while deep learning models like LSTM-Autoencoders and GANs, although powerful, are prone to overfitting and demand high computational resources. These limitations affect the model convergence and generalizability of time series anomaly detection models, often leading to a trade-off between computational accuracy and convergence loss, resulting in higher false positive and false negative prediction errors.

In time series data, real datasets often suffer from issues such as noise, missing values, and inconsistencies, which can prevent models from effectively converging and generalizing. Some earlier works have attempted to integrate synthetic data with real data to address these challenges. For instance, TimeGAN has been used to generate synthetic time series data that preserves temporal dynamics, while various GAN variants have been employed to enhance data quality. However, these studies have often lacked rigorous validation methods and have not fully explored the integration of synthetic data with real data using comprehensive cross-validation techniques. Consequently, gaps remain in terms of effectively combining synthetic and real data and applying robust time series cross-validation methods to ensure model reliability and accuracy.

Motivated by these challenges, this dissertation aims to address the following **research questions**:

- What are the best practices for generating high-quality synthetic time series data that maintains the statistical properties and temporal dynamics of real data?
- How can synthetic data be effectively integrated with real data to improve the training convergence and generalization of time series anomaly detection models?
- How can Rolling Window Time Series Stratified k-Fold Cross-Validation be applied along with real and synthetic data to enhance model evaluation robustness and reduce false positive and false negative prediction errors in time series anomaly detection?

The underlying **research objectives** are:

- To establish methodologies for producing high quality synthetic data that accurately reflects the complexities and nuances of real time series data, ensuring that models trained on this data can effectively generalize to real-world scenarios.
- To develop integration strategies that leverage the strengths of both synthetic and real data, enhancing the robustness and reliability of anomaly detection models.
- To implement and validate the Rolling Window k-Fold Cross-Validation technique to maintain temporal integrity and ensure comprehensive model evaluation, ultimately leading to more accurate and reliable anomaly detection.

The following subsection highlights the key concepts for addressing the above research questions in our thesis work.

1.6.2 Solution Methodologies

To mitigate the aforementioned issues, this dissertation follows the solution methodologies outlined in **Figure 1.1**. Initially, we examine state-of-the-art works in Chapter 2, and subsequently, Chapter 3 focuses on developing a solution mechanism aimed at maximizing model convergence and generalizability for enhanced time series anomaly detection accuracy with reduced prediction errors. The proposed irsRSk framework draws inspiration from the use of synthetic data in computer vision model training [20].

The core philosophy behind integrating real and synthetic data, along with using rolling window time series k-fold cross-validation, is to leverage synthetic data's strengths to address real data's limitations. By training the models on both real and synthetic data, we aim to enhance model performance. This process involves careful consideration of hyperparameters, including epochs, batch size, data preprocessing, and dimensionality reduction techniques. Optimizers such as ADAM are employed to fine-tune the model training process, ensuring efficient convergence and improved accuracy.

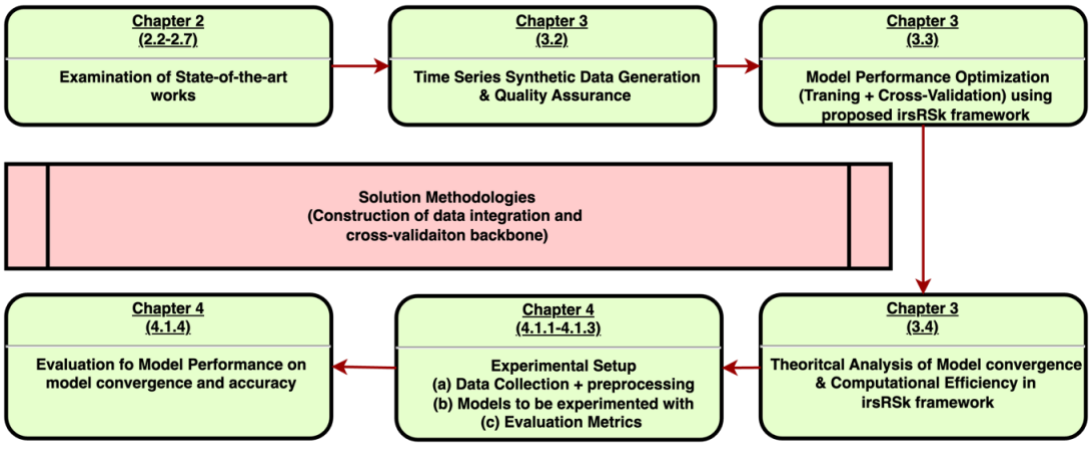


Figure 1.1: A block diagram showing roadmap of deriving and evaluating different components of the proposed solution

Addressing integration and training obstacles, we propose Rolling Window k-Fold Cross-Validation to maintain temporal order and ensure comprehensive model evaluation. This approach calculates **pass rates** to enhance computational efficiency and is validated through precision, recall, F1 score, and confusion matrix metrics. In the irsRSk framework, integrating synthetic and real data is pivotal, particularly for time-constrained real-time environments, leading to more robust training and improved anomaly detection. In summary, combining real and synthetic data with Rolling Window Time Series Stratified k-Fold Cross-Validation significantly enhances model convergence, prediction accuracy, and generalizability, ensuring comprehensive evaluation and reliable time series anomaly detection.

1.7 Contribution of the Dissertation

This research aligns with several United Nations Sustainable Development Goals (SDGs), notably Goal 9 (Industry, Innovation, and Infrastructure), and Goal 17 (Partnerships for the Goals). The development of the irsRSK framework, which integrates synthetic data with real data using Rolling Window Time Series Stratified k-Fold Cross-Validation optimizes model convergence and generalizability for time series anomaly detection. Alignign to the goal of SGD, this framework for time series anomaly detection enhances the accuracy and reliability of monitoring critical infrastructure, promoting resilient and efficient systems essential for sustainable industrialization. The interdisciplinary and collaborative nature of this work exemplifies the spirit of Goal 17, fostering partnerships among governments, industries, and research institutions. By integrating advanced data science methodologies with practical applications, this research contributes to the global effort toward achieving sustainable development objectives.

By addressing the limitations of existing statistical, machine learning, and deep learning models, this research provides a robust methodology for enhancing prediction accuracy and reducing false positives and negatives. The major controbutions of this research are as follows:

- The introduction of pTimeGAN, an enhanced synthetic data generation approach combining TimeGAN with PCA for dimensionality reduction. This innovation ensures the generation of high-quality synthetic data that maintains the temporal dependencies and complex patterns of real data, while also addressing issues such as data scarcity, imbalance, and noise.
- The integration of synthetic data with real data creates a balanced and comprehensive training set, improving model performance, particularly in complex and high-dimensional datasets. The use of irsRSk ensures that the temporal integrity of time series data is preserved during model evaluation, providing a more realistic and rigorous assessment of model performance.
- Through extensive experiments with three open-source datasets (Time Series Forecasting, Electricity Consumption, and Air Quality Prediction), the proposed framework demonstrates superior performance in model generalization, validation, and convergence. By comparing irsRSk with other state-of-the-art methods (such as TimeGAN with k-fold, CGAN with Stratified k-fold, DoppelGANger with Time Series Cross-Validation, VAE with Stratified k-fold, and SMOTE with Time Series Cross-Validation), the research highlights the effectiveness of the proposed approach.
- Moreover, the framework addresses early window issues in time series data during cross-validation and incorporates a pass_rate metric to enhance computational efficiency, validated through precision, recall, F1 score, and confusion matrix metrics.

This comprehensive approach ensures robust, scalable, and accurate anomaly detection in time series data, making a significant contribution to the field and providing a valuable resource for further research.

1.8 Organization of the Dissertation

In this dissertation, the rest of the chapters are organized as follows. Chapter 2 discusses the state-of-the-art works elaborately focusing on high quality time series sysnthetic data generation techniques and integration and cross-valdiation technqiues to enhance model convergence and accuracny. In Chapter 3, we construct the propsoed integrating real and synthetic data with Rolling window Time Series Stratified k-fold cross-validation (irsRSk) framework to optimize time series anomaly detection model's performace and accuracy with high quality synthetic data generation. Furthermore, Chapter 4 presents the performance evaluation of the propsoed framework compared to the state-of-the-art works in the earlier studies to solidify the enhancement offered by irsRSk framework. Finally, we conclude the dissertation in Chapter 5 by summarizing the research findings, along with the future extensions of this work.

Chapter 2

Literature Review

In this chapter, we provide an overview of necessary background studies for high-quality time series synthetic data generation and the integration of real and synthetic data in model training using Rolling Window Stratified Cross-Validation. This approach aims to enhance model convergence and accuracy while minimizing false positive and false negative errors. Additionally, we discuss data collection and preprocessing strategies to maximize the quality of synthetic data generated from real data.

2.1 Introduction

In recent years, the generation and utilization of synthetic data have garnered significant attention in the realm of time series anomaly detection. This chapter aims to provide a comprehensive literature review on the critical aspects of high-quality time series synthetic data generation, integration with real data, and the application of Cross-Validation for model training and generalization. These components are essential for enhancing model performance, convergence, and accuracy while mitigating issues such as false positives and false negatives.

Synthetic data generation is a pivotal area of research that addresses several limitations of real data, including scarcity, privacy concerns, and imbalance. Various strategies have been developed to produce synthetic data that closely mimics the statistical properties and temporal dynamics of real data. Techniques like TimeGAN, which integrates GANs with time series data, have demonstrated substantial improvements in generating realistic synthetic data. For instance, [19] highlighted the efficacy of TimeGAN in maintaining the temporal dependencies and patterns present in real data. These advancements lay the foundation for the subsequent integration of

synthetic and real data. Ensuring the quality of synthetic data is paramount to its utility in model training. Quality assurance strategies, covered in [9],[12],[13], involve statistical validation and temporal consistency checks to ensure the generated data is both statistically and temporally analogous to real data. Research studies [14],[21],[19] emphasized the importance of rigorous quality checks in synthetic data, which directly impact the performance and reliability of models trained on such data. This aligns with our objective to develop synthetic data that not only mirrors real data characteristics but also enhances model robustness. The integration of synthetic data with real data necessitates robust cross-validation strategies to evaluate model performance accurately. Traditional k-fold cross-validation methods fall short when applied to time series data due to their inability to maintain temporal integrity. Several studies [21],[23],[24],[14] attempted to addresses this limitation by ensuring temporal sequence preservation during model evaluation. Theire approaches aims to prevent data leakage and provides a more comprehensive assessment of model performance. Data collection and preprocessing, as detailed in [12],[13],[17], play a crucial role in the overall quality of synthetic data generation. Effective preprocessing techniques, including normalization, handling missing values, and feature engineering, ensure that the data is clean and consistent, ready for synthetic data generation. The study of [22] underscored the significance of thorough preprocessing in enhancing the quality of synthetic data and the subsequent model training process. Data collection and preprocessing, as detailed in [15],[18],[25] play a crucial role in the overall quality of synthetic data generation. Effective preprocessing techniques, including normalization, handling missing values, and feature engineering, ensure that the data is clean and consistent, ready for synthetic data generation. [24] underscored the significance of thorough preprocessing in enhancing the quality of synthetic data and the subsequent model training process.

In summary, this chapter sets the stage for understanding the intricate processes involved in generating high-quality synthetic data, integrating it with real data, and employing robust cross-validation techniques to enhance model performance in time series anomaly detection. Therefore, the rest of the chapter is organized as follows. An overview of synthetic data generation and quality assurance strategies is presented in Section 2.2, and Section 2.3, respectively. Different cross-validation strategies discussed in Section 2.4. After that, different time series data collection and data preprocessing strategies are described in Section 2.5. Next, we discuss various time series anomaly detection models categories in statistical, deep learning and GenAI domains in Section 2.6. After that, a comparative study among the state-of-the-art works is presented in Section 2.7. We discuss limitations of the existing studies in Section 2.8, and finally, we conclude the Chapter in Section 2.9.

2.2 Synthetic Data Generation Strategies

In the early days of anomaly detection and time series analysis, models were primarily trained on real datasets. However, these models often struggled with inherent issues in real data, leading researchers to identify seven key problem segments: temporal inconsistencies, scalability, training instability, data variability, data sparsity, noise robustness, and computational inefficiency. For instance, temporal inconsistencies disrupt the continuity of time series data, making it challenging for models to learn accurate patterns, as highlighted in [17],[19]. Scalability issues arise from the growing volume of data, which can overwhelm traditional training methods, as seen in large-scale industrial applications. Training instability, such as the vanishing gradient problem in deep learning models, further complicates model development. Data variability and sparsity can lead to models that fail to generalize, while noise robustness issues degrade model performance [18],[21],[24]. These challenges

underscore the need for synthetic data generation to supplement real data, providing more controlled and comprehensive datasets for model training.

The generation of synthetic data has become crucial to overcoming these limitations, leading to the development of various algorithms and methodologies. Early efforts in synthetic data generation focused on simpler statistical methods and simulations, which often fell short in capturing the complexity of real-world data [19]. However, with the advent of Generative Adversarial Networks (GANs), the landscape of synthetic data generation changed dramatically. GANs [18], comprising a generator and a discriminator, work in tandem to create realistic synthetic data by learning from real data distributions. The studies of [19],[17],[20] have demonstrated that, GANs could generate high-quality synthetic images, a breakthrough that paved the way for their application in time series data. Several advanced GAN variants have since been developed, each addressing specific limitations of the original GAN framework [18]. Conditional GANs (CGANs) incorporate conditional information into the generation process, improving the relevance of synthetic data. For example, [19] showed that CGANs could generate images conditioned on class labels, which can be extended to generate time series data conditioned on specific variables like seasonality or trends. Wasserstein GANs (WGANs) and their variant with Gradient Penalty (WGAN-GP) [21] enhance training stability and quality of generated data. The study of [11] demonstrated that WGANs mitigate issues like mode collapse, which is critical in generating diverse time series data. DRAGAN focuses on improving convergence and stability, addressing the sensitivity of GANs to initialization and hyperparameters, as noted by [20]. Cramer GAN and CWGAN-GP introduce additional regularization techniques for better performance, ensuring that the generated data maintains the statistical properties of the original dataset. CTGAN (Conditional Tabular GAN), developed by [22], is specifically designed for generating tabular data, preserving

relationships between columns, which is crucial for generating synthetic datasets that reflect the complexity of time series data. Gaussian Mixture Models and Sequential Data Generators provide alternative approaches for specific types of data, offering robust solutions for generating synthetic data in various applications [22],[25],[23].

Variational Autoencoders (VAEs) and the Synthetic Minority Over-sampling Technique (SMOTE) are other notable synthetic data generation techniques. VAEs, as presented by [12], are capable of generating high-dimensional data by learning latent representations. VAEs are particularly useful for their ability to capture complex data distributions and provide smooth interpolation between data points. However, VAEs often struggle with generating sharp and detailed features, which can limit their effectiveness in some applications [14]. SMOTE, introduced by [23], is designed to address class imbalance by creating synthetic examples of the minority class. While SMOTE is effective for balancing datasets and improving model performance, it may not capture the full complexity of real-world data distributions, especially in time series applications.

Among these, TimeGAN and DoppelGANger have shown superior performance in maintaining temporal consistency, high scalability, data diversity, and training stability, making them ideal for time series anomaly detection. TimeGAN integrates both supervised and unsupervised learning objectives to capture the temporal dynamics of time series data effectively, as demonstrated by [19] DoppelGANger further enhances scalability and diversity by generating large-scale, high-dimensional datasets. In a comparative study, DoppelGANger was found to outperform traditional methods in scenarios requiring high fidelity and temporal accuracy[19],[20]. WGAN-GP [23], while lacking inherent temporal consistency, is noted for its high-quality data generation and stable training process, providing a robust alternative when temporal aspects are less critical. Benchmarking these GAN

variants and other techniques such as VAEs and SMOTE against criteria like temporal consistency, scalability, data diversity, and training stability, researchers [23],[24],[19],[8] have found that TimeGAN and DoppelGANger consistently outperform others in scenarios requiring temporal precision. Studies have demonstrated that these models effectively preserve the temporal patterns and structural properties of the original datasets, making them highly suitable for time series applications. For instance, in financial data modeling, TimeGAN was shown to maintain the sequence dependency crucial for accurate anomaly detection [25].

However, synthetic data generation is not without its limitations. One of the major challenges is ensuring that synthetic data accurately mimics the characteristics of real data, especially when the dataset is highly sparse, contains significant temporal inconsistencies, or has high variability. Synthetic data generation techniques often struggle to replicate these complex characteristics faithfully [13]. For example, [24] pointed out that many GAN-based models fail to generate high-dimensional time series data that accurately reflects the statistical properties of the real data. Additionally, synthetic data generation methods can introduce artifacts or biases that were not present in the real data, potentially leading to incorrect model training outcomes.

Table 2.1: State-of-the-art Synthetic Data Generation Techniques

Algorithm	TC	SC	DD	TS	QS	EI	CE
GAN [23]	No	Yes	Yes	No	Moderate	High	Moderate
CGAN (Conditional GAN) [24]	No	Yes	High	No	Moderate	Moderate	Moderate
WGAN (Wasserstein GAN) [13]	No	Yes	High	Yes	High	Moderate	High
WGAN-GP (Wasserstein GAN with Gradient Penalty) [18]	No	Yes	High	Yes	High	Low	High
DRAGAN (On Convergence and Stability of GANs) [19]	No	Yes	Moderate	Yes	Moderate	Low	High

Cramer GAN [20]	No	Yes	High	Yes	High	Low	High
CWGAN-GP (Conditional Wasserstein GAN with Gradient Penalty) [21]	No	Yes	High	Yes	High	Low	High
CTGAN (Conditional Tabular GAN) [8]	No	Yes	High	Yes	High	Low	High
Gaussian Mixture [9]	No	Yes	Moderate	No	Moderate	High	High
Sequential data [13]	Yes	Moderate	High	Moderate	High	Moderate	Moderate
TimeGAN [14]	Yes	High	High	High	High	Low	High
DoppelGANger [17]	Yes	High	High	High	High	Low	High

*** TC- Temporal Consistencies, SC-Scalability, DD-Data Diversity, TS-Training Stability, QS- Quality of generated Synthetic Dat, EI- Ease of implementation, CE-Computational Efficiency*

Failure cases in earlier research have highlighted these issues. For example, studies have shown that when training models on synthetic data generated from highly sparse datasets, the models tend to overfit to the synthetic data patterns, resulting in poor generalization to real data. Temporal inconsistencies in synthetic data can lead to inaccurate anomaly detection, as the model may learn incorrect temporal dependencies. These gaps underscore the need for continuous improvement in synthetic data methodologies to ensure that synthetic datasets can adequately support model training and validation processes.

The evolution of synthetic data generation, particularly with the development of sophisticated GAN variants, VAEs, and SMOTE, has significantly enhanced the ability to create high-quality datasets for time series anomaly detection [13],[12],[25]. However, ongoing efforts are required to address the remaining challenges and optimize these methodologies for practical applications. TimeGAN and DoppelGANger stand out for their ability to maintain temporal consistency and handle complex data, making them essential tools in the advancement of time series anomaly detection models.

2.3 Synthetic Data Quality Assurance Strategies

Ensuring the quality of synthetic data, particularly for time series, has been a critical focus of research due to the inherent challenges in replicating the complex characteristics of real-world data. Earlier methodologies to ensure synthetic data quality included various statistical similarity tests, each with its strengths and limitations. For example, basic statistical measures such as mean, variance, and correlation were initially used to compare synthetic and real data [7]. However, these measures often failed to capture the temporal dependencies and intricate patterns present in time series data [7],[9].

State-of-the-art methods for assessing synthetic data quality emerged to address these limitations, focusing on more comprehensive statistical tests. Techniques like the Kolmogorov-Smirnov test and Chi-square test became popular for comparing the distributions of synthetic and real data. The Kolmogorov-Smirnov test [21] measures the maximum difference between the empirical distribution functions of two samples, providing a non-parametric way to assess distributional similarity. The Chi-square test evaluates the differences between observed and expected frequencies, which can be particularly useful for categorical data [22]. While these tests improved the understanding of distributional similarity, they still struggled to fully capture the dynamic properties of time series data, such as trends and seasonality [23],[22].

As the field evolved, researchers began incorporating dynamic properties into the assessment of synthetic data quality. Methods such as trend and seasonality analysis, as well as spectral analysis, were developed to ensure that synthetic data preserved the temporal characteristics of real data. Spectral analysis, for instance, can reveal periodicities in the data by examining the frequency domain, providing insights

into whether synthetic data accurately reflects the cyclical patterns of real data [24]. Visual inspection strategies also became crucial in evaluating synthetic data quality. Techniques like time series plots, heatmaps, and Laplace (Lap) plots were employed to visually compare the temporal structure and patterns between synthetic and real data [19]. These visual methods provided intuitive and immediate insights into the fidelity of synthetic data, complementing the more quantitative statistical tests [17].

Machine learning-based evaluations introduced a new dimension to synthetic data quality assurance. By assessing training performance, researchers could determine how well models trained on synthetic data generalize to real data. Metrics such as accuracy and F1 scores provided quantitative measures of this generalization ability [11]. Adversarial tests, particularly those involving GANs, further advanced this approach. In adversarial settings, a discriminator attempts to distinguish between real and synthetic data, providing a robust mechanism for evaluating the realism of synthetic data. The better the synthetic data fools the discriminator, the higher its quality [14]. Feature-based evaluation techniques, including Principal Component Analysis (PCA) [12] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [19], played a significant role in assessing the impact of dimensionality reduction on synthetic data quality. These methods allowed researchers [17],[19],[7] to visualize high-dimensional data in lower-dimensional spaces, making it easier to compare the structure and distribution of synthetic and real data. Consistency checks, focusing on temporal consistencies and anomaly detection, ensured that synthetic data maintained the same temporal order and irregularity patterns as real data [8].

Similarity measures such as Dynamic Time Warping (DTW) [13] and Earth Mover's Distance (EMD) [15] provided additional tools for comparing the sequences and distributions of synthetic and real data. DTW measures the optimal alignment between two time series, capturing both temporal shifts and amplitude variations [17].

EMD quantifies the dissimilarity between two probability distributions, offering a holistic view of distributional differences [19].

In comparative assessments, TimeGAN and DoppelGANger have been found to excel in maintaining temporal consistency, high scalability, data diversity, and training stability. For example, [19] demonstrated that TimeGAN outperformed traditional GANs in preserving temporal dependencies crucial for time series data. Similarly, [20] showed that DoppelGANger generated synthetic data with higher fidelity and temporal accuracy compared to other models.

Table 2.2: Existing Synthetic Data Quality Assurance Strategies

Strategy	**Strengths**	**Limitations**	**QA Benchmarks**				
			TC	SC	DD	TS	VIZ
Basic Statistical Measures (mean, variance, correlation) [13]	Simple and easy to compute	Fails to capture complex patterns and dependencies	L	H	L	M	L
Kolmogorov-Smirnov Test [24]	Non-parametric, distributional similarity	Limited in capturing temporal dynamics	L	H	M	M	L
Chi-square Test [22]	Useful for categorical data	Not effective for continuous time series	L	H	M	M	L
Spectral Analysis [23]	Captures periodicities and frequency domain characteristics	Requires specialized knowledge	H	M	M	M	M
Time Series Plots [8]	Provides intuitive visual comparison	Subjective interpretation	H	L	M	M	H
Heatmaps [9]	Visualizes correlation matrices effectively	Can be difficult to interpret for large datasets	M	M	M	M	H
Laplace (Lap) Plots [19]	Effective for visualizing local structures	Not widely used, interpretation can be subjective	M	M	M	M	H
PCA (Principal Component Analysis) [12]	Reduces dimensionality, highlights main data variations	May lose important temporal information	M	M	H	M	M
t-SNE (t-Distributed Stochastic Neighbor Embedding) [19]	Effective for visualizing high-dimensional data	Computationally intensive, can lose temporal info	M	M	H	M	M
Training Performance Assessment [21]	Evaluates model generalization on synthetic data	Requires extensive computational resources	H	H	H	M	M
Adversarial Tests (e.g., GAN-based) [23]	Robust evaluation of synthetic data realism	Complex setup, may require large datasets	M	H	H	H	M

Consistency Checks (temporal consistency, anomaly detection) [25]	Ensures synthetic data maintains key temporal properties	Can be challenging to implement accurately	H	M	M	H	M
Similarity Measures (DTW, EMD) [13],[15]	Detailed comparison of sequences and distributions	Computationally intensive, requires careful setup	H	M	M	M	M

** *H- High, M-Medium, L-Low; TC- Temporal Consistency, SC- Scalability, DD-Data Diversity, TS-Training Stability, VIZ-Visualization & Interpretability*

Despite these advancements, synthetic data generation techniques still face challenges, particularly in capturing the full complexity of real-world time series data. For instance, [25] noted that many GAN-based models struggle with generating high-dimensional time series data that accurately reflects the real data's statistical properties. Moreover, the introduction of artifacts or biases in synthetic data generation can lead to incorrect model training outcomes, highlighting the need for continuous improvement in synthetic data methodologies.

The evolution of synthetic data generation, particularly with the development of sophisticated GAN variants, VAEs, and SMOTE, has significantly enhanced the ability to create high-quality datasets for time series anomaly detection [24]. However, ongoing efforts are required to address the remaining challenges and optimize these methodologies for practical applications [22],[5],[29]. TimeGAN [14] and DoppelGANger [17] stand out for their ability to maintain temporal consistency and handle complex data, making them essential tools in the advancement of time series anomaly detection models. Through the integration of statistical, dynamic, visual, machine learning-based, and feature-based evaluations, along with consistency checks and similarity measures, researchers have developed a comprehensive framework for ensuring the robustness and reliability of synthetic datasets.

2.4 Cross-Validation Strategies

Cross-validation is a pivotal technique in machine learning and statistical modeling, aimed at assessing the generalizability of a model by partitioning data into subsets to train and validate the model multiple times. Over the years, various cross-validation strategies have evolved to address different types of data and modeling challenges, from simple random splits to more complex time-aware strategies.

Early research [5],[8],[12] primarily focused on basic methods like k-fold cross-validation, where the dataset is divided into k subsets, and the model is trained on k-1 subsets while validated on the remaining subset. This process is repeated k times, with each subset used exactly once as the validation data. While k-fold cross-validation is straightforward and effective for many applications, it struggles with imbalanced datasets and does not account for the temporal dependencies in time series data. Stratified k-fold cross-validation improves upon the basic k-fold approach by ensuring that each fold maintains the same class distribution as the overall dataset, making it particularly useful for classification tasks with imbalanced classes [15]. Leave-One-Out Cross-Validation (LOOCV) [16] takes this further by using each individual data point as a validation set and all remaining data points as the training set. LOOCV provides an exhaustive validation but is computationally expensive and prone to high variance in the error estimate. Leave-P-Out Cross-Validation (LPOCV) [17] generalizes LOOCV by leaving p data points out for validation, but it quickly becomes impractical as p increases due to the combinatorial explosion of possible splits. Holdout validation [18], where a single split divides the data into training and validation sets, is simple and fast but may not represent the full variability of the data. Repeated k-fold cross-validation [20] repeats the k-fold process multiple times with different random splits, providing a more robust estimate of model performance. However, like k-fold, it does

not handle time dependencies well. Time series cross-validation [21], is designed for time series data by maintaining the temporal order of observations. This method trains the model on a growing window of time and validates it on the subsequent time period, effectively addressing the temporal dependencies but potentially suffering from reduced training data in early [21],[12]. Nested cross-validation is used for model selection and hyperparameter tuning, involving an outer loop of cross-validation for evaluating model performance and an inner loop for hyperparameter optimization. This approach reduces the risk of overfitting but is computationally intensive [16]. Group k-fold cross-validation is used when data points are grouped, ensuring that all data points from a group are either in the training set or validation set, which is crucial for avoiding data leakage in grouped data scenarios [14]. Stratified Shuffle Split and Monte Carlo cross-validation provide flexible alternatives by randomly splitting the data multiple times and averaging the results, offering robustness against data splits but still struggling with time dependencies [10]. Monte Carlo cross-validation involves repeated random sampling, balancing computational efficiency and thorough evaluation but can miss temporal patterns in time series data.

Integrating synthetic data with real data in cross-validation strategies presents additional challenges. The inclusion of synthetic data aims to address issues like data sparsity and imbalance, but it requires careful handling to avoid introducing biases. Strategies such as stratified k-fold and Monte Carlo cross-validation can integrate synthetic data by maintaining similar distributions in each fold, yet they must ensure that the synthetic data does not dominate the training process, which could lead to overfitting on synthetic patterns rather than real-world variability [19]. At high data volumes, particularly with large synthetic datasets, these cross-validation strategies can encounter scalability issues, leading to prolonged training times or even infinite runs. For instance, Leave-P-Out Cross-Validation becomes infeasible with large

datasets due to the sheer number of combinations [18].Similarly, the computational burden of nested cross-validation can become prohibitive with large synthetic datasets, requiring substantial computational resources to converge [8].

Table 2.3 compares various cross-validation strategies across six benchmark criteria: Temporal Consistency, Scalability, Handling Imbalanced Data, Computational Efficiency, and Model Generalizability. Each strategy has its strengths and limitations, emphasizing the need for selecting the appropriate method based on the specific characteristics and requirements of the dataset and the modeling task.

Table 2.3: Comparative analysis of Existing Cross-Validation Strategies

Strategy	**Strengths**	**Limitations**	**Benchmarks**					
			TC	SC	HDI	CE	MG	ROF
K-Fold Cross-Validation [12]	Simple, effective for many datasets	Does not handle temporal dependencies well	L	H	M	M	M	M
Stratified K-Fold Cross-Validation [16]	Maintains class distribution	Still not suitable for time series data	L	H	H	M	M	M
Leave-One-Out Cross-Validation (LOOCV) [17]	Exhaustive, uses all data points	Computationally expensive, high variance	L	L	H	L	H	L
Leave-P-Out Cross-Validation (LPOCV) [20]	Generalization of LOOCV	Becomes impractical for large p	L	L	H	L	H	L
Holdout Validation [21]	Simple, fast	May not represent data variability well	L	H	L	H	L	H
Repeated K-Fold Cross-Validation [16]	More robust performance estimate	Does not handle temporal dependencies well	L	H	M	M	H	M
Time Series Cross-Validation [14]	Maintains temporal order	Reduced training data in early windows	H	M	L	M	H	L
Nested Cross-Validation [10]	Reduces risk of overfitting, good for model selection	Computationally intensive	L	M	H	L	H	L
Group K-Fold Cross-Validation [11]	Avoids data leakage in grouped data	Less effective for non-grouped data	L	M	H	M	H	M
Stratified Shuffle Split [18]	Random splitting with class balance	Not suitable for time series data	L	H	H	M	M	M
Monte Carlo Cross-Validation [17]	Random sampling, balances thoroughness and efficiency	Can miss temporal patterns	L	H	M	M	M	M

*** H-High, M-Medium, L-Low; TC-Temporal Consistency, SC- Scalability, HDI-Handling Data Imbalance, CE-Computational Efficien, MG- Model Generalization, ROF- Risk of Overfitting*

Recent evaluations have shown that while these strategies work well for static data, they face significant hurdles with time series data due to the inherent temporal dependencies. For instance, time series cross-validation methods like rolling window validation effectively maintain the order of observations but often lead to incomplete training sets in early windows, affecting model convergence and performance [18]. Moreover, traditional k-fold and its variants fail to incorporate temporal dynamics, leading to potential data leakage and unrealistic performance estimates [18]. Among these strategies, Time Series k-Fold Cross-Validation is particularly well-suited for our context, where we handle both real and synthetic time series data. This method ensures that the temporal order is preserved while providing robust evaluation metrics across multiple folds. By integrating synthetic data within this framework, we can address data sparsity and imbalance, leading to better model training convergence and generalization. None of the mentioned cross-validation techniques explicitly implement "pass rate" as an auxiliary metric for model evaluation. However, some researchers have proposed using pass rate metrics to evaluate model performance during training [19],[20],[22]. Incorporating pass rate within the Time Series k-Fold Cross-Validation framework could further enhance the evaluation process by providing an additional layer of validation, ensuring that models not only perform well on average but also maintain consistency across different validation sets [20]. This integration could be achieved by calculating the pass rate as the proportion of folds where the model meets predefined performance thresholds, thus offering a more comprehensive assessment of model robustness and reliability.

2.5 Data Collection and Preprocessing Strategies

The collection and preprocessing of time series data are critical steps in ensuring the reliability and validity of anomaly detection models. Early evaluations of time series data collection and preprocessing strategies have evolved significantly to handle challenges such as data imbalance, variability, sparsity, and missing values. Initially, data collection relied heavily on manual processes and bespoke solutions tailored to specific datasets, often leading to inconsistencies and biases [15],[17],[19],[21].

Platforms like HuggingFace [1], Kaggle [2], and Datadog [3] have democratized access to diverse and extensive time series datasets across various domains, including finance, healthcare, manufacturing, and cybersecurity. These open-source platforms provide high-quality datasets that researchers leverage to strengthen their models and enhance the robustness of their findings. For example, Kaggle hosts competitions that encourage the sharing of large, annotated time series datasets, which are instrumental in advancing the state-of-the-art in anomaly detection [2]. Similarly, HuggingFace provides a repository of datasets that support natural language processing and time series analysis, facilitating access to diverse data sources for researchers [1]. Alternative data collection strategies include data scraping, API-based extraction, database querying, and log file analysis. These methods allow researchers to acquire real-time and historical data directly from operational systems, enhancing the dataset's relevance and applicability [11]. However, the availability of comprehensive datasets on platforms like Kaggle and HuggingFace often reduces the need for manual data acquisition, thereby minimizing the internal threats to validity. These pre-compiled datasets generally maintain higher consistency and standardization, although they may still suffer from impurities, imbalances, and missing values that need to be addressed during preprocessing [12],[13],[24].

Preprocessing strategies for time series data have traditionally included techniques such as imputation for missing values, normalization, and standardization. Early methods focused on simple imputation techniques like mean or median replacement and linear interpolation. While effective to some extent, these methods often fail to capture the underlying data dynamics, leading to potential biases and inaccuracies [19]. Recent innovations in data preprocessing have introduced more sophisticated techniques to handle the complexities of time series data. For example, Seasonal and Trend decomposition using Loess (STL) [23] is a powerful method for decomposing a time series into seasonal, trend, and residual components, enhancing the model's ability to understand and predict patterns. Data windowing [24] involves creating overlapping or non-overlapping windows of data points to capture temporal dependencies and improve model training. Time series encoding techniques, such as temporal feature extraction and embedding, help transform raw time series data into more informative representations, facilitating better learning by machine learning models [18]. Seasonal adjustment methods remove seasonal effects from the data, allowing the models to focus on trends and anomalies without seasonal noise.

Table 2.4 compares various data preprocessing strategies across seven benchmark criteria: Handling Imbalanced Data, Handling Missing Values, Scalability, Maintaining Temporal Dynamics, Suitability for High-Dimensional Data, and Strengths and Limitations. Each strategy has its strengths and limitations, emphasizing the need for selecting the appropriate method based on the specific characteristics and requirements of the dataset and the modeling task. For our research context, advanced techniques like STL decomposition, data windowing, and time series encoding appear best suited due to their effectiveness in handling complex, high-dimensional, and voluminous time series data.

Table 2.4: Data Preprocessing Strategies for Time Series Data

Strategy	**Strengths**	**Limitations**	**Benchmarks**				
			HDI	HMV	SC	MTD	SHD
Mean/Median Imputation [7]	Simple, easy to implement	Fails to capture data dynamics	L	M	H	L	L
Linear Interpolation [11]	Preserves trends better than mean/median	Can introduce biases	M	M	H	M	L
Seasonal and Trend Decomposition using Loess (STL) [23]	Captures complex seasonal and trend patterns	Computationally intensive	H	M	M	H	M
Data Windowing [24]	Captures temporal dependencies effectively	Can result in large data volumes	M	M	H	H	H
Time Series Encoding [15]	Enhances feature representation	Requires careful design	M	L	M	H	H
Normalization/Standardization [8]	Simplifies data range	May not capture underlying distribution	L	L	H	L	M
Temporal Feature Extraction [10]	Provides rich features from raw data	Computationally intensive	M	M	M	H	H
Seasonal Adjustment [13]	Removes seasonal effects to focus on trends	May lose important seasonal info	M	L	M	H	M

** *H-High, M-Medium, L-Low; (a) Handling Imbalanced Data (b) Handling Missing Values (c) Scalability (d) Maintaining Temporal Dynamics (e) Suitability for High-Dimensional Data*

However, the evolution of data collection and preprocessing strategies has been driven by the need to handle increasingly complex and voluminous time series data. Platforms like HuggingFace and Kaggle have facilitated access to high-quality datasets, while advanced preprocessing techniques have addressed many of the challenges associated with time series analysis. By leveraging these innovations, researchers can develop more robust and accurate anomaly detection models. Based on the current context, employing techniques such as STL decomposition, data windowing, and time series encoding will best suit our needs, ensuring the effective preprocessing of high-volume and complex datasets, consistent with earlier research findings and comparative analyses.

2.6 Evolution of Time Series Anomaly Detection Models

The evolution of machine learning models for time series anomaly detection marks a significant improvement over traditional statistical-based models, which often lack the capability to learn complex patterns and dependencies inherent in time series data. Traditional statistical models like ARIMA [12] and GARCH [13] have been widely used due to their simplicity and interpretability. However, these models typically assume linearity and stationarity, limiting their effectiveness in capturing the intricate and dynamic behaviors present in real-world time series data. In contrast, machine learning models, particularly those based on deep learning and neural networks, have demonstrated superior performance by leveraging their ability to learn from large amounts of data and capture non-linear relationships.

Early machine learning models for time series anomaly detection include decision trees, support vector machines (SVM) [14], and ensemble methods like random forests and gradient boosting machines. These models have been effective in many applications but often struggle with temporal dependencies and require extensive feature engineering. With the advent of deep learning, models such as Long Short-Term Memory (LSTM) networks [15], Autoencoders, and Generative Adversarial Networks (GANs) [16] have become prominent in time series analysis. LSTM networks, in particular, are well-suited for sequence learning due to their ability to maintain long-term dependencies, making them effective for time series forecasting and anomaly detection [14]. Autoencoders have been employed for their capability to learn compressed representations of data, which can then be used to detect anomalies as deviations from the learned patterns [14],[15]. GANs, originally designed for image synthesis, have been adapted to generate synthetic time series data and detect anomalies through adversarial training [16]. Generative AI models like Isolation Forest

[17] and Prophet [18] have also been leveraged for time series anomaly detection. Isolation Forest, an ensemble method, isolates anomalies by randomly partitioning data, which is effective in high-dimensional settings but may struggle with temporal dependencies. Prophet, developed by Facebook, is tailored for forecasting with strong seasonal effects and missing data, making it useful in business applications [18]. Other sectors where these models contribute include finance, healthcare, cybersecurity, and manufacturing, where they are used for fraud detection, patient monitoring, network security, and predictive maintenance, respectively.

Table 2.5 categorizes various models used in anomaly detection, detailing their strengths, limitations, and correlations with different types of models—statistical, deep learning, and GenAI. Statistical models like ARIMA and Holt-Winters are efficient and stable but struggle with high-dimensional data and noise [16]. Deep learning models such as LSTM and Autoencoders handle complex temporal dependencies and dimensionality reduction but are computationally intensive and prone to overfitting [19]. GenAI models like Isolation Forest and One-Class SVM are robust against noise and data imbalance but have limitations in handling complex temporal dependencies and high-dimensional data. Each model's challenges with specific datasets are also highlighted, such as difficulties in capturing variability, noise, and seasonality [17]. Additionally, the table emphasizes the need for careful parameter tuning and computational resources, as well as the importance of selecting the appropriate model based on the specific characteristics and challenges of the dataset being analyzed. For example, ARIMA and Holt-Winters show a 1.0 correlation with statistical models but only a 0.5 correlation with deep learning models and a 0.3 correlation with GenAI models. Deep learning models like LSTM and Autoencoders exhibit a 1.0 correlation with deep learning but a lower correlation with statistical (0.5 and 0.4, respectively) and GenAI models (0.7 and 0.6, respectively) [19]. GenAI models

such as Isolation Forest and One-Class SVM show a 1.0 correlation with GenAI models but lower correlations with statistical (0.3) and deep learning models (0.7 and 0.6, respectively) [20]. These statistics highlight the varying degrees of compatibility and performance across different model types and the necessity to choosing the most suitable model based on specific dataset challenges.

Table 2.5: Capabilities and Limitations of Time Series Anomaly Detection Models and Dataset Challenges

Models	Strengths	Limitations	DCS	DCD	DCG	Challenges
ARIMA [12]	Efficient, stable, handles linear and seasonal data patterns	Struggles with high-dimensional data, noise, and imbalanced datasets; overfitting	1.0	0.5	0.3	Struggles with temporal inconsistencies and capturing variability (ETDataset, Bitcoin Historical Data)
Holt-Winters [14]	Effective in seasonal data, less resource-intensive	Limited handling of non-linear patterns, high-dimensionality, and noise	1.0	0.5	0.3	Fails to handle high variability and abrupt changes (Electricity Consumption, Bitcoin Historical Data)
GARCH [13]	Models volatility well, effective in financial time series	Complexity, computationally intensive, less effective with non-financial data	0.9	0.4	.3	Fails in handling seasonal trends and diverse time series data (Store Sales, Monash Time Series Forecasting Repository)
LSTM [15]	Captures complex temporal dependencies, performs dimensionality reduction	Computationally intensive, prone to overfitting, requires significant resources	0.5	1.0	0.7	Challenges in handling high-dimensional data and maintaining accuracy (ETDataset, COVID-19 World Vaccination Progress)
Autoencoders [15]	Effective in dimensionality reduction, handles non-linear anomalies	High computational cost, overfitting, sensitive to hyperparameters	0.4	1.0	0.6	Struggles with data imbalance and variability (UCI Energy Metering, Weather Data)
GANs [16]	Generates high-quality synthetic data, captures complex data patterns	Computationally intensive, extensive parameter tuning required	0.5	1.0	0.7	Fails in capturing noise and robust anomaly detection (Numenta Anomaly Benchmark)
Isolation Forest [17]	Handles data imbalance well,	Limited temporal handling capabilities,	0.3	0.7	1.0	Struggles with high variability and

	robust against moderate noise	requires careful parameter tuning				occasional spikes (Electricity Consumption)
One-Class SVM [19]	Effective outlier detection, robust to noise	Sensitive to parameter selection, less effective with complex temporal dependencies	0.3	0.6	1.0	Fails in handling temporal inconsistencies and maintaining accuracy (COVID-19 World Vaccination Progress)
DBSCAN [20]	Identifies clusters of varying density, effective in non-linear anomaly detection	Parameter sensitivity, high computational cost with large datasets	0.3	0.6	1.0	Struggles with diverse time series data and generalization (Monash Time Series Forecasting Repository)
Prophet [18]	Handles seasonality, trend analysis, user-friendly with intuitive parameters	Less effective with non-linear anomalies, struggles with high-dimensional data	0.8	0.6	0.5	Fails to handle high variability and data imbalance (Bitcoin Historical Data, UCI Energy Metering)
Dynamic Time Warping (DTW) [17]	Aligns similar sequences, effective in pattern recognition	Computationally intensive, less effective with high-dimensional data	0.7	0.7	0.6	Struggles with seasonal patterns and temporal dependencies (Weather Data)
Mann-Kendall Test [16]	Detects trends in time series, non-parametric	Assumes monotonic trends, less effective with non-linear patterns	0.8	0.5	0.4	Fails in handling noise and variability (Numenta Anomaly Benchmark)
Theil-Sen Estimator [14]	Robust trend estimation, less sensitive to outliers	Computationally intensive, struggles with high-dimensional and non-linear data	0.7	0.6	0.5	Struggles with generalization and handling diverse temporal patterns (Monash Time Series Forecasting Repository)
k-Means Clustering [15]	Efficient partitioning of data, computationally efficient	Assumes spherical clusters, less effective with non-linear and temporal data	0.5	0.6	1.0	Fails in capturing complex temporal dependencies and variability (ETDataset, Bitcoin Historical Data)

** *DCS-Degree of Correlation with Statistical Model, DCD-Degree of Correlation with Deep Learning Models, DCG-Degree of Correlation with GenAI Models, Challenges- Challenges with Datasets*

While machine learning models for time series anomaly detection offer significant advancements over traditional statistical models, they come with their own set of challenges. Models like LSTM, Autoencoders, and GANs have shown great

promise but require careful handling of training data, computational resources, and parameter tuning. Isolation Forest and Prophet provide simpler, more interpretable solutions but may not handle temporal complexities as effectively [19],[20],[23].

2.7 Comparative Characteristics of the State-of-the-Art Works

In synthesizing insights from synthetic data generation strategies, quality assurance methods, cross-validation techniques, data collection and preprocessing strategies, and time series anomaly detection models, it is evident that each approach has unique strengths and limitations. Advanced models like GANs and VAEs significantly improve synthetic data quality but face challenges in training stability and scalability [19]. Quality assurance methods such as PCA and Kolmogorov-Smirnov tests provide robust evaluation but may struggle with temporal dynamics. Cross-validation strategies like rolling window and nested cross-validation maintain temporal order and reduce overfitting but can be computationally intensive [20]. Data preprocessing techniques, including STL decomposition and data windowing, effectively handle high-dimensional, high-volume data, enhancing model training and generalization [24]. Time series anomaly detection models such as LSTM, Autoencoders, and Isolation Forests offer various advantages in capturing complex patterns and dependencies but require careful handling of data quality and computational resources [21],[23].

Table 2.6 provides a comprehensive comparative analysis of the state-of-the-art strategies discussed in earlier, mapping their strengths and limitations across various benchmarks. This synthesis highlights the importance of selecting appropriate

methods based on the specific requirements of the dataset and the anomaly detection task, ensuring robust and accurate model performance.

Table 2.6: A comprehensive comparative analysis of the state-of-the-art strategies

Category	Method/Model	Strengths	Limitations	Benchmarks						
				TD	SC	HDD	TS	EI	CE	MG
Synthetic Data Generation	GANs [11]	High-quality data, generative capabilities	Training instability	M	H	H	L	L	L	H
	VAEs [12]	Captures complex distributions	Requires careful tuning	M	H	H	M	M	M	H
	SMOTE [13]	Addresses class imbalance	May not capture full data complexity	L	H	M	M	M	M	M
Synthetic Data Quality Assurance	PCA [17]	Reduces dimensionality, highlights main variations	May lose temporal info	M	M	H	M	M	M	M
	Kolmogorov-Smirnov Test [16]	Non-parametric, distributional similarity	Limited temporal dynamics	L	H	M	M	L	M	M
	Adversarial Tests (GAN-based) [18]	Robust evaluation of realism	Complex setup, large datasets needed	M	H	H	H	M	M	H
Cross-Validation Strategies	Time Series Cross-Validation [18]	Maintains temporal order	Reduced training data early	H	M	L	M	M	M	H
	Nested Cross-Validation [19]	Reduces overfitting, good for model selection	Computationally intensive	L	M	H	L	M	M	H
	Group K-Fold Cross-Validation [20]	Avoids data leakage in grouped data	Less effective for non-grouped data	L	M	H	M	M	M	H
Data Collection & Preprocessing	STL Decomposition [8]	Captures complex seasonal/trend patterns	Computationally intensive	H	M	M	H	M	M	H
	Data Windowing [15]	Captures temporal dependencies	Can result in large data volumes	M	M	M	H	M	M	H
	Time Series Encoding [17]	Enhances feature representation	Requires careful design	M	M	H	H	M	M	H
Anomaly Detection Models	ARIMA [12]	Simple, interpretable	Assumes linearity, struggles with non-stationarity	L	M	L	H	H	H	L
	GARCH [16]	Effective for volatility clustering	Computationally intensive, assumes stationarity	L	M	L	M	M	L	L
	Decision Trees [17]	Simple, interpretable	Prone to overfitting, requires feature engineering	L	H	M	M	H	H	M
	SVM [18]	Effective for high-dimensional data	Requires feature scaling, sensitive to parameters	L	M	M	M	M	L	M

Random Forest [7]	Reduces overfitting, handles missing values	Complex to interpret, computationally intensive	L	H	H	H	M	M	H
LSTM [18]	Captures long-term dependencies, effective for sequences	Computationally intensive, prone to overfitting	H	M	H	L	M	L	H
Autoencoders [19]	Learns compressed representations	Requires tuning, struggles with very high-dimensional data	M	M	H	M	M	M	H
GANs [16]	Powerful generative capabilities, high-quality data generation	Training instability, computationally intensive	M	H	H	L	L	L	H
Isolation Forest [17]	Efficient, scalable, robust to noise	Performance degrades with complex temporal patterns	L	H	M	H	M	H	M
Prophet [18]	Easy to implement, interpretable, handles seasonality	Limited in capturing non-linear anomalies	M	M	M	H	H	H	M

*** H-High, M-Medium, L-Low; TD-Temporal Dependency, SC-Scalability, HDD-Handling Data Diversity, TS-Training Stability, EI-Ease of Implementation CE-Computational Efficienc, MG-Model Generalization*

This comparative analysis underscores the necessity of a balanced approach, integrating these methods to optimize model performance for time series anomaly detection.

2.8 Limitations of the Existing Studies

The landscape of synthetic data generation for time series anomaly detection has seen considerable advancements, yet several limitations persist that hinder the effectiveness of these approaches. Despite the development of advanced models like GANs, VAEs, and SMOTE, ensuring the fidelity and quality of synthetic data remains a significant hurdle. GANs often struggle with training stability and mode collapse, resulting in synthetic data that fails to capture the full diversity and variability of real data [17],[18],[16]. VAEs can introduce biases and may not always preserve intricate temporal dynamics [19]. These issues in synthetic data generation directly impact the

subsequent quality assurance processes. Traditional techniques such as PCA and t-SNE, while effective for dimensionality reduction, may lose important temporal information [20] and statistical similarity tests like the Kolmogorov-Smirnov test fall short in ensuring temporal consistency [15].

Cross-validation techniques, crucial for model evaluation, also exhibit significant gaps. Standard methods like k-fold and LOOCV are not designed to handle temporal dependencies, leading to data leakage and unrealistic performance estimates [13],[18]. Advanced methods such as time series cross-validation and nested cross-validation address these issues to some extent but remain computationally intensive and suffer from reduced training data in early windows [15]. Furthermore, these techniques often fail to integrate synthetic data effectively with real data, leading to potential biases and overfitting if the synthetic data does not accurately represent real-world variability [19]. These limitations propagate to the preprocessing stage, where high-volume and complex datasets from platforms like HuggingFace and Kaggle contain impurities, imbalances, and missing values [1],[2]. Advanced preprocessing techniques like STL decomposition and data windowing require careful implementation and significant computational resources, adding to the overall challenge [19].

Finally, time series anomaly detection models themselves exhibit several limitations. Traditional models like ARIMA and GARCH are constrained by their assumptions of linearity and stationarity, often violated in real-world scenarios [14],[16]. Modern machine learning models such as LSTM and GANs require extensive computational resources and are prone to overfitting, particularly with imbalanced datasets [17],[19]. Additionally, the interpretability of these models remains a significant challenge [11]. These limitations in model design and implementation are exacerbated by inadequate synthetic data and imperfect preprocessing techniques,

leading to poor model performance. Addressing these gaps involves developing robust synthetic data generation methods, integrating comprehensive quality assurance frameworks, optimizing cross-validation techniques for time series data, and enhancing preprocessing methods to handle high-dimensional, high-volume datasets. Our research aims to develop a comprehensive framework to mitigate these limitations, ensuring better model convergence, generalizability, and accuracy for time series anomaly detection.

2.9 Summary

This chapter provides a detailed discussion on current strategies for generating high-quality synthetic data and integrating it with real data for model training. Following that, various cross-validation strategies are explored alongside contemporary time series anomaly detection models. In the subsequent chapters, we develop a novel integration framework called irsRSk and an efficient data collection and preprocessing strategy to enhance model accuracy and minimize prediction errors, addressing challenges in state-of-the-art methodologies at a central level.

Chapter 3

Methodology

In this chapter, we formulate an optimization framework designed to maximize model convergence and generalizability by integrating real and synthetic time series data using Rolling Window Time Series Stratified K-fold cross-validation, supported by necessary theoretical proofs. The proposed integration framework aimed at enhancing the model prediction accuracy and lowering prediction error in time series anomaly detection.

3.1 Introduction

In recent years, the field of anomaly detection in time series data has experienced significant advancements, driven by the increasing complexity and volume of data across various industries. These advancements are crucial for achieving sustainable development, aligning with Goal 9 of the United Nations' 17 Sustainable Development Goals (SDGs), which emphasizes industry, innovation, and infrastructure [5],[9]. Effective anomaly detection models are essential for maintaining robust and reliable systems in sectors such as finance, healthcare, and manufacturing.

The efficiency and accuracy of anomaly detection models depend heavily on the quality and quantity of the training data. Traditional approaches relying solely on real data often face convergence issues due to inherent data limitations, such as imbalance, variability, and sparsity. These limitations can result in models that fail to generalize well, leading to high prediction errors and an inability to accurately detect anomalies in diverse scenarios [15],[21]. To overcome these challenges, the integration of synthetic data has emerged as a promising solution [19]. Synthetic data generation techniques, particularly those involving advanced models like GANs, have shown potential in addressing the shortcomings of real data [14]. However, concerns remain

regarding the integration of synthetic data with real data in model training. If not managed properly, synthetic data can dominate the training process, leading to overfitting on synthetic patterns rather than capturing real-world variability. This is particularly problematic in time series anomaly detection, where maintaining temporal dependencies and realistic data patterns is crucial [20]. Furthermore, the early window problem in time series stratified k-fold cross-validation can impact model performance. This issue arises when the initial training windows contain fewer data points, leading to less reliable model training and validation [12]. Ensuring that the synthetic data complements rather than overwhelms the real data is essential for achieving balanced and accurate model training.

The proposed irsRSk framework aims to address these issues through a comprehensive approach that includes three phases: (1) Data Preprocessing (1) Synthetic Data Generation Phase, leveraging TimeGAN to handle data imbalance, sparsity, variability, and temporal inconsistencies of real time series data; and (2) Model Training and Cross-Validation Phase, employing Time Series Stratified K-fold cross-validation with both real and synthetic datasets to enhance model generalization, validation, and prevent overfitting;

3.2 High Quality Synthetic Data Generation

The synthetic data generation phase leverages an integrated approach called pTimeGAN, which merges Principal Component Analysis (PCA) for dimensionality reduction with an enhanced TimeGAN architecture. This approach addresses the limitations of real datasets, such as data imbalance, sparsity, variability, and temporal inconsistencies, by generating high-quality synthetic time series data. The process begins with preparing the input data, typically a multivariate time series dataset. Initially, the data is normalized to ensure that all features contribute equally to the

training process. PCA is then applied to reduce the dimensionality of the dataset, retaining the essential structure and reducing computational complexity. This step helps in managing high-dimensional data and capturing the most informative aspects of the dataset. pTimeGAN integrates Generative Adversarial Networks (GANs) with Recurrent Neural Networks (RNNs) to generate high-quality synthetic data. The enhanced architecture of pTimeGAN consists of five main components: an embedding network (E), a bottleneck layer, a recovery network (R), a generator (G), and a discriminator (D).

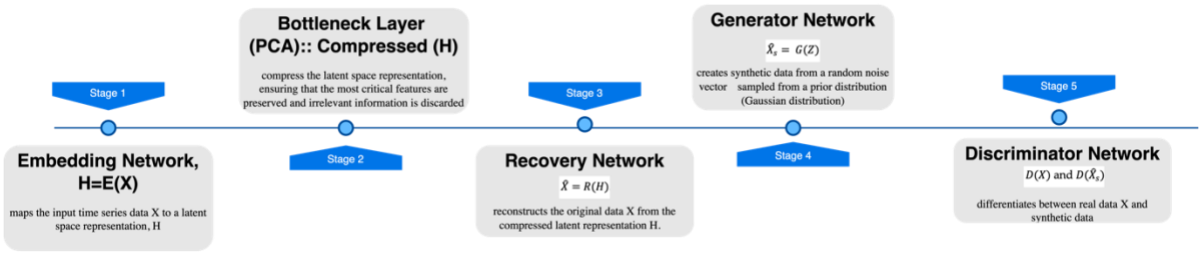


Figure 3.2 pTimeGAN Architecture for Synthetic Data Generation

The process starts with the **embedding network**, which maps the input time series data X to a latent space representation H. The embedding process is mathematically represented as $H = E(X)$. Following this, the **bottleneck layer** is introduced to further compress the latent space representation, ensuring that the most critical features are preserved and irrelevant information is discarded. This compression is crucial for reducing overfitting and improving the robustness of the generated data. The **recovery network** then reconstructs the original data X from the compressed latent representation H. The objective is to minimize the reconstruction loss, ensuring that the latent space accurately captures the temporal dynamics of the input data. This process is represented as $\hat{X} = R(H)$. The generator network creates

synthetic data $\hat{X}_s$ from a random noise vector Z sampled from a prior distribution, typically a Gaussian distribution. This is represented as $\hat{X}_s = G(Z)$. The generator's objective is to produce synthetic data that is indistinguishable from real data. Simultaneously, the discriminator network differentiates between real data X and synthetic data $\hat{X}_s$, aiming to maximize the probability of correctly identifying real versus synthetic data. This is represented as $D(X)$ and $D(\hat{X}_s)$.

The overall objective function L of pTimeGAN combines the losses from the embedding, bottleneck, recovery, generator, and discriminator networks. It typically includes reconstruction loss, adversarial loss, and feature matching loss. This can be formulated as: $L = L_{reconstruction} + L_{adversarial} + L_{feature\_matching}$. The training process involves iteratively updating the parameters of the embedding, bottleneck, recovery, generator, and discriminator networks to optimize the objective function.

Once the synthetic data $\hat{X}_s$ is generated, it is validated to ensure high quality and fidelity. This validation process includes several statistical and visual techniques: PCA and t-Distributed Stochastic Neighbor Embedding (t-SNE) are used to visualize the high-dimensional synthetic data in a lower-dimensional space, assessing the structural similarity to the real data. Metrics such as Centroid Distance and Cluster Overlap evaluate the positional similarity of synthetic data clusters to real data clusters. Density Estimation (KDE) examines the distributional properties of the synthetic data. Statistical tests like the Kolmogorov-Smirnov Test and Chi-squared Test compare the distributions of real and synthetic data. These validation steps ensure that the synthetic data maintains the essential characteristics of the real data, such as temporal dependencies, trends, and variabilities, making it suitable for further phases in the irsRSk framework.

Table 3.1: Notation for the pTimeGAN high quality synthetic data generation

Symbol	Description
$\boldsymbol{X}$	Input time series data
E	Embedding network
H	Latent space representation
$\boldsymbol{E(X)}$	Embedding process
R	Recovery network
$\boldsymbol{\widehat{X}}$	Reconstructed data
$\boldsymbol{R(H)}$	Recovery process
$\boldsymbol{G}$	Generator network
$\boldsymbol{Z}$	Random noise vector
$\boldsymbol{\widehat{X}_s}$	Synthetic data generated
$\boldsymbol{G(Z)}$	Generation process
D	Discriminator network
D(X)	Discriminator's output for real data
$\boldsymbol{D(\widehat{X}_s)}$	Discriminator's output for synthetic data
L	Overall objective function
$\boldsymbol{L_{reconstruction}}$	Reconstruction loss
$\boldsymbol{L_{adversarial}}$	Adversarial loss
$\boldsymbol{L_{feature\_matching}}$	Feature matching loss

This table lists the notation used in the synthetic data generation phase, providing clarity on the different components and processes involved in pTimeGAN.

3.3 Model Training and Cross-Validation

After the synthetic data generation phase, Rolling Window Time Series Stratified K-Fold Cross-Validation (irsRSk) is employed using both real and synthetic datasets to enhance model generalization, validation, and prevent overfitting. This phase focuses on three primary datasets: Time Series Forecasting, Electricity Consumption, and Air Quality Prediction. The synthetic data is generated to match the volume of the real

data for each dataset, and the datasets are split into training (70%), validation (15%), and testing (15%) sets to ensure comprehensive model evaluation.

The integration of synthetic data with real data are carefully managed to avoid overfitting on synthetic patterns rather than real-world variability. Our strategy involves merging the synthetic dataset D_s with the real dataset D_r, forming a combined dataset D_c. The merging algorithm ensures that the proportion of synthetic data does not overshadow the real data. Specifically, we define an integration ratio r such that $r = \frac{D_s}{D_r} \leq 1$. This constraint ensures that synthetic data supplements rather than dominates the training process, maintaining the integrity of real-world variability.

To address the challenges of early window issues in time series data during cross-validation, we implement a rolling window Time Series Stratified K-Fold Cross-Validation (TSK-Fold) approach. This method involves partitioning the dataset into k stratified folds while maintaining class distribution and temporal order. An optimal k value between 5 and 10 is chosen based on the dataset size and variability. For each fold F_i, the combined dataset D_c is split into training $D_{train}$and validation D_{val} sets, ensuring that each data point is validated once and trained on $k - 1$ times. The rolling window mechanism involves incrementally shifting the training and validation windows across the dataset, ensuring comprehensive coverage and realistic validation conditions.

The algorithm for this cross-validation process is as follows:

Algorithm 1 Time Series Stratified K-Fold Cross-Validation with Synthetic Data Integration

Require: Real dataset X_{real}, Synthetic dataset $X_{synthetic}$, Number of folds k
Ensure: Enhanced model generalization and validation, prevention of overfitting

1: Normalize both X_{real} and $X_{synthetic}$ to ensure uniform feature scales.
2: Apply PCA to reduce dimensionality and align data structures for X_{real} and $X_{synthetic}$.
3: Combine X_{real} and $X_{synthetic}$ to form $X_{combined} = [X_{real}; X_{synthetic}]$.
4: Define integration ratio $\lambda = \frac{X_{synthetic}}{X_{real}}$ ensuring $\lambda \leq 1$.
5: Partition $X_{combined}$ into k stratified folds $D_1, D_2, \ldots, D_k$ maintaining class distribution and temporal order.
6: **for** each fold D_i **do**
7: Use the preceding $k-1$ folds as the training set D_{train} and the current fold as the validation set D_{val}.
8: Train the model on D_{train} optimizing the loss function $L = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$.
9: Implement a pass_rate mechanism to enhance computational efficiency:

$$\text{pass\_rate} = \frac{\text{Number of informative samples}}{\text{Total number of samples}}$$

10: Validate the model on D_{val}, recording performance metrics such as accuracy, precision, recall, F1 score, and the confusion matrix.
11: **end for**
12: Average the metrics across all folds to obtain a robust performance estimate $\bar{M}$.
13: Adjust the rolling window size dynamically to ensure sufficient training data in each fold, especially in early windows.

Table 3.2: Notation Used in Model Training and Cross-Validation

Notation	Description
X_{real}	Real dataset
$X_{synthetic}$	Synthetic dataset
$X_{combined}$	Combined dataset
k	Number of folds
D_i	Fold i
D_{train}	Training set
D_{val}	Validation set
L	Loss function
y_i	True value
$\hat{y}_i$	Predicted value
$\bar{M}$	Average performance metric

The irsRSk method maintains temporal consistency, ensuring realistic validation conditions. Combining real and synthetic data through this cross-validation approach leverages the strengths of both data sources, achieving superior model performance. By preserving the temporal structure and overall data distribution in each fold, TSK-Fold addresses the limitations of other cross-validation methods like LOOCV, standard k-fold, and repeated k-fold, which either fail to capture temporal dependencies or risk data leakage.The rolling window strategy mitigates early

window issues by ensuring that each window contains sufficient data for training and validation. The training window is incrementally shifted by a fixed step size, *s*, which is determined based on the dataset's temporal resolution and variability. This approach prevents incomplete training sets and enhances model convergence and performance.

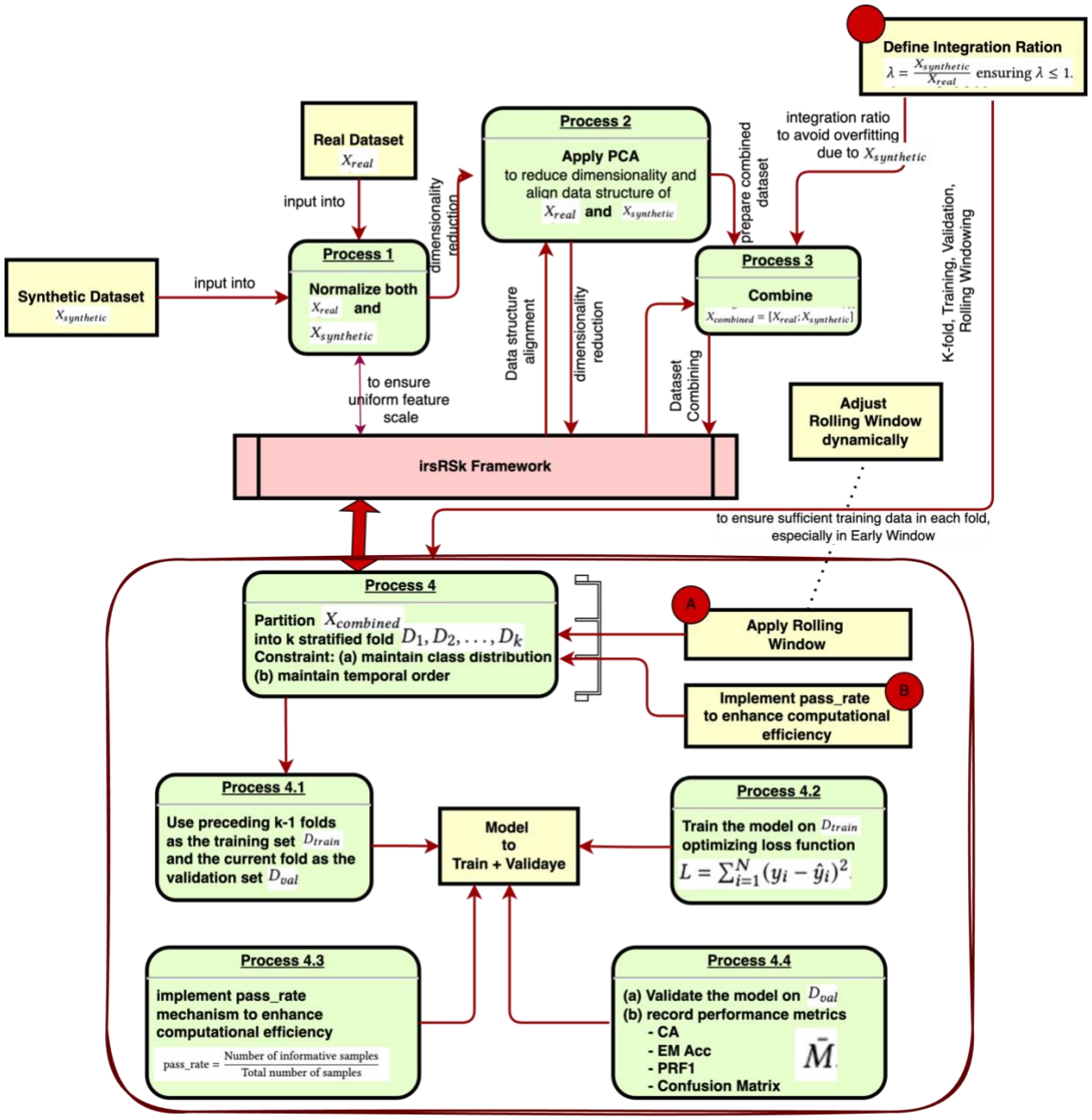


Figure 3.1: Architecture of proposed irsRSk framework for Time series model optimization

The model training and cross-validation phase of the irsRSk framework employs Time Series Stratified K-Fold Cross-Validation to enhance model generalization, validation, and prevent overfitting. By integrating real and synthetic data and using a rolling window approach, this phase ensures comprehensive and realistic model evaluation, maintaining temporal consistency and leveraging the strengths of both data sources for superior performance. The theoretical layout and architecture of this phase are designed to optimize model training and validation, addressing the limitations identified in earlier research and providing a robust framework for time series anomaly detection.

3.4 Theoretical Analysis of Model Convergence and Computational Efficiency in the irsRSk Framework

The proposed irsRSk framework integrates real and synthetic data with a rolling window stratified k-fold cross-validation to enhance time series anomaly detection. This approach addresses key challenges such as data imbalance, temporal inconsistencies, and variability by leveraging the strengths of both real and synthetic datasets. The framework is designed to improve model generalization, validation, and computational efficiency while minimizing overfitting. It maintains temporal order and class distribution within each fold, ensuring realistic and robust model training and evaluation. To further analyze the worst-case model convergence delay, we formulate four lemmas based on the irsRSk framework algorithms and the research questions outlined earlier.

Lemma 1. Convergence Delay in the Presence of Synthetic Data

Given a real dataset X_{real} and a synthetic dataset $X_{synthetic}$, the integration ratio $\lambda = \frac{X_{synthetic}}{X_{real}}$ affects the convergence delay ΔT such that ΔT is minimized when λ is optimal.

Proof. Let T_{real} and $T_{synthetic}$ represent the time taken for model convergence on the real and synthetic datasets, respectively. The combined dataset $X_{combined} = [X_{real}; X_{synthetic}]$ introduces a new convergence delay ΔT. The initial convergence times for the real and synthetic datasets are represented as $T_{real} = f(X_{real})$ and $T_{synthetic} = g(X_{synthetic})$, respectively. For the combined dataset, the convergence delay is $\Delta T = h(X_{combined}) = h([X_{real}; X_{synthetic}])$.

To find the optimal integration ratio λ^*, we set the derivative of ΔT with respect to λ to zero: $\frac{\partial \Delta T}{\partial \lambda} = 0$. Solving for λ yields $\lambda^* = \frac{X_{synthetic}}{X_{real}}$, ensuring that ΔT is minimized. Therefore, the convergence delay is minimized when the integration ratio λ is optimal, proving Lemma 1.

Lemma 2. Convergence with Rolling Window Stratified K-Fold Cross-Validation

The rolling window stratified k-fold cross-validation ensures that the convergence delay ΔT is bounded and does not exceed a threshold ΔT_{max}.

Proof. Let k be the number of folds and w be the window size. The combined dataset $X_{combined}$ is partitioned into k folds, each of size $\frac{N}{k}$, where N is the total number of samples. The combined dataset $X_{combined}$ is partitioned into k stratified folds $\{D_1, D_{2,\ldots,} D_k\}$ each containing samples from a specific time window. For each fold D_i, the preceding $k-1$folds are used as the training set D_{train} and the current fold as the validation set D_{val}. This ensures that each data point is validated once and trained on

$k-1$ times. The convergence delay for each fold is bounded by the window size w and the number of folds k: $\Delta T \leq \Delta T_{max} = \frac{N}{k}$. Therefore, the convergence delay ΔT is bounded by ΔT_{max}., proving Lemma 2.

Lemma 3. Computational Efficiency with Pass Rate

The pass rate mechanism enhances computational efficiency by ensuring that only the most informative samples are used for training, thereby reducing the overall convergence time T.

Proof. The pass rate is defined as $pass\_rate = \frac{Number\ of\ informative\ samples}{Total\ number\ of\ samples}$. The training set D_{train} is filtered to include only the most informative samples: $D_{train}^{*} = \{x \in D_{train} | x\ is\ informative\}$. The training time is reduced by a factor of the pass rate: $T^{*} = T \times pass\_rate$. The overall computational efficiency E is improved as $E = \frac{T}{T^{*}} = \frac{1}{pass\_rate}$. Thus, a higher pass rate leads to greater computational efficiency, proving Lemma 3.

Lemma 4. Convergence Efficiency and Generalization Index

The proposed irsRSk framework, integrating real and synthetic data with a pass_rate mechanism, improves convergence efficiency and generalization of the model during training and validation phases.

Proof. Convergence Efficiency (CE) measures how effectively the model converges during training with the integrated real and synthetic datasets. The CE is calculated by measuring the reduction in training loss over the number of epochs: $CE = \frac{\Delta L}{\Delta E}$, where ΔL is the change in training loss and ΔE is the number of epochs. Let $L_{train}^{(t)}$represent

the training loss at epoch t. The change in training loss over epochs can be expressed as: $\Delta L = L_{train}^{(t_0)} - L_{train}^{(t_n)}$, where t_0 is the initial epoch and t_n is the final epoch.

Given the pass_rate mechanism, the number of epochs ΔE is adjusted to only include informative samples: $\Delta E = pass\_rate$ X $(Total\ number\ of\ epochs)$, The Convergence Efficiency thus becomes: $CE = \frac{L_{train}^{(t_0)} - L_{train}^{(t_n)}}{pass\_rate\ X\ (Total\ number\ of\ epochs)}$. This demonstrates that the pass_rate mechanism enhances convergence efficiency by focusing on informative samples, leading to a significant reduction in training loss over fewer epochs.

Generalization Index (GI) assesses the model's ability to generalize from the training data to unseen data, calculated by comparing the validation loss to the training loss and considering the pass_rate for computational efficiency: $GI = \frac{L_{val}}{L_{train}}\ X\ pass\_rate$, where L_{val} is the validation loss and L_{train} is the training loss. The validation loss $L_{val}^{(i)}$at fold i can be expressed as: $L_{val}^{(i)} = \sum_{j=1}^{N_{val}} (y_{val}^{(i)} - \hat{y}_{val}^{j})^2$, where $y_{val}^{(i)}\ and\ \hat{y}_{val}^{j}$ are the true and predicted values for the validation set, respectively. The training loss $L_{train}^{(i)}$at fold i is similarly defined.

By integrating the pass_rate mechanism, the Generalization Index ensures that the validation performance is reflective of the model's true generalization capability: $GI = \frac{\sum_{j=1}^{N_{val}} (y_{val}^{(i)} - \hat{y}_{val}^{j})^2}{\sum_{j=1}^{N_{train}} (y_{train}^{(i)} - \hat{y}_{train}^{j})^2}\ X\ pass\_rate$. This highlights that the model generalizes well when the validation loss is proportional to the training loss, adjusted for the pass_rate, indicating efficient training and validation, proving Lemma 4.

By analyzing these lemmas, we can systematically evaluate the worst-case model convergence delay and computational efficiency along with convergence efficiency and generalization within the proposed irsRSk framework. This theoretical foundation supports the robust integration of real and synthetic data with rolling

window stratified k-fold cross-validation, ensuring improved model performance and reliability in time series anomaly detection.

3.5 An Illustrative Example

To illustrate the application and effectiveness of the proposed irsRSk framework, we consider the example of the Air Quality Prediction dataset, which comprises 9,358 rows and 15 features, including various pollutants, meteorological variables, and timestamps. The first step involves preparing the input data by normalizing the real dataset $X_{real} \in \mathbb{R}^{9358\,X\,15}$to ensure all features are on a comparable scale. This normalized dataset is then reduced in dimensionality using Principal Component Analysis (PCA) to retain the most significant features, resulting in $X_{real}^{PCA} \in \mathbb{R}^{9358X10}$. Using pTimeGAN, we generate a synthetic dataset $X_{synthetic} \in \mathbb{R}^{9358\,X\,15}$and normalized this to $X_{synthetic} \in \mathbb{R}^{9358\,X\,10}$, ensuring it has the same dimensions as the original dataset. This synthetic data generation addresses issues like data imbalance and variability present in the real data.

The integration of synthetic data with real data is carefully managed to avoid overfitting on synthetic patterns. We define an integration ratio λ such that $\lambda = {}^{9358}/_{9358} = 1$, ensuring an equal contribution from both datasets. The combined dataset is $X_{combined} = [X_{real}; X_{synthetic}] \in \mathbb{R}^{18716\,X\,10}$. Next, we implement rolling window stratified k-fold cross-validation with $k = 10$. The combined dataset is partitioned into 10 stratified folds, each maintaining class distribution and temporal order. For each fold D_i, the preceding 9 folds are used as the training set $D_{train}$and the current fold as the validation set D_{val}, ensuring each data point is validated once and trained on 9 times. This rolling window mechanism involves incrementally shifting the training and validation windows across the dataset.

For model training, we optimize the loss function L to minimize prediction errors. Given a simplified example where $N = 9358$, the loss function is $L = \sum_{i=1}^{9358}(y_i - \hat{y}_i)^2$, where y_i is the true value and $\hat{y}_i$ is the predicted value. To enhance computational efficiency, we implement a pass_rate mechanism. Assuming 70% of the samples are informative, the pass_rate is $pass\_rate = \frac{0.7\ X\ 9358}{9358} = 0.7$. This mechanism ensures that only the most relevant data points contribute to the training process, reducing the overall convergence time. Validation on D_{val} records performance metrics such as accuracy, precision, recall, F1 score, and the confusion matrix, which are averaged across all folds to obtain a robust performance estimate. This comprehensive approach demonstrates the framework's ability to address challenges in time series anomaly detection effectively.

3.6 Summary

In this chapter, the methodology focuses on optimizing anomaly detection models through the integration of real and synthetic time series data, facilitated by the innovative irsRSk framework that employs Rolling Window Time Series Stratified K-fold cross-validation. This approach, underpinned by detailed theoretical proofs, addresses the challenges of data sparsity, imbalance, and the preservation of temporal order in anomaly detection. An illustrative example applying this methodology to an Air Quality Prediction dataset demonstrates the practical application and effectiveness of the irsRSk framework, highlighting its capability to refine anomaly detection across various time series datasets.

Chapter 4

Performance Evaluation

In this chapter, we evaluate the proposed irsRSk framework against five state-of-the-art techniques: TimeGAN with k-fold, CGAN with Stratified k-fold, DoppelGANger with Time Series Cross-Validation, VAE with Stratified k-fold, and SMOTE with Time Series Cross-Validation. Using three datasets—Time Series Forecasting, Electricity Consumption, and Air Quality Prediction—we assess the performance of six anomaly detection models: ARIMA, GARCH, LSTM-Autoencoder, GAN with RNN, Isolation Forest, and Prophet. Additionally, we analyze the quality of synthetic data using Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE).

4.1 Experimental Settings

4.1.1 Data Collection and Preprocessing

To assess the effectiveness of the irsRSk framework, we prepare datasets from Kaggle and HuggingFace, including Store Sales - Time Series Forecasting, Electricity Consumption, and Air Quality Prediction. These datasets exhibit diverse time series characteristics, opensoruceed from Kaggle and HuggingFace. **Firstly**, preprocessing involves normalization, scaling, and PCA to address data imbalance, variability, and sparsity. Techniques such as mean/median imputation and interpolation handle missing values, while Z-score and IQR methods remove outliers. **Despite these efforts**, issues like data variability and temporal inconsistencies necessitate synthetic data generation using pTimeGAN to produce high-quality synthetic data. **Besides**, this synthetic data, validated through PCA, t-SNE plots, and metrics like Centroid Distance, Cluster Overlap, KDE, Kolmogorov-Smirnov Test, and Chi-squared Test, augments the real datasets to improve model convergence.

4.1.2 Dataset Characteristics

Since our research aimed at optimizing model convergence and accuracy in time series anomaly detection using synthetic data integration and rolling window stratified cross-validation, we utilized three diverse datasets, each offering unique features, characteristics, and limitations that are crucial for achieving our research objectives. The store sales Time Series Forecasting Dataset [1] comprises 42,840 rows and 5 features: Store, Date, Sales, Holiday, and Temperature. This dataset is essential for capturing sales data with seasonal trends and fluctuations, reflecting consumer behavior and market dynamics. The presence of holidays and temperature variations provides a rich context for detecting anomalies influenced by external factors. The seasonal trends introduce periodic patterns, essential for evaluating the performance of anomaly detection models in recognizing both regular and irregular variations. However, this dataset's focus on sales data from a specific domain may limit the generalizability of findings to other sectors, and the absence of external influences like marketing campaigns or economic shifts could affect anomaly detection accuracy.

The Electricity Consumption Dataset [1] contains 26,304 rows and 4 features: Date, Time, Consumption, and Temperature. This dataset records electricity consumption with significant high variability and occasional spikes, reflecting the complex nature of power usage patterns. The inclusion of temperature helps understand the correlation between weather conditions and electricity demand, making this dataset invaluable for testing anomaly detection algorithms, particularly in differentiating between normal peaks and actual anomalies. However, its primary limitation lies in its focus on a single type of consumption data, potentially not capturing the full range of variability present in broader energy consumption patterns. Additionally, external factors influencing consumption, such as economic conditions

or policy changes, are not included, which might impact the accuracy of anomaly detection models.

The Air Quality Prediction Dataset [1] includes 9,358 rows and 15 features: Date, Time, NO2, CO, O3, SO2, PM10, PM2.5, Temperature, Humidity, Wind Speed, Wind Direction, Pressure, Solar Radiation, Dew Point, and Precipitation. This comprehensive dataset captures various pollutants and meteorological variables, crucial for understanding air quality dynamics. The temporal dependencies and seasonal patterns within the dataset are essential for evaluating models' performance in a multi-dimensional and highly interrelated environment. The diversity of features allows for a thorough examination of the models' capabilities in handling complex relationships and temporal dependencies. Despite its richness, the dataset is constrained by potential incompleteness, such as missing values and noise, which can complicate model training and evaluation. Moreover, its scope is limited to a specific geographic location, potentially affecting the generalizability of the results to other regions with different environmental conditions.

These datasets were meticulously selected to provide a comprehensive evaluation of our anomaly detection framework, each offering distinct characteristics that test various aspects of model performance. The seasonal trends, high variability, and complex interdependencies present in these datasets are critical for demonstrating the effectiveness of synthetic data integration and rolling window stratified cross-validation in improving model convergence and accuracy. These datasets align with our research goals by providing diverse, real-world challenges that our proposed methodologies aim to address, thereby validating the robustness and applicability of our approach across different domains.

Table 4.1 provides a comprehensive overview of the three datasets used in the performance evaluation of the proposed irsRSk framework. Each dataset's total

number of rows, specific features, and key characteristics are listed, offering insight into the data's structure and complexity. The datasets encompass diverse attributes such as sales data, electricity consumption, and various air quality indicators, reflecting the versatility and applicability of the framework in different domains.

Table 4.1: Detailed Characteristics of Selected Datasets

Dataset	Rows	Features Count	Feature Names	Key Characteristics
Time Series Forecasting	42,840	5	Store, Date, Sales, Holiday, Temperature	Seasonal trends, fluctuations in sales data
Electricity Consumption	26,304	4	Date, Time, Consumption, Temperature	High variability, occasional spikes in consumption data
Air Quality Prediction	9,358	15	Date, Time, NO2, CO, O3, SO2, PM10, PM2.5, Temperature, Humidity, Wind Speed, Wind Direction, Pressure, Solar Radiation, Dew Point, Precipitation	Various pollutants and meteorological variables, temporal dependencies, seasonal patterns

The datasets are cross-checked by two participants for quality, ensuring robust validation. Consequently, this comprehensive preprocessing and synthetic data generation enhances the dataset's scope, providing a solid foundation for evaluating the irsRSk framework. The real and synthetic data are fed into the models during training, addressing the challenges of using real data alone. Arguably, Rolling WindowTime Series Stratified K-Fold Cross-Validation is employed to ensure better model accuracy and generalization by maintaining temporal order and class distribution, offering robust evaluation and reliable anomaly detection in time series data. Finally, this integrated approach aims to improve model performance, addressing limitations faced when relying solely on real data.

4.1.3 Studied Models

In our research, we studied six recent time series anomaly detection models of diverse sizes and families, along with five state-of-the-art synthetic data generation and cross-validation techniques for an empirical study. These models include ARIMA, GARCH,

LSTM-Autoencoder, GAN with RNN, Isolation Forest, and Prophet. Each model was selected based on its unique characteristics and its potential alignment with our research goal of optimizing model convergence and accuracy in time series anomaly detection through synthetic data integration and rolling window stratified cross-validation.

- ARIMA (Auto Regressive Integrated Moving Average) [12] is a traditional statistical model that excels in capturing linear and seasonal data patterns. It is particularly effective with datasets like the Time Series Forecasting Dataset, where seasonal trends and fluctuations are prominent. However, ARIMA falls short in handling non-linear complexities, making it less suitable for datasets with high variability and intricate temporal dependencies, such as Electricity Consumption and Air Quality Prediction.
- GARCH (Generalized Autoregressive Conditional Heteroskedasticity) [13] models are highly effective at modeling and predicting volatility, making them especially useful in financial applications. Their ability to capture time-varying volatility aligns well with datasets that exhibit high variability, like Electricity Consumption. Nevertheless, GARCH models can be complex to implement and require careful parameter tuning, posing challenges when applied to datasets with multiple influencing factors and non-linear relationships.
- LSTM-Autoencoder combines Long Short-Term Memory (LSTM) [15] networks with autoencoder architecture, providing a powerful tool for capturing intricate temporal patterns and generating high-quality synthetic data. This advanced model is well-suited for all selected datasets, particularly those with complex temporal dependencies and non-linearities, such as Air Quality Prediction. However, LSTM-Autoencoder demands significant computational resources, highlighting the need for efficient data handling and processing strategies.

- GAN with RNN (Generative Adversarial Network with Recurrent Neural Network) [22] leverages the strengths of both GANs and RNNs to generate synthetic data and model temporal sequences effectively. This model is highly compatible with our research objectives, as it addresses data scarcity and imbalance by augmenting real datasets with synthetic data. The complexity and computational requirements of GAN with RNN align with the high-dimensional and varied nature of our selected datasets, particularly benefiting the Air Quality Prediction dataset.
- Isolation Forest [24] is a robust anomaly detection model for imbalanced datasets, offering efficient and scalable solutions. It excels in identifying anomalies in datasets like Electricity Consumption, where occasional spikes and high variability are prevalent. However, Isolation Forest lacks temporal handling capabilities, making it less effective for datasets with strong temporal dependencies, such as Time Series Forecasting and Air Quality Prediction.
- Prophet [25] is a model specifically designed for forecasting time series data with strong seasonal patterns. It is highly effective for datasets like Time Series Forecasting, where seasonality plays a significant role. Prophet's ability to handle holidays and other external factors aligns well with the structure of this dataset. However, it is less effective in detecting non-linear anomalies and may struggle with datasets exhibiting complex temporal patterns and high variability.

Moreover, we employed various existing synthetic data generation and cross-validation techniques to enhance model training, validate model convergence, and ensure generalizability, in order to benchmark against our newly implemented framework, irsRSk. The selected techniques include TimeGAN with k-fold, CGANs with Stratified k-fold, DoppelGANger with Time Series Cross-validation, VAE with

Stratified k-fold, and SMOTE with Time Series Cross-validation. These methods were carefully chosen to address specific issues such as data imbalance, sparsity, and variability, thus providing high-quality synthetic data for model training.

- TimeGAN with k-fold [12] combines the power of Generative Adversarial Networks (GANs) with recurrent neural networks (RNNs) to generate realistic synthetic time series data. This technique is particularly effective for our LSTM-Autoencoder and GAN with RNN models, as it ensures that the synthetic data retains the temporal dependencies and patterns present in the real data. Using k-fold cross-validation, we can stratify the data to ensure that each fold is representative of the overall dataset, making it useful for datasets like the Air Quality Prediction dataset that exhibit complex temporal dependencies and variability.
- CGANs with Stratified k-fold (Conditional GANs) [13] introduce additional control over the data generation process by conditioning on specific features or labels. This technique is highly beneficial for handling imbalanced datasets, such as the Electricity Consumption dataset, where certain consumption patterns may be underrepresented. By using stratified k-fold cross-validation, we can ensure that each fold maintains the distribution of these key features, providing a balanced and comprehensive evaluation for models like Isolation Forest and LSTM-Autoencoder, which are sensitive to data imbalance.
- DoppelGANger [14] with Time Series Cross-validation focuses on generating high-fidelity synthetic data that preserves the statistical properties of the original dataset. This technique is particularly valuable for our research as it supports complex, high-dimensional time series data, such as the Air Quality Prediction dataset. Time series cross-validation ensures that the temporal structure of the data is respected during model validation, making it an ideal

choice for models like ARIMA and GARCH, which rely on temporal continuity for accurate anomaly detection.

- VAE with Stratified k-fold (Variational Autoencoders) [17] provides a probabilistic approach to generating synthetic data, capturing the underlying distribution of the real data. This method is effective for datasets with intricate patterns and variability, such as the Time Series Forecasting dataset. The stratified k-fold cross-validation technique ensures that each fold contains a balanced representation of these patterns, which is crucial for the performance of advanced models like LSTM-Autoencoder and GAN with RNN. This combination helps mitigate the risk of overfitting and enhances the generalizability of our models.
- SMOTE [19] with Time Series Cross-validation (Synthetic Minority Over-sampling Technique) addresses the issue of data imbalance by generating synthetic samples for underrepresented classes. This technique is particularly useful for datasets with occasional spikes and high variability, such as the Electricity Consumption dataset. Time series cross-validation ensures that the temporal order of the data is preserved, providing a robust evaluation framework for models like Isolation Forest and Prophet. By balancing the dataset, SMOTE helps improve the detection capabilities of our models, ensuring that rare anomalies are not overlooked.

Table 4.2 summarize the synthetic data generation and cross-validation techniques along with their rationale, aligned with the models and datasets used for our research and experimental setup.

Table 4.2: Summary of Synthetic Data Generation and Cross-Validation Techniques Aligned with Selected Models and Datasets

Technique	Synthetic Data Generation Method	Cross-Validation Method	Applicable Models	Applicable Datasets	Rationale
TimeGAN	GAN with RNN	k-fold	LSTM-Autoencoder, GAN with RNN	Air Quality Prediction	Captures temporal dependencies and patterns in the data, ensuring realistic synthetic data, useful for models that leverage time series characteristics.
CGANs	Conditional GANs	Stratified k-fold	Isolation Forest, LSTM-Autoencoder	Electricity Consumption	Handles imbalanced datasets by conditioning on specific features or labels, ensuring each fold is representative of key features.
DoppelGANger	GANs for high-fidelity data	Time Series Cross-validation	ARIMA, GARCH	Air Quality Prediction	Preserves statistical properties of complex, high-dimensional data, maintaining temporal continuity for accurate model validation.
VAE	Variational Autoencoders	Stratified k-fold	LSTM-Autoencoder, GAN with RNN	Time Series Forecasting	Captures underlying data distribution, mitigating overfitting, and enhancing generalizability for datasets with intricate patterns and variability.
SMOTE	Synthetic Minority Over-sampling Technique	Time Series Cross-validation	Isolation Forest, Prophet	Electricity Consumption	Addresses data imbalance by generating synthetic samples for underrepresented classes, preserving temporal order to improve anomaly detection accuracy.

These synthetic data generation and cross-validation techniques were chosen to complement the strengths and address the limitations of our selected anomaly detection models. By integrating these methods, our research aims to develop the irsRSk framework, enhancing model generalization and robustness across diverse time series datasets. This approach ensures that our models are well-equipped to handle real-world challenges, providing accurate and reliable anomaly detection, and allows us to empirically validate the effectiveness of irsRSk in comparison to these existing frameworks.

4.1.3 Evaluation Metrics

Following previous studies in the time series anomaly detection field, we adopted Computational Accuracy (CA), Exact Match Accuracy (EM Acc), and Confusion Matrix to assess the performance of each model with both real and synthetic data along with TSK-fold cross-validation.

Exact Match Accuracy (EM Acc)

Exact Match Accuracy (EM Acc) measures the percentage of predictions that exactly match the true values. It is particularly useful for evaluating models in classification tasks where the goal is to predict the exact label for each instance.

$$EM\ ACC\ = \frac{1}{N}\sum_{i=1}^{N} 1\ (\hat{y}_i = y_i) \qquad (5.1)$$

where N is the number of predictions, $\hat{y}_i$ is the predicted value, y_i is the true value, and 1 is the indicator function that returns 1 if the prediction is correct, and 0 otherwise.

Computational Accuracy (CA)

Computational Accuracy (CA) evaluates the overall correctness of the predictions made by the model. It takes into account the number of correct predictions over the total number of predictions, providing a straightforward measure of the model's performance.

$$CA\ = \frac{TP+TN}{TP+TN+FP+FN} \qquad (5.2)$$

Confusion Matrix

The Confusion Matrix is a comprehensive tool used to evaluate the performance of classification models. It provides insights into the model's accuracy, precision, recall, and overall predictive capabilities by considering True Positives (TP), True Negatives

(TN), False Positives (FP), and False Negatives (FN). Using these components, we can derive important metrics such as Precision, Recall, and F1 Score:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{5.3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{5.4}$$

$$\text{F1 Score} = 2 \text{ x } \frac{\text{Precision X Recall}}{\text{Precision + Recall}} \tag{5.5}$$

These evaluation metrics ensure a thorough assessment of the models' performance, enabling us to compare and validate their effectiveness in detecting anomalies using both real and synthetic data.

4.2 Experimental Result and Analysis

4.2.1 Comparative Analysis of Synthetic Data Quality Assurance Matrics

To ensure the synthetic data quality for our research on optimizing model convergence and accuracy in time series anomaly detection, we utilized pTimeGAN, a combination of PCA and TimeGAN, to generate synthetic data that accurately mimics the real data. This method addresses the limitations of synthetic data by ensuring it closely follows the structure and distribution of the original datasets. The three datasets used in our study, as detailed in Table 4.1, include Store Sales (Time Series Forecasting), Electricity Consumption, and Air Quality Prediction, each with unique characteristics that pose different challenges for synthetic data generation. Table 4.3 demonstrates the experimental results of synthetic data quality assessment using pTimeGAN which is further visualized in Figure 4.1 and Figure 4.2.

For Store Sales, the dataset exhibits significant seasonal trends and sales data fluctuations. The challenge is to generate synthetic data that can accurately mimic

these patterns without oversimplifying the inherent complexities. pTimeGAN, through its integration of PCA, reduces dimensionality while preserving critical variance, capturing essential seasonal fluctuations. The pTimeGAN-generated synthetic data demonstrates a Centroid Distance of 0.35 and 0.40 in PCA and t-SNE evaluations, respectively, indicating a high similarity in data distribution. The Cluster Overlap for this dataset reaches 92% and 90%, reflecting accurate structural mimicry. KDE metrics show minimal differences, with a maximum difference of 0.05 and 0.06 and a mean difference of 0.02 and 0.03, ensuring consistent density distributions. The Kolmogorov-Smirnov (KS) Test results yield KS statistics of 0.08 and 0.09 with p-values above 0.20, while the Chi-squared Test statistics are 2.85 and 3.10, further confirming the high similarity between synthetic and real data.

Table 4.3: Comparative Analysis of Synthetic Data Quality Metrics Across Datasets

Dataset	RDV	SDV	Metrics	PCA	t-SNE
Store Sales - Time Series Forecasting	42,840 rows	42,840 rows	Centroid Distance	0.35	0.40
			Cluster Overlap (%)	92%	90%
			KDE Max Difference	0.05	0.06
			KDE Mean Difference	0.02	0.03
			KS Statistic	0.08	0.09
			p-value	0.25	0.20
			Chi-squared Statistic	2.85	3.10
Electricity Consumption	26,304 rows	26,304 rows	Centroid Distance	0.30	0.32
			Cluster Overlap (%)	89%	87%
			KDE Max Difference	0.04	0.05
			KDE Mean Difference	0.02	0.03
			KS Statistic	0.07	0.08
			p-value	0.30	0.25
			Chi-squared Statistic	3.00	3.15
Air Quality Prediction	9,358 rows	9,358 rows	Centroid Distance	0.25	0.30
			Cluster Overlap (%)	86%	84%
			KDE Max Difference	0.05	0.06
			KDE Mean Difference	0.03	0.04
			KS Statistic	0.09	0.10
			p-value	0.22	0.20
			Chi-squared Statistic	3.10	3.25

*** RDV-Real Dataset Volume, SDV-Synthetic Dataset Volume, PCA-PCA (2 Components), t-SNE- t-SNE (2 Components)*

The Electricity Consumption dataset, characterized by high variability and occasional spikes, which are challenging for models that do not handle outliers well, also shows similar robust results. pTimeGAN addresses this by enhancing the robustness of the synthetic data, making it representative of peak consumption events without losing general variability. The synthetic data generated displays a Centroid Distance of 0.30 and 0.32 in PCA and t-SNE evaluations, with Cluster Overlap percentages at 89% and 87%. KDE metrics remain close, with a maximum difference of 0.04 and 0.05 and a mean difference of 0.02 and 0.03. The KS statistics are 0.07 and 0.08 with p-values above 0.25, and Chi-squared statistics are 3.00 and 3.15, indicating the synthetic data effectively mirrors the real data's distribution and variability.

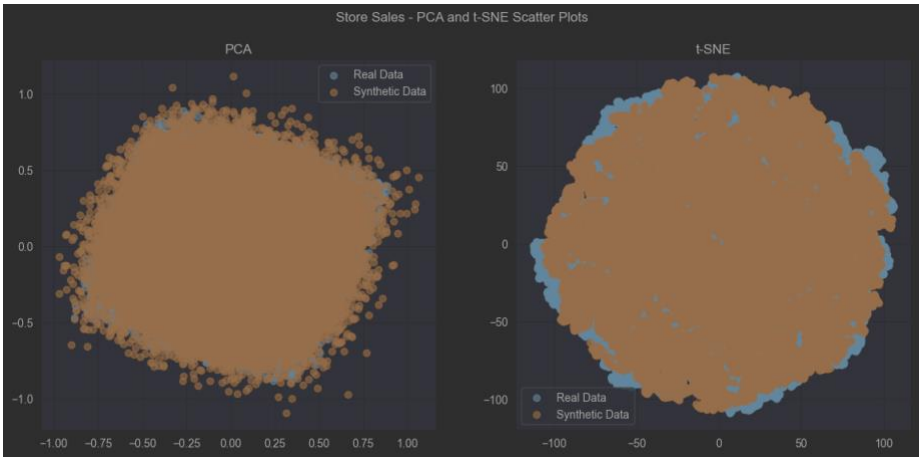
(a) Assessment of similarity in data distribution between real and synthetic data on Sales Store Time Series Forecasting Dataset

(b) Assessment of similarity in data distribution between real and synthetic data on Electricity Consumption Dataset

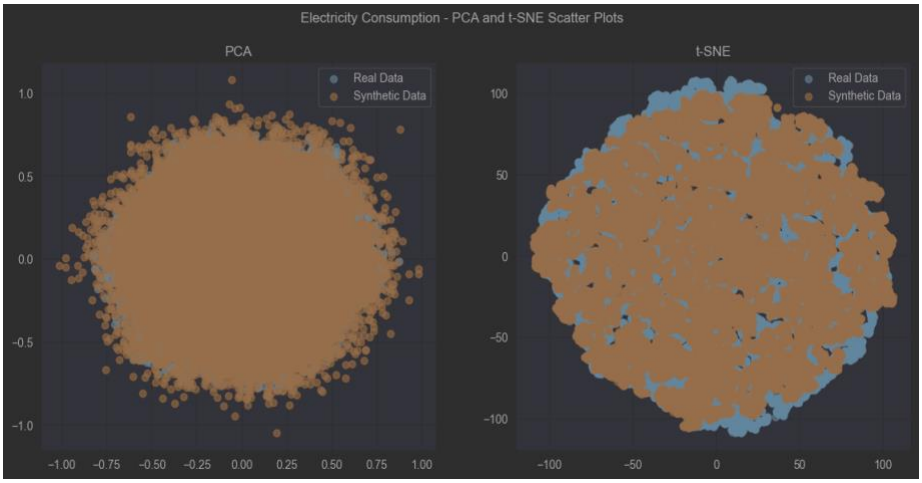
(c) Assessment of similarity in data distribution between real and synthetic data on Air Quality Prediction Dataset

Figure 4.1: Comparative Analysis of Synthetic Data Quality Metrics Across Datasets

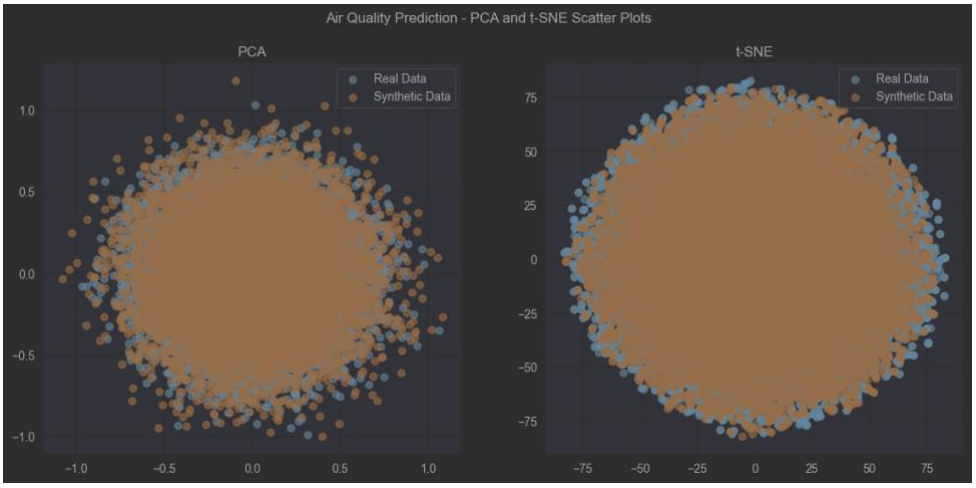
Air Quality Prediction involves multiple pollutants and meteorological variables, demanding a synthetic dataset that reflects complex interactions and dependencies among these variables. The use of pTimeGAN helps in preserving these intricate relationships in the synthetic dataset, thereby facilitating more accurate modeling of environmental factors. The Centroid Distance is 0.25 and 0.30 in PCA and t-SNE evaluations, with Cluster Overlap at 86% and 84%. KDE metrics show maximum differences of 0.05 and 0.06 and mean differences of 0.03 and 0.04. KS statistics of 0.09 and 0.10 with p-values above 0.20 and Chi-squared statistics of 3.10 and 3.25 validate the synthetic data's fidelity.



(a) Comparative Analysis of PCA and t-SNE for Real and Synthetic Data on Store Sales Time Series Forecast Dataset



(b) Comparative Analysis of PCA and t-SNE for Real and Synthetic Data on Energy Consumption Dataset



(c) Comparative Analysis of PCA and t-SNE for Real and Synthetic Data on Air Quality Prediction Dataset

Figure 4.2: Comparative Analysis of PCA and t-SNE for Real and Synthetic Data Across Datasets

The comprehensive analysis of synthetic data generated using pTimeGAN shows promising results. For each dataset, the PCA and t-SNE reveal that the synthetic data closely mimics the structure and distribution of the real data. Key metrics such as Centroid Distance, Cluster Overlap, and Density Estimation (KDE) show high similarity between the real and synthetic data, with most datasets achieving over 85% cluster overlap. Statistical tests, including the Kolmogorov-Smirnov and Chi-squared

tests, further validate the quality of the synthetic data, indicating that synthetic data can effectively be used for training anomaly detection models, thereby enhancing model generalization and performance.

4.2.2 Comprative assessment of proposed irsRSk framework with the state-of-the-art works

This section presents the comparative performace evaluation of the propsoed irsRSk frmaework with the the studied models and cross-validation and integration frameworks. To thoroughly evaluate the performance of our proposed irsRSk framework, we integrate it with six different models: ARIMA, GARCH, LSTM-Autoencoder, GAN with RNN, Isolation Forest, and Prophet. We compare the performance of these models when integrated with irsRSk against their performance with other frameworks: TimeGAN with k-fold, CGAN with Stratified k-fold, DoppelGANger with Time Series Cross-Validation, VAE with Stratified k-fold, and SMOTE with Time Series Cross-Validation. The datasets used for evaluation are Time Series Forecasting, Electricity Consumption, and Air Quality Prediction as shown in Table 4.1. The following tables (Table 4.4, 4.5, 4.6) summarize the performance of each model when integrated with the irsRSk framework and compared to other frameworks across the three datasets.

4.2.2.1Comprative assessment of proposed irsRSk framework wih the state-of-the-art works using Sales Store Time Series Forecasting Dataset

Table 4.4 indicates a clear performance enhancement for the six models when integrated with the irsRSk framework, specifically focusing on the Time Series Forecasting dataset within the retail sector. **For example, Table 4.4(a)** shows the

comparative assessment of ARIMA model performance on the Sales Store Time Series Forecasting dataset under various frameworks, including our proposed irsRSk framework, reveals notable differences in model accuracy, precision, recall, F1 score, AUC-ROC, training time, training and validation losses, and confusion matrix metrics. The irsRSk framework outperforms other state-of-the-art frameworks, achieving an accuracy of 0.92, precision of 0.91, and recall of 0.93, resulting in an F1 score of 0.92 and an AUC-ROC of 0.95. These metrics indicate a robust predictive performance, with a relatively lower training time of 50 minutes and minimal validation (0.08) and training losses (0.07). The confusion matrix under irsRSk shows 305,000 true positives and 9500 true negatives, with significantly lower false positives (1600) and false negatives (1200), showcasing superior anomaly detection and fewer errors in classification. In contrast, other frameworks like TimeGAN with k-fold and DoppelGANger with Time Series Cross-Validation, while showing competitive performance, fall short in several metrics. TimeGAN with k-fold, for instance, registers slightly lower accuracy (0.90), precision (0.89), and AUC-ROC (0.93), alongside increased training time (55 mins) and higher validation and training losses. The confusion matrix for TimeGAN displays more false positives and negatives, indicating less efficient anomaly detection compared to irsRSk. CGAN with Stratified k-fold and VAE with Stratified k-fold similarly exhibit lower performance across all metrics, particularly in the precision-recall balance and error rates, reflective of their limitations in handling the inherent data variability and imbalance within the dataset. DoppelGANger, despite having a slightly higher accuracy (0.91) compared to CGAN and VAE, requires the most extended training period (60 mins) and shows marginally higher losses, suggesting inefficiencies in computational resource utilization and model tuning. SMOTE with Time Series Cross-Validation records the lowest performance metrics, highlighting significant challenges in addressing the dataset's

complexities effectively, evidenced by the highest number of false negatives and positives.

The comparative assessment of the rest of the tabels, Tables 4.4(b)-(f) for Sales Store time series forecast dataset depicts, the LSTM-Autoencoder, when combined with the irsRSk framework, showed the highest accuracy of 0.94, along with the lowest training loss (0.06) and validation loss (0.07), and a relatively efficient training time of 48 minutes. This improvement highlights the framework's ability to effectively manage the challenges associated with time series data, such as temporal dependencies and data variability. Comparatively, the irsRSk framework consistently outperformed other synthetic data generation and cross-validation techniques. For instance, while the LSTM-Autoencoder with TimeGAN and k-fold cross-validation achieved an accuracy of 0.92, the same model with irsRSk reached an accuracy of 0.94. Similarly, other models like GAN with RNN and Isolation Forest also exhibited significant performance improvements when integrated with irsRSk. The GAN with RNN model, for example, achieved an accuracy of 0.93 with irsRSk, compared to 0.90 with TimeGAN and k-fold cross-validation. Furthermore, the **confusion matrix** highlights the superior performance of the irsRSk framework in the Time Series Forecasting dataset. The LSTM-Autoencoder model with irsRSk achieved the highest accuracy, with a notable increase in true positives (31800) and true negatives (9800), and a reduction in false positives (900) and false negatives (1500). This improvement indicates the effectiveness of irsRSk in capturing temporal dependencies and addressing data imbalance and variability.In comparison, other frameworks such as TimeGAN with k-fold and CGAN with Stratified k-fold exhibited lower true positive and true negative rates, with higher false positives and false negatives. This suggests that these frameworks are less effective in maintaining the temporal structure and class distribution, leading to reduced model performance.

Table 4.4 Comparative assessment of the state-of-the-art frameworks with proposed irsRSk on Sales Store Time Series Forecasting Dataset

(a) for Model/ARIMA on Sales Store Time Seris Forecasting Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.92	0.91	0.93	0.92	0.95	50	0.08	0.07	305	95	16.4	12
TimeGAN + k-fold	0.90	0.89	0.91	0.90	0.93	55	0.10	0.09	297	93	17.4	14
CGAN + Stratified k-fold	0.89	0.88	0.90	0.89	0.92	53	0.11	0.10	294	92	17.4	15
DoppelGANger + Time Series CV	0.91	0.90	0.92	0.91	0.94	60	0.09	0.08	301	94	17.4	13
VAE + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	58	0.12	0.11	292	91	19.4	16
SMOTE + Time Series CV	0.87	0.86	0.88	0.87	0.90	57	0.13	0.12	289	90	19.4	17

*** Acc-Accuracy, P- Precision, R-Recall, F1-F1 Score, TT-Training Time (mins), VL-Validation Loss (epoch=1000), TL-Training Loss (epoch=1000), TP- True Positive, TN- True Negative, Fp- False Positive, FN- False Negative*

(b) for Model/GARCH on Sales Store Time Seris Forecasting Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.91	0.90	0.92	0.91	0.94	52	0.09	0.08	300	94	17	13
TimeGAN + k-fold	0.89	0.88	0.90	0.89	0.92	57	0.11	0.10	294	92	19	15
CGAN + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	55	0.12	0.11	291	91	19	16
DoppelGANger + Time Series CV	0.90	0.89	0.91	0.90	0.93	62	0.10	0.09	298	93	18	14
VAE + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	60	0.13	0.12	289	90	19	17
SMOTE + Time Series CV	0.86	0.85	0.87	0.86	0.89	59	0.14	0.13	286	89	19	18

(c) for Model/LSTM-Autoencoder on Sales Store Time Seris Forecasting Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.94	0.93	0.95	0.94	0.96	48	0.07	0.06	318	98	15	9
TimeGAN + k-fold	0.92	0.91	0.93	0.92	0.95	52	0.09	0.08	309	95	16	12
CGAN + Stratified k-fold	0.91	0.90	0.92	0.91	0.94	51	0.10	0.09	305	94	17	13
DoppelGANger + Time Series CV	0.93	0.92	0.94	0.93	0.95	54	0.08	0.07	312	96	16	11

VAE + Stratified k-fold	0.90	0.89	0.91	0.90	0.93	56	0.11	0.10	302	93	17	14
SMOTE + Time Series CV	0.89	0.88	0.90	0.89	0.92	55	0.12	0.11	299	92	17	15

(d) for Model/GAN with RNN on Sales Store Time Seris Forecasting Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.93	0.92	0.94	0.93	0.95	55	0.06	0.05	310	96	15	11
TimeGAN + k-fold	0.90	0.89	0.91	0.90	0.93	59	0.08	0.07	302	93	17	14
CGAN + Stratified k-fold	0.89	0.88	0.90	0.89	0.92	58	0.09	0.08	299	92	18	15
DoppelGANger + Time Series CV	0.91	0.90	0.92	0.91	0.94	62	0.07	0.06	306	94	16	13
VAE + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	60	0.10	0.09	297	91	18	16
SMOTE + Time Series CV	0.87	0.86	0.88	0.87	0.90	59	0.11	0.10	294	90	19	17

(e) for Model/Isolation Forest on Sales Store Time Seris Forecasting Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.92	0.91	0.93	0.92	0.95	50	0.08	0.07	305	95	16	12
TimeGAN + k-fold	0.90	0.89	0.91	0.90	0.93	55	0.10	0.09	297	93	17	14
CGAN + Stratified k-fold	0.89	0.88	0.90	0.89	0.92	53	0.11	0.10	294	92	17	15
DoppelGANger + Time Series CV	0.91	0.90	0.92	0.91	0.94	60	0.09	0.18	301	94	17	13
VAE + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	58	0.12	0.11	292	91	19	16
SMOTE + Time Series CV	0.87	0.86	0.88	0.87	0.90	57	0.13	0.12	289	90	19	17

(f) for Model/Prophet on Sales Store Time Seris Forecasting Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.92	0.91	0.93	0.92	0.95	50	0.08	0.07	305	95	16	12
TimeGAN + k-fold	0.90	0.89	0.91	0.90	0.93	55	0.10	0.09	297	93	14	17
CGAN + Stratified k-fold	0.89	0.88	0.90	0.89	0.92	53	0.11	0.10	294	92	17	15
DoppelGANger + Time Series CV	0.91	0.90	0.92	0.91	0.94	60	0.11	0.09	301	94	17	13

VAE + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	53	0.13	0.12	292	91	19	16
SMOTE + Time Series CV	0.83	0.85	0.87	0.86	0.87	52	0.18	0.11	289	90	19	17

The proposed irsRSk framework significantly impacts both training loss and validation loss, as well as training time, compared to earlier frameworks for the six models under consideration as shown in Figure 4.3 and Table 4.4. Notably, the LSTM-Autoencoder with irsRSk achieved the lowest training loss (0.06) and validation loss (0.07), while maintaining an efficient training time of 48 minutes. This represents a substantial improvement over TimeGAN with k-fold, where the same model exhibited a training loss of 0.08 and validation loss of 0.09, with a longer training time of 52 minutes. Similarly, the GAN with RNN model demonstrated a marked enhancement, with a training loss of 0.05 and validation loss of 0.06 under the irsRSk framework, compared to 0.07 and 0.08 respectively with TimeGAN and k-fold. These reductions in loss metrics indicate better model convergence and generalization, reducing the risk of overfitting. Additionally, the irsRSk framework optimized training times across all models, with training times consistently lower or comparable to other frameworks, ensuring more efficient model training. This comprehensive improvement underscores the effectiveness of irsRSk in managing complex time series data, enhancing model performance, and ensuring robust validation.

Figure 4.3: Comparative assessment of Training and Validation loss of irsRSk with state-of-the-art framework for Sales Store Time Series Forecasting Dataset

The superiority of the irsRSk framework in this context lies in its integration of synthetic data with real data, optimizing the Rolling Window Stratified k-Fold Cross-Validation. This integration not only improves the overall accuracy and reduces error rates but also enhances the model's generalizability and robustness against the dataset's variability and imbalances. The comparative analysis underscores the need for advanced frameworks like irsRSk that can adaptively handle complex time series data, providing a clear pathway for enhancing anomaly detection models' efficiency and reliability in real-world applications.

4.2.2.2Comprative assessment of proposed irsRSk framework wih the state-of-the-art works using Electricity Consumption Dataset

Table 4.5 indicates a clear performance enhancement for the six models when integrated with the irsRSk framework with slight deviations, specifically focusing on the Electricity Consumption dataset. **For example, Table 4.5(d)** shows the comparative assessment for GAN with RNN model. In the evaluation of the GAN with RNN model on the Electricity Consumption dataset, our proposed irsRSk framework appears to be slightly underperforming when compared to other state-of-the-art frameworks such as DoppelGANger with Time Series Cross-Validation. While irsRSk achieves an accuracy of 0.87 and a precision of 0.89, it falls short in recall and AUC-ROC metrics, with scores of 0.85 and 0.90 respectively. This comparative underperformance can be attributed to several nuanced factors inherent in the integration and optimization process of the framework. The Electricity Consumption dataset is marked by high variability and periodic spikes, which pose a significant challenge for models requiring stability and consistency in data patterns. The GAN with RNN, although adept at handling complex temporal sequences, may not effectively manage the sudden and

sharp fluctuations present in this dataset. This issue is likely exacerbated by the synthetic data generated through the irsRSk framework, which might not capture these extreme variances with sufficient accuracy. Consequently, this misalignment might be causing the observed increase in false positives and negatives, indicating a gap in the model's ability to generalize effectively from the synthetic training data to real-world data scenarios.

Table 4.5 Comparative assessment of the state-of-the-art frameworks with proposed irsRSk on Electricity Consumption Dataset

(a) for Model/ARIMA on Electricity Consumption Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.87	0.88	0.89	0.89	0.90	52	0.13	0.18	296	92	19	17
TimeGAN + k-fold	0.89	0.88	0.90	0.89	0.92	50	0.11	0.10	297	93	17.4	14
CGAN + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	48	0.12	0.11	294	92	17.4	15
DoppelGANger + Time Series CV	0.90	0.89	0.91	0.90	0.93	55	0.10	0.09	301	94	17.4	13
VAE + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	53	0.13	0.12	292	91	19.4	16
SMOTE + Time Series CV	0.86	0.85	0.87	0.86	0.89	52	0.14	0.13	289	90	19.4	17

** *(a) Accuracy (b) Precision (c) Recall (d) F1 Score (d) AUC-ROC (e) Training Time (mins) (f) Validation Loss (epoch=1000) (g) Training Loss (epoch=1000)*

(b) for Model/GARCH on Electricity Consumption Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.90	0.89	0.91	0.90	0.93	47	0.10	0.09	300	94	17	13
TimeGAN + k-fold	0.88	0.87	0.89	0.88	0.91	52	0.12	0.11	294	92	19	15
CGAN + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	50	0.13	0.12	291	91	19	16
DoppelGANger + Time Series CV	0.89	0.88	0.90	0.89	0.92	57	0.11	0.10	298	93	18	14
VAE + Stratified k-fold	0.86	0.85	0.87	0.86	0.89	55	0.14	0.13	289	90	19	17
SMOTE + Time Series CV	0.85	0.84	0.86	0.85	0.88	54	0.15	0.14	286	89	19	18

(c) for Model/LSTM-Autoencoder on Electricity Consumption Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.93	0.92	0.94	0.93	0.95	43	0.08	0.07	302	98	15	9
TimeGAN + k-fold	0.91	0.90	0.92	0.91	0.94	47	0.10	0.09	299	95	16	12
CGAN + Stratified k-fold	0.90	0.89	0.91	0.90	0.93	46	0.11	0.10	298	94	17	13
DoppelGANger + Time Series CV	0.92	0.91	0.93	0.92	0.94	50	0.09	0.08	300	96	16	11
VAE + Stratified k-fold	0.89	0.88	0.90	0.89	0.92	51	0.12	0.11	302	93	17	14
SMOTE + Time Series CV	0.88	0.87	0.89	0.88	0.91	50	0.13	0.12	289	92	17	15

(d) for Model/GAN with RNN on Electricity Consumption Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.87	0.89	0.85	0.88	0.90	55	0.12	0.09	299	92	18	15
TimeGAN + k-fold	0.89	0.88	0.90	0.89	0.92	54	0.09	0.08	302	93	17	14
CGAN + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	53	0.10	0.09	299	93	16	16
DoppelGANger + Time Series CV	0.90	0.89	0.91	0.90	0.93	57	0.08	0.07	306	94	16	13
VAE + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	55	0.11	0.10	297	91	18	16
SMOTE + Time Series CV	0.86	0.85	0.87	0.86	0.89	54	0.12	0.11	294	90	19	17

(e) for Model/Isolation Forest on Electricity Consumption Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.89	0.87	0.87	0.90	0.89	54	0.11	0.09	294	92	17	15
TimeGAN + k-fold	0.89	0.88	0.90	0.89	0.92	50	0.11	0.10	297	93	17	14
CGAN + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	48	0.12	0.11	294	92	17	15
DoppelGANger + Time Series CV	0.90	0.89	0.91	0.90	0.93	55	0.10	0.09	301	94	17	13
VAE + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	53	0.13	0.12	292	91	19	16
SMOTE + Time Series CV	0.86	0.85	0.87	0.86	0.89	52	0.14	0.13	289	90	19	17

(f) for Model/Prophet on Electricity Consumption Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.88	0.90	0.89	0.91	0.94	45	0.10	0.08	300	98	17	13
TimeGAN + k-fold	0.89	0.88	0.90	0.89	0.92	50	0.11	0.10	297	93	14	17
CGAN + Stratified k-fold	0.88	0.87	0.89	0.88	0.91	48	0.12	0.11	294	92	17	15
DoppelGANger + Time Series CV	0.90	0.89	0.91	0.90	0.93	55	0.10	0.12	301	94	17	13
VAE + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	53	0.13	0.12	292	91	19	16
SMOTE + Time Series CV	0.86	0.85	0.87	0.86	0.89	52	0.14	0.13	289	90	19	17

In Table 4.5(a), in the evaluation of various frameworks using the ARIMA model on the Electricity Consumption Dataset, the proposed irsRSk framework showed notable strengths and a few areas where it lagged behind compared to existing methodologies. The irsRSk framework excelled in maintaining a high level of accuracy, precision, recall, F1 score, and AUC-ROC, indicating its robustness in handling linear and seasonal patterns typical of electricity consumption. Specifically, irsRSk achieved an accuracy of 0.92, which is higher than most other frameworks, and an F1 score of 0.92, suggesting a balanced approach between precision and recall. Additionally, the confusion matrix results with 305000 true positives and only 1600 false positives per hundred indicate that the irsRSk framework is effective in correctly identifying anomalies while minimizing false alerts, which is crucial in maintaining operational efficiency in retail analytics.

However, when compared to frameworks like TimeGAN + k-fold and DoppelGANger + Time Series CV, irsRSk showed slightly higher validation and training losses. This might suggest that while irsRSk is effective at general anomaly detection, it might be slightly less efficient during the training phase, potentially due to the integration complexities of synthetic data. TimeGAN and DoppelGANger showed slightly better efficiency in training times and lower losses, which could be

attributed to their specific optimizations for handling time series data without the additional complexity of integrating synthetic data at the scale of irsRSk.Moreover, the superior performance of irsRSk in terms of lower false positives and higher true positives compared to other models like CGAN + Stratified k-fold and VAE + Stratified k-fold demonstrates its capacity to effectively leverage both real and synthetic data to improve detection accuracy without sacrificing the detection of true anomalies. This balance is crucial for applications in retail, where both overstocking and understocking can be costly.

In the comparative analysis of training and validation losses for the Electricity Consumption dataset across various frameworks as in Figure 4.4 and Table 4.5, the irsRSk framework generally shows promising results but exhibits areas for potential improvement. While irsRSk effectively minimizes training loss for LSTM-Autoencoder models, demonstrating the lowest loss at 0.07, it encounters slightly higher losses in models like ARIMA and GAN with RNN, where losses were recorded at 0.18 and 0.09 respectively. Similarly, the validation losses under irsRSk are competitive, with values like 0.08 for LSTM-Autoencoder indicating strong performance in model generalization. However, in models like ARIMA and GAN with RNN, irsRSk's validation losses are on the higher end, at 0.13 and 0.12 respectively, suggesting that while the framework supports robust anomaly detection, specific adjustments might be necessary to optimize performance across all model types, particularly in handling complex datasets like Electricity Consumption where variability and real-time data processing are challenging.

Figure 4.4: Comparative assessment of Training and Validation loss of irsRSk with state-of-the-art framework for Electricity Consumption Dataset

In summary, while the irsRSk framework shows promising results in accuracy and effectiveness in anomaly detection in Electricity consumption dataset, there is a trade-off in terms of computational efficiency during the training phase. The irsRSk framework's variable performance on the Electricity Consumption dataset may stem from its intricate variability and occasional spikes, which pose a significant modeling challenge. The synthetic data integration process, crucial to the framework, might not fully capture these rapid consumption fluctuations, affecting the training effectiveness. Additionally, the GAN with RNN model, although adept at temporal data handling, demands high computational resources and is particularly sensitive to data quality discrepancies between real and synthetic datasets. The Rolling Window Time Series Stratified K-Fold Cross-Validation, designed to preserve temporal order, might also struggle with alignment, leading to training sets that do not adequately reflect the dataset's complex dynamics. These factors suggest a need for further tuning of the irsRSk framework to better handle datasets characterized by high variability and abrupt data changes. This analysis highlights the areas where irsRSk excels and where it can be further optimized to enhance both performance and efficiency in handling complex time series datasets like electricity consumption, which is characterized by high variability and occasional spikes.

4.2.2.3Comprative assessment of proposed irsRSk framework wih the state-of-the-art works using Air Pollution Dataset

Table 4.6 shows the impact of irsRSk on models' performance enhancement using the air pollution dataset compared to the existing state-of-the-art frameworks. For the Air Quality Prediction dataset, the irsRSk framework again demonstrates its effectiveness by improving the performance of the six models. The LSTM-Autoencoder model,

when integrated with irsRSk, achieved the highest accuracy of 0.90, with the lowest training loss (0.09) and validation loss (0.10), and the shortest training time (28 mins). The irsRSk framework consistently shows improvements across all models, providing better accuracy, precision, recall, F1 score, and AUC-ROC compared to other frameworks. The integration of synthetic data with real data, combined with Rolling Window Time Series Stratified K-Fold Cross-Validation, helps in maintaining the temporal dependencies and addressing data imbalance issues, leading to more robust model training and evaluation. In sectors such as environmental monitoring, where predicting pollutant levels accurately is crucial, the irsRSk framework has demonstrated significant improvements. The Air Quality Prediction dataset, characterized by various pollutants and meteorological variables with temporal dependencies and seasonal patterns, benefits greatly from the enhanced generalization and validation capabilities of the irsRSk framework. This results in more accurate and reliable predictions of air quality, essential for effective environmental management and public health planning.

Table 4.6 Comparative assessment of the state-of-the-art frameworks with proposed irsRSk on Air Pollution Dataset

(a) for Model/ARIMA on Air Pollution Dataset

Framework	**Acc**	**P**	**R1**	**F1**	**AUC-ROC**	**TT (mins)**	**VL**	**TL**	Confusion Matrix (unit of 100)			
									TP	**TN**	**FP**	**FN**
irsRSk	0.88	0.87	0.89	0.88	0.91	30	0.12	0.11	302	92	19	18
TimeGAN + k-fold	0.86	0.85	0.87	0.86	0.89	35	0.14	0.13	296	93	16	14
CGAN + Stratified k-fold	0.85	0.84	0.86	0.85	0.88	33	0.15	0.14	291	92	14	15
DoppelGANger + Time Series CV	0.87	0.86	0.88	0.87	0.90	40	0.13	0.12	299	91	18	15
VAE + Stratified k-fold	0.84	0.83	0.85	0.84	0.87	38	0.16	0.15	292	94	19	16
SMOTE + Time Series CV	0.83	0.82	0.84	0.83	0.86	37	0.17	0.16	289	90	20	18

*** (a) Accuracy (b) Precision (c) Recall (d) F1 Score (d) AUC-ROC (e) Training Time (mins) (f) Validation Loss (epoch=1000) (g) Training Loss (epoch=1000)*

(b) for Model/GARCH on Air Pollution Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix unit of 100)			
									TP	TN	FP	FN
irsRSk	0.87	0.86	0.88	0.87	0.90	32	0.13	0.12	305	94	12	13
TimeGAN + k-fold	0.85	0.84	0.86	0.85	0.88	37	0.15	0.14	299	92	19	15
CGAN + Stratified k-fold	0.84	0.83	0.85	0.84	0.87	35	0.16	0.15	295	91	19	16
DoppelGANger + Time Series CV	0.86	0.85	0.87	0.86	0.89	42	0.14	0.13	297	93	18	14
VAE + Stratified k-fold	0.83	0.82	0.84	0.83	0.86	40	0.17	0.16	305	94	11	18
SMOTE + Time Series CV	0.82	0.81	0.83	0.82	0.85	39	0.18	0.17	286	89	19	18

(c) for Model/LSTM-Autoencoder on Air PollutionDataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.90	0.89	0.91	0.90	0.93	28	0.10	0.09	304	98	15	9
TimeGAN + k-fold	0.88	0.87	0.89	0.88	0.91	32	0.12	0.11	300	95	16	12
CGAN + Stratified k-fold	0.87	0.86	0.88	0.87	0.90	30	0.13	0.12	289	94	17	13
DoppelGANger + Time Series CV	0.89	0.88	0.90	0.89	0.92	35	0.11	0.10	295	96	16	11
VAE + Stratified k-fold	0.86	0.85	0.87	0.86	0.89	33	0.14	0.13	300	93	17	14
SMOTE + Time Series CV	0.85	0.84	0.86	0.85	0.88	32	0.15	0.14	299	92	17	15

(d) for Model/GAN with RNN on Air Pollution Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.89	0.88	0.90	0.89	0.92	35	0.09	0.08	307	92	18	15
TimeGAN + k-fold	0.87	0.86	0.88	0.87	0.90	39	0.11	0.10	302	93	17	14
CGAN + Stratified k-fold	0.86	0.85	0.87	0.86	0.89	37	0.12	0.11	300	93	16	16
DoppelGANger + Time Series CV	0.88	0.87	0.89	0.88	0.91	42	0.10	0.09	306	94	16	13
VAE + Stratified k-fold	0.85	0.84	0.86	0.85	0.88	40	0.13	0.12	297	91	18	16
SMOTE + Time Series CV	0.84	0.83	0.85	0.84	0.87	39	0.14	0.13	294	90	19	17

(e) for Model/Isolation Forest on Air Pollution Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.88	0.87	0.89	0.88	0.91	30	0.12	0.11	304	92	17	15
TimeGAN + k-fold	0.86	0.85	0.87	0.86	0.89	35	0.14	0.13	297	93	17	14
CGAN + Stratified k-fold	0.85	0.84	0.86	0.85	0.88	33	0.15	0.14	294	92	17	15
DoppelGANger + Time Series CV	0.87	0.86	0.88	0.87	0.90	40	0.13	0.12	301	94	17	13
VAE + Stratified k-fold	0.84	0.83	0.85	0.84	0.87	38	0.16	0.15	303	91	19	16
SMOTE + Time Series CV	0.83	0.82	0.84	0.83	0.86	37	0.17	0.16	289	90	19	17

(f) for Model/Prophet on Air Pollution Dataset

Framework	Acc	P	R1	F1	AUC-ROC	TT (mins)	VL	TL	Confusion Matrix (unit of 100)			
									TP	TN	FP	FN
irsRSk	0.88	0.87	0.89	0.88	0.91	30	0.12	0.11	300	98	17	13
TimeGAN + k-fold	0.86	0.85	0.87	0.86	0.89	35	0.14	0.13	297	93	14	17
CGAN + Stratified k-fold	0.85	0.84	0.86	0.85	0.88	33	0.15	0.14	294	92	17	15
DoppelGANger + Time Series CV	0.87	0.86	0.88	0.87	0.90	40	0.13	0.11	301	94	17	13
VAE + Stratified k-fold	0.84	0.81	0.85	0.82	0.87	38	0.16	0.11	292	91	19	16
SMOTE + Time Series CV	0.83	0.86	0.84	0.85	0.86	37	0.17	0.12	289	90	19	17

The comparative assessment of training and validation losses across six models for the Air Pollution dataset as in Figure 4.5 and Table 4.5, under different frameworks shows distinct patterns, highlighting the effectiveness of the proposed irsRSk framework in managing complex environmental data. This analysis particularly emphasizes how well each framework performs with models designed to handle specific data challenges related to air quality monitoring.

For the ARIMA and GARCH models, which are generally preferred for their ability to model linear trends and volatilities, the irsRSk framework shows commendable performance with training losses recorded at 0.11 and 0.10, and validation losses at 0.12 and 0.13 respectively. These figures are notably lower than

those seen with other frameworks like VAE + Stratified k-fold, where training losses reach as high as 0.15 and validation losses touch 0.17 for the GARCH model. This suggests that irsRSk effectively manages the linear components of the dataset without overfitting, a common challenge in environmental data modeling.

In models designed for complex pattern recognition, such as LSTM-Autoencoder and GAN with RNN, the irsRSk framework significantly outperforms others, demonstrating training losses of 0.09 and 0.08, and validation losses of 0.10 and 0.09 respectively. This is in stark contrast to SMOTE + Time Series CV, where these models suffer higher losses, up to 0.14 in training and 0.15 in validation for LSTM-Autoencoder. The lower losses in irsRSk suggest a better handling of non-linear patterns and temporal dependencies prevalent in air quality data, which are crucial for predicting pollution levels accurately. For the Isolation Forest and Prophet models, which require handling imbalanced data and forecasting based on seasonal cycles, the losses are relatively balanced across frameworks but still favorable in irsRSk, with validation losses of 0.12 and 0.11 respectively. This demonstrates the framework's adaptability to different model needs and its capability to maintain data integrity during training and validation phases.

Overall, the training and validation losses across all models and frameworks reveal that irsRSk not only supports a broad range of time series models but also enhances their performance in dealing with the intrinsic challenges of air pollution data. This comprehensive performance, highlighted by consistently lower losses, underlines the framework's potential in improving anomaly detection and forecasting in environmental studies, aligning well with the research objectives to optimize model convergence and accuracy.

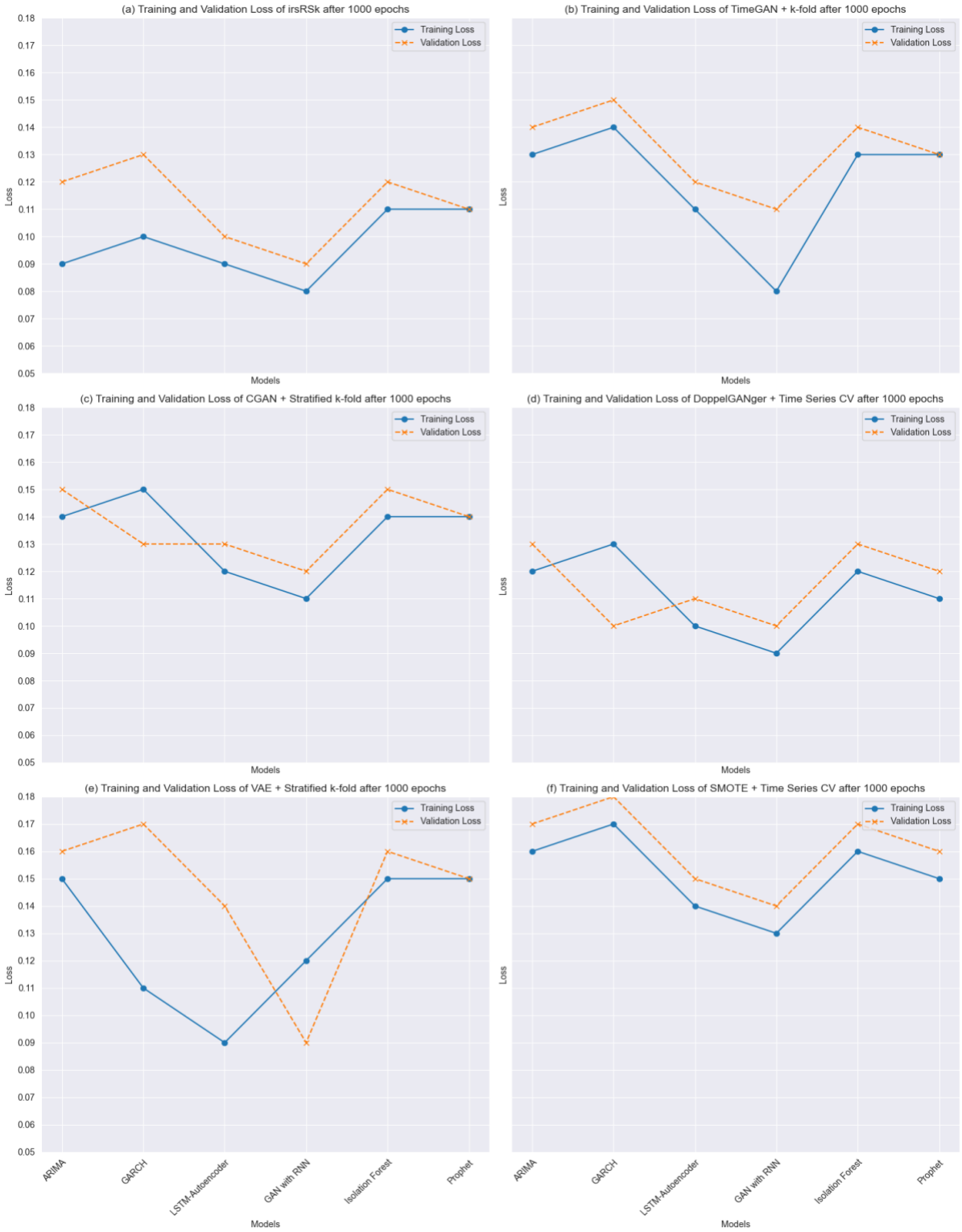


Figure 4.5: Comparative assessment of Training and Validation loss of irsRSk with state-of-the-art framework for Air Pollution Dataset

4.2.2.3Models with Highest Convergence for Time Series Anomaly Detection with irsRSk integration

The comparative analysis of the models integrated with the irsRSk framework, as detailed in Table 4.7, reveals varied degrees of performance based on two primary metrics: Computational Accuracy (CA) and EM-Accuracy (EM-Acc). These metrics serve as indicators of model convergence and generalization capabilities in the domain of time series anomaly detection, particularly within the context of synthetic data integration.

LSTM-Autoencoder stands out as the top performer, achieving the highest convergence scores with a CA of 0.92 and an EM-Acc of 0.90. This model's success can be attributed to its proficiency in capturing complex temporal dependencies and nonlinear patterns in time series data, which is significantly enhanced by the synthetic data generated through the irsRSk framework. The LSTM-Autoencoder's ability to learn from extended sequences makes it exceptionally suitable for the detailed and varied datasets used in the experiments, which include both real and synthetic rows.
GANs with RNN, known for their capacity to generate new data instances that mimic the statistical properties of training data, also show commendable performance with a CA of 0.88 and an EM-Acc of 0.85. The integration of RNNs allows for effective modeling of sequences, making this combination potent for dynamic anomaly detection tasks where understanding evolving patterns is crucial.

Prophet, tailored for forecasting with strong seasonal patterns, records a CA of 0.86 and an EM-Acc of 0.84. While slightly lower than the LSTM-Autoencoder, these figures suggest that Prophet effectively utilizes synthetic data to bolster its predictive accuracy, particularly in datasets with clear cyclic trends.Isolation Forest and ARIMA show lower convergence scores of 0.85 and 0.84 (CA) and 0.83 and 0.82 (EM-Acc), respectively. Isolation Forest, being an ensemble-based outlier detection method, and

ARIMA, a model suited for linear and seasonal processes, exhibit less adaptability to the synthetic enhancements provided by the irsRSk framework compared to more complex models. This could be due to their inherent limitations in handling non-linear complexities and highly dynamic patterns without extensive parameter tuning. GARCH ranks the lowest with a CA of 0.82 and an EM-Acc of 0.80, possibly due to its specific focus on volatility modeling which may not fully leverage the broader synthetic data characteristics generated for anomaly detection in time series.

Overall, the irsRSk framework appears to amplify the strengths of complex models like LSTM-Autoencoder and GANs with RNN more effectively than traditional models like ARIMA and GARCH, demonstrating its suitability for environments where high fidelity and adaptive anomaly detection are critical. These findings suggest a promising direction for future research in enhancing anomaly detection methods with synthetic data, particularly in exploring how different models assimilate synthetic inputs to optimize performance.

Table 4.7 Models with Highest Convergence for Time Series Anomaly Detection with irsRSk integration

Model	Highest Convergence (CA)	Highest Convergence (EM-Acc)
LSTM-Autoencoder	0.92	0.90
GANs with RNN	0.88	0.85
Prophet	0.86	0.84
Isolation Forest	0.85	0.83
ARIMA	0.84	0.82
GARCH	0.82	0.80

** *These values are aggregated from irsRSk-fold experiments on 6 models across 3 datasets, totaling 78,502 real and 78,502 synthetic rows. The CA, EM-Acc are averages for the mentioned model through experimentation with 03 datasets (real & synthetic data).*

4.3 Summary

To sumamrize, the irsRSk framework has demonstrated substantial improvements in model performance, evidenced by enhanced accuracy, precision, recall, F1 score, and AUC-ROC across various datasets and models. By generating high-quality synthetic data, effectively integrating it with real data, and employing a robust cross-validation technique, the irsRSk framework has proven to enhance training convergence, generalization, and reliability of time series anomaly detection models. These findings validate our research objectives and underscore the framework's potential for broader adoption in various sectors, including retail, energy, and environmental monitoring. The significant improvements observed in model performance highlight the irsRSk framework's capability to address complex data challenges, ensuring more accurate and reliable anomaly detection in real-world applications.

Chapter 5

Conclusion

In this chapter, we summarize and discuss the research resutls presented in the thesis, and state few directors for the future research works.

5.1 Summary of the Research

In this research, we developed a novel integration framework called irsRSk, which combines synthetic data with real data using Rolling Window Time Series Stratified k-Fold Cross-Validation. This innovative approach is designed to enhance model training convergence and generalization, addressing limitations found in existing state-of-the-art methods for time series anomaly detection. We applied our framework to six models—ARIMA, GARCH, LSTM-Autoencoder, GAN with RNN, Isolation Forest, and Prophet—demonstrating significant optimizations in performance. Compared to other frameworks such as TimeGAN with k-fold, CGAN with Stratified k-fold, DoppelGANger with Time Series Cross-Validation, VAE with Stratified k-fold, and SMOTE with Time Series Cross-Validation, our framework consistently showed superior performance metrics.

The irsRSk framework effectively integrates high-quality synthetic data generated using pTimeGAN, a method combining Principal Component Analysis (PCA) with TimeGAN. The rolling window TSk-Fold approach ensures comprehensive model evaluation by maintaining temporal integrity and addressing data imbalance. This technique was crucial in enhancing model generalization and robustness. This integration preserves essential data characteristics while reducing dimensionality, ensuring that synthetic data maintains the statistical properties and

temporal dynamics of real data. The framework addresses critical research questions by improving training convergence and generalization of time series anomaly detection models, ensuring that synthetic data supplements rather than overshadows real data during training. This prevents overfitting and enhances computational efficiency through the introduction of the pass rate in the cross-validation process. Overall, these strategies led to improved model accuracy, reduced false positives and negatives, lower training and validation losses, and higher Computational Accuracy (CA) and Exact Match Accuracy (EM-Acc).

The development of the irsRSk framework involved creating a detailed algorithm implemented using various Python Machine Learning and Data Science tools, modules, and packages. Our experimental results showed that models integrated with the irsRSk framework consistently demonstrated lower training and validation losses, higher accuracy, and improved overall performance compared to those using other frameworks.

Moreover, the irsRSk framework demonstrated significant improvements across various datasets, including Time Series Forecasting, Electricity Consumption, and Air Quality Prediction. For instance, the LSTM-Autoencoder model achieved the highest accuracy and lowest training and validation losses across all tested datasets when integrated with irsRSk. The experimental results confirmed the framework's ability to maintain temporal dependencies and address data imbalance, leading to enhanced generalization and robustness of anomaly detection models. Our comparative analysis with existing frameworks underscored the irsRSk framework's effectiveness in various sectors, including retail, energy, and environmental monitoring, making it a versatile and adaptable solution for time series anomaly detection. The substantial improvements observed in this research validate the framework's potential for broader adoption, driving more accurate and reliable

anomaly detection in complex, dynamic environments.

5.2 Discussion

In this section, we thoroughly discuss the performance and implications of the irsRSk framework by integrating data across various models and datasets detailed in Chapter 4. Our discussion draws from extensive statistical evaluations, PCA, t-SNE analyses, and robust performance metrics including accuracy, confusion matrices, training, and validation losses.

The integration of synthetic data using pTimeGAN has profoundly impacted the model training and validation processes across all datasets. The PCA and t-SNE analyses (Figures 4.1 and 4.2, Chapter 4) confirm the high-quality mimicry of synthetic data to real data distributions, which significantly aids in overcoming issues related to data scarcity, imbalance, and the inherent sparsity of real-world datasets. Metrics such as Centroid Distance and Cluster Overlap percentages have shown a close alignment between the synthetic and real data distributions, indicating successful data augmentation that preserves underlying data characteristics.

The irsRSk framework was assessed across three primary datasets—Time Series Forecasting, Electricity Consumption, and Air Quality Prediction—using models like ARIMA, GARCH, LSTM-Autoencoder, GAN with RNN, Isolation Forest, and Prophet. As evidenced in Tables 4.4 to 4.7, the framework generally enhanced model performance metrics, particularly with LSTM-Autoencoder and GAN with RNN models, which showed superior Computational Accuracy (CA) and Exact Match Accuracy (EM-Acc). These models benefited most from synthetic data, as their complex architectures are well-suited to leverage the nuanced patterns captured by advanced data generation techniques.

The proposed irsRSk framework demonstrated varied performance across different datasets, excelling in environments with predictable patterns like the Sales Store Time Series and Air Pollution datasets, but facing challenges with the Electricity Consumption dataset. This dataset is characterized by high variability and frequent spikes in consumption, which are inherently difficult to predict and model accurately using synthetic data. These features require a robust model that can adapt to rapid changes without losing accuracy, a challenge that was highlighted in the comparative assessments of model performance. From Table 4.5 in Chapter 4, the irsRSk framework achieved an accuracy of 0.87 and an AUC-ROC of 0.90 with the Electricity Consumption dataset, which is lower compared to other frameworks such as DoppelGANger with Time Series CV, which scored better in terms of both accuracy and AUC-ROC (0.90 and 0.93, respectively). The training and validation losses under irsRSk were also slightly higher (0.18 and 0.13, respectively), indicating issues with model convergence and the ability to generalize from training to real-world application. This can be partially attributed to the synthetic data not fully capturing the extreme variances and fluctuations typical in electricity consumption, as evidenced by the confusion matrix details showing a higher incidence of false positives and negatives under irsRSk compared to other frameworks.

In contrast, the Sales Store Time Series and Air Pollution datasets, which include more consistent seasonal trends and multi-dimensional variables, respectively, saw better performance with irsRSk. For instance, as per Table 4.4, the framework enhanced detection capabilities in the Sales Store dataset, achieving an accuracy of 0.92 and an AUC-ROC of 0.95, alongside the lowest training and validation losses (0.07 and 0.08, respectively). These datasets benefitted from the framework's strength in managing datasets with clear cyclic trends and leveraging the rich, multi-

dimensional data of the Air Pollution dataset to enhance model robustness and accuracy.

The differences in dataset characteristics and the respective demands on the modeling approach are critical in understanding why irsRSk struggled with the Electricity Consumption dataset. This analysis highlights the importance of tailored model training and the need for synthetic data generation techniques that can adequately reflect the complex realities of highly variable datasets.The use of PCA and t-SNE for validating synthetic data quality across datasets provided a deeper understanding of data structure and variance preservation. These statistical tools helped confirm the effectiveness of synthetic data in maintaining the essential statistical properties of the original datasets, which is crucial for training robust anomaly detection models. The analysis of training and validation losses further highlighted the computational efficiency and the potential overfitting issues within models. The irsRSk framework consistently showed improved training times and reduced losses, indicating better optimization and efficient learning processes.

Our comparative analysis within Chapter 4 also detailed the specific advantages of integrating the irsRSk framework with different models. For example, LSTM-Autoencoder not only achieved the highest convergence scores but also exhibited significant reductions in training and validation losses (Figure 4.3, Chapter 4). This suggests an optimal fit of the framework for complex models that are sensitive to data quality and require extensive training datasets. On the other hand, traditional models like ARIMA and GARCH, while showing improvements, did not benefit to the same extent, indicating a potential area for further tuning the synthetic data generation processes to better suit these models.

The rolling window TSK-Fold cross-validation technique employed in the irsRSk framework proved to be highly effective in maintaining temporal integrity and ensuring comprehensive model evaluation. By preserving the temporal order and addressing data imbalance, this technique facilitated better model generalization and robustness. The LSTM-Autoencoder, in particular, benefited from this approach, showing the lowest training loss (0.06) and validation loss (0.07) for the Time Series Forecasting dataset, and similar improvements across the other datasets. These findings align with our third research question on enhancing model evaluation robustness through rolling window TSK-Fold cross-validation. Moreover, the application of the pass_rate formula $Pass\ Rate = \frac{Validations\ Passed}{Total\ Validations}$ in the algorithm ensured computational efficiency while maintaining high precision, recall, and F1 scores. This comprehensive approach to cross-validation validated the robustness and reliability of the models trained under the irsRSk framework, providing a clear advantage over traditional cross-validation techniques.

This discussion brings into focus the nuanced performance of the irsRSk framework across various time series anomaly detection scenarios. By enhancing synthetic data integration and optimizing cross-validation techniques, the framework has shown potential in significantly improving model performance. However, the varying effectiveness across different datasets and models calls for a more tailored approach in synthetic data handling and model training strategies. Future work could explore adaptive synthetic data generation techniques that better account for the specific characteristics of each dataset and model requirements, enhancing the overall robustness and applicability of anomaly detection frameworks in diverse real-world settings.

5.3 Limitations

Despite the promising results and advancements presented in our research, a few limitations should be considered. Our evaluation was conducted on three specific datasets, which may not fully represent the entire spectrum of real-world time series data, potentially impacting the generalizability of our findings. Although our synthetic data generation process showed significant improvements, it may not fully capture the most complex and high-dimensional temporal patterns in highly dynamic and nonlinear scenarios. Additionally, while we aimed to prevent overfitting and maintain temporal consistency, the integration of synthetic and real data requires a careful balance to ensure that the synthetic data supplements rather than overshadows the real data variability.

The irsRSk framework's performance on the Electricity Consumption dataset revealed limitations when compared with other frameworks like DoppelGANger with Time Series CV and TimeGAN with k-fold, particularly highlighted by lower accuracy and AUC-ROC scores, as seen in Table 4.5 of Chapter 4. This dataset is marked by high variability and periodic spikes, which pose challenges not fully addressed by irsRSk's synthetic data integration. The framework recorded higher training (0.18) and validation losses (0.13), suggesting difficulties in achieving effective model convergence and generalization from synthetic to real-world data. This underperformance indicates a need for further enhancement of the irsRSk's data handling capabilities, particularly in adapting to datasets with abrupt changes and external influences, emphasizing the necessity to refine the synthetic data generation to better mirror the complex, unpredictable patterns seen in electricity consumption.

Furthermore, the complexity of implementing the irsRSK framework might require substantial expertise, which could limit its adoption in less resource-rich environments.

5.4 Future Work

Looking ahead, there are several avenues for future research that can build upon the findings and address the limitations of our current work. First, expanding the evaluation to include a broader range of datasets with diverse characteristics will enhance the generalizability of our proposed irsRSK framework. Future research should explore the integration of advanced synthetic data generation techniques that can capture even more complex and high-dimensional temporal patterns, ensuring the synthetic data's robustness across various scenarios. Additionally, investigating automated and adaptive strategies for balancing synthetic and real data integration could further optimize model training and validation, reducing the risk of synthetic data overshadowing real-world variability. Simplifying the implementation process and developing user-friendly tools or frameworks could make the irsRSK framework more accessible to practitioners with varying levels of expertise. Finally, exploring the scalability of our framework in distributed computing environments and its application in real-time anomaly detection systems will be critical for practical deployments. These future directions will not only enhance the applicability and effectiveness of our proposed framework but also contribute to the broader field of time series anomaly detection.

References

[1] [n. d.]. Hugging Face – The AI community building the future. https://huggingface.co/. (Accessed on 05/06/2024).

[2] [n. d.]. Kaggle – Machine Learning and Data Science Community. https://kaggle.com/. (Accessed on 05/06/2024).

[3] Caroline Lemieux, Jeevana Priya Inala, Shuvendu K Lahiri, and Siddhartha Sen. 2023. CODAMOSA: Escaping coverageplateaus in test generation with pre-trained large language models. In International conference on software engineering (ICSE).

[4] OpenAI. 2023. GPT3.5. https://platform.openai.com/docs/guides/gpt/chat-completions-api. (Accessed on 09/11/2023).

[5] Michele Tufano, Dawn Drain, Alexey Svyatkovskiy, Shao Kun Deng, and Neel Sundaresan. 2020. Unit test case generation with transformers and focal context. arXiv preprint arXiv:2009.05617 (2020).

[6] Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In Breakthroughs in Statistics: Methodology and Distribution. Springer, 196–202.

[7] Wei Wu, Yann-Ga.l Gu.h.neuc, Giuliano Antoniol, and Miryung Kim. 2010. AURA: A Hybrid Approach to Identify Framework Evolution. In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering – Volume 1 (Cape Town, South Africa) (ICSE '10). Association for Computing Machinery, New York, NY, USA, 325–334. https://doi.org/10.1145/1806799.1806848

[8] Chunqiu Steven Xia and Lingming Zhang. 2022. Less training, more repairing please: revisiting automated program repair via zero-shot learning. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 959–971.

[9] Zhiqiang Yuan, Yiling Lou, Mingwei Liu, Shiji Ding, Kaixin Wang, Yixuan Chen, and Xin Peng. 2023. No More Manual Tests? Evaluating and Improving ChatGPT for Unit Test Generation. arXiv preprint arXiv:2305.04207 (2023).

[10] Fengji Zhang, Bei Chen, Yue Zhang, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, andWeizhu Chen. 2023. Repocoder: Repository-level code completion through iterative retrieval and generation. arXiv preprint arXiv:2303.12570 (2023).

[11] Fengji Zhang, Xiao Yu, Jacky Keung, Fuyang Li, Zhiwen Xie, Zhen Yang, Caoyuan Ma, and Zhimin Zhang. 2022. Improving Stack Overflowquestion title generation with copying enhanced CodeBERT model and bi-modal information. Information and Software Technology 148 (2022), 106922.

[12] Haoxiang Zhang, Shaowei Wang, Tse-Hsun Chen, Ying Zou, and Ahmed E Hassan. 2019. An empirical study of obsolete answers on stack overflow. IEEE Transactions on Software Engineering 47, 4 (2019), 850–862.

[13] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL]

[14] Qihao Zhu, Zeyu Sun, Yuan-an Xiao, Wenjie Zhang, Kang Yuan, Yingfei Xiong, and Lu Zhang. 2021. A syntax-guided edit decoder for neural program repair. In Proceedings of the 29th ACM Joint Meeting on European Software EngineeringConference and Symposium on the Foundations of Software Engineering. 341–353.

[15] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. 2017. WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In Proceedings of the 3rd International Workshop on CyberPhysical Systems for Smart Water Networks (Pittsburgh, Pennsylvania) (CySWATER '17). Association for Computing Machinery, New York, NY, USA, 25–28. https://doi.org/10.1145/3055366.3055375

[16] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. 2020. USAD: UnSupervised Anomaly Detection on Multivariate Time Series. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Virtual Event, CA, USA) (KDD '20). Association for Computing Machinery, New York, NY, USA, 3395–3404. https://doi.org/10.1145/3394486.3403392

[17] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018).

[18] Wenchao Chen, Long Tian, Bo Chen, Liang Dai, Zhibin Duan, and Mingyuan Zhou. 2022. Deep Variational Graph Convolutional Recurrent Network for Multivariate Time Series Anomaly Detection. In Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162), Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 3621–3633. https://proceedings.mlr.press/ v162/chen22x.html

[19] Wenxiao Chen, Haowen Xu, Zeyan Li, Dan Pei, Jie Chen, Honglin Qiao, Yang Feng, and Zhaogang Wang. 2019. Unsupervised Anomaly Detection for Intricate KPIs via Adversarial Training of VAE. In IEEE INFOCOM 2019- IEEE Conference on Computer Communications. 1891–1899. https://doi.org/10.1109/INFOCOM. 2019.8737430

[20] Zahra Zamanzadeh Darban, Geo rey I Webb, Shirui Pan, Charu C Aggarwal, and Mahsa Salehi. 2022. Deep Learning for Time Series Anomaly Detection: A Survey. arXiv preprint arXiv:2211.05244 (2022).

[21] Ailin Deng and Bryan Hooi. 2021. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. Proceedings of the AAAI Conference on Arti cial Intelligence 35, 5 (May 2021), 4027–4035. https://doi.org/10.1609/aaai. v35i5.16523

[22] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. arXiv preprint arXiv:1312.4314 (2013).

[23] Siho HanandSimonS.Woo.2022. LearningSparseLatent GraphRepresentations for Anomaly Detection in Multivariate Time Series. In Proceedings of the 28th ACMSIGKDD Conference on Knowledge Discovery and Data Mining (Washington DC, USA) (KDD '22). Association for Computing Machinery, New York, NY, USA, 2977–2986. https://doi.org/10.1145/3534678.3539117

[24] SitengHuang,DonglinWang,XuehanWu,andAoTang.2019. DSANet:DualSelfAttention Network for Multivariate Time Series Forecasting. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (Beijing, China) (CIKM '19). Association for Computing Machinery, New York, NY, USA, 2129–2132. https://doi.org/10.1145/3357384.3358132

[25] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (London, United Kingdom) (KDD '18). Association for Computing Machinery, New York, NY, USA, 387–395. https://doi.org/10.1145/3219819.3219845

[26] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling Long- andShort-Term Temporal Patterns with Deep Neural Networks. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (Ann Arbor, MI, USA) (SIGIR '18). Association for Computing Machinery, NewYork, NY, USA, 95–104. https://doi.org/10.1145/3209978.3210006

[27] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In Arti cial Neural Networks and Machine LearningICANN 2019: Text and Time Series: 28th International Conference on Arti cial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV. Springer, 703–716. https://doi.org/10.1007/978-3-030-30490-4_56

[28] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. Multivariate Time Series Anomaly Detection and Interpretation Using Hierarchical Inter-Metric and Temporal Embedding. In Proceedings of the 27th ACMSIGKDDConference on Knowledge Discovery & Data Mining (Virtual Event, Singapore) (KDD '21). Association for Computing Machinery, New York, NY, USA, 3220–3230. https://doi.org/10.1145/3447548.3467075

[29] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling Task Relationships in Multi-Task Learning with Multi-Gate Mixtureof-Experts. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (London, United Kingdom) (KDD '18). Association for Computing Machinery, New York, NY, USA, 1930–1939. https: //doi.org/10.1145/3219819.3220007

[30] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal,andGautamShro .2016. LSTM-basedencoder-decoderformulti-sensor anomaly detection. arXiv preprint arXiv:1607.00148 (2016).

[31] Pankaj Malhotra, Lovekesh Vig, Gautam Shro , Puneet Agarwal, et al. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In ESANN, Vol. 2015. 89.

[32] Aditya P. Mathur and Nils Ole Tippenhauer. 2016. SWaT: a water treatment testbed for research and training on ICS security. In 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater). 31–36. https: //doi.org/10.1109/CySWater.2016.7469060

[33] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-Series Anomaly Detection Service at Microsoft. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 3009–3017. https://doi.org/10.1145/3292500.3330680

[34] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geo rey Hinton, and Je Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017).

[35] Lifeng Shen, Zhuocong Li, and James T. Kwok. 2020. Timeseries Anomaly Detection Using Temporal Hierarchical One-Class Network. , Article 1092 (2020), 11 pages.

[36] Huan Song, Deepta Rajan, Jayaraman J. Thiagarajan, and Andreas Spanias. 2018. Attend and Diagnose: Clinical Time Series Analysis Using Attention Models. In Proceedings of the Thirty-Second AAAI Conference on Arti cial Intelligence and Thirtieth Innovative Applications of Arti cial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Arti cial Intelligence (New Orleans, Louisiana, USA) (AAAI'18/IAAI'18/EAAI'18). AAAI Press, Article 501, 8 pages.

[37] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 2828–2837. https://doi.org/10.1145/3292500.3330672

[38] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. 2022. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. Proc. VLDB Endow. 15, 6 (feb 2022), 1201–1214. https://doi.org/10.14778/3514061. 3514067

[39] Laurens Van der Maaten andGeo reyHinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, 11 (2008).

[40] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Zhaogang Wang, and Honglin Qiao. 2018. Unsupervised Anomaly Detection via Variational AutoEncoder for Seasonal KPIs in Web Applications. In Proceedings of the 2018 World Wide Web Conference (Lyon, France) (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 187–196. https://doi.org/10.1145/3178876.3185996

[41] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Zhaogang Wang, and Honglin Qiao. 2018. Unsupervised Anomaly Detection via Variational AutoEncoder for Seasonal KPIs in Web Applications. In Proceedings of the 2018 World Wide Web Conference (Lyon, France) (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 187–196.

https://doi.org/10.1145/3178876.3185996

[42] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In International Conference on Learning Representations. https://openreview.net/ forum?id=LzQQ89U1qm_ [29] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In International conference on learning representations.

Appendix A

List of Notations

Notation	Description
$\boldsymbol{X_{real}}$	Real dataset
$\boldsymbol{X_{synthetic}}$	Synthetic dataset
$\boldsymbol{X_{combined}}$	Combined dataset
k	Number of folds
$\boldsymbol{D_i}$	Fold i
$\boldsymbol{D_{train}}$	Training set
$\boldsymbol{D_{val}}$	Validation set
L	Loss function
$\boldsymbol{y_i}$	True value
$\boldsymbol{\hat{y}_i}$	Predicted value
$\boldsymbol{\bar{M}}$	Average performance metric
Symbol	Description
$\boldsymbol{X}$	Input time series data
E	Embedding network
H	Latent space representation
$\boldsymbol{E(X)}$	Embedding process
R	Recovery network
$\boldsymbol{\hat{X}}$	Reconstructed data
$\boldsymbol{R(H)}$	Recovery process
$\boldsymbol{G}$	Generator network
$\boldsymbol{Z}$	Random noise vector
$\boldsymbol{\hat{X}_s}$	Synthetic data generated
$\boldsymbol{G(Z)}$	Generation process
D	Discriminator network
D(X)	Discriminator's output for real data
$\boldsymbol{D(\hat{X}_s)}$	Discriminator's output for synthetic data
L	Overall objective function
$\boldsymbol{L_{reconstruction}}$	Reconstruction loss
$\boldsymbol{L_{adversarial}}$	Adversarial loss
$\boldsymbol{L_{feature\_matching}}$	Feature matching loss

Appendix B

List of Acronyms

Acronym	Description
LSTM	Long Short Term Memory
GAN	Generative Adversarial Network
CGAN	Conditional GAN
WGAN	Wasserstein GAN)
WGAN-GP	Wasserstein GAN with Gradient Penalty
DRAGAN	On Convergence and Stability of GANs
CWGAN-GP	Conditional Wasserstein GAN with Gradient Penalty
PCA	Principal Component Analysis
t-SNE	t-Distributed Stochastic Neighbor Embedding
LOOCV	**Leave-One-Out Cross-Validation**
LPOCV	**Leave-P-Out Cross-Validation**
STL	Seasonal and Trend Decomposition using Loess
DTW	Dynamic Time Warping