UTILIZING MACHINE LEARNING TECHNIQUES FOR UNIVARIATE TIME SERIES ANOMALY DETECTION

2025
MASTER THESIS
COMPUTER ENGINEERING

Mohamad ALKHOJA

Assist.Prof.Dr. Kürşat Mustafa KARAOĞLAN

UTILIZING MACHINE LEARNING TECHNIQUES FOR UNIVARIATE TIME SERIES ANOMALY DETECTION

Mohamad ALKHOJA

Assist.Prof.Dr. Kürşat Mustafa KARAOĞLAN

T.C.

Karabuk University

Institute of Graduate Programs

Department of Computer Engineering

Prepared as

Master Thesis

KARABUK

April 2025

I certify that in my opinion the thesis submitted by Mohamad ALKHOJA titled "UTILIZING MACHINE LEARNING TECHNIQUES FOR UNIVARIATE TIME SERIES ANOMALY DETECTION" is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Assist.Prof.Dr. Kürşat Mustafa KARAOĞLAN .....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. April 15, 2025

Examining Committee Members (Institutions)	Signature
Chairman : Assist.Prof.Dr. Nehad T.A. RAMAHA (KBU)	.....
Member : Assist.Prof.Dr. Kürşat Mustafa KARAOĞLAN (KBU)	.....
Member : Assist.Prof.Dr. Alper Talha KARADENİZ (SAMÜ)	.....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Doç. Dr. Zeynep ÖZCAN .....
Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Mohamad ALKHOJA

ABSTRACT

M. Sc. Thesis

UTILIZING MACHINE LEARNING TECHNIQUES FOR UNIVARIATE TIME SERIES ANOMALY DETECTION

Mohamad ALKHOJA

Karabük University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Kürşat Mustafa KARAOĞLAN
April 2025, 56 pages

Time Series (TS) represents a chronological sequence of data recorded at regular intervals, and it is critical for behavioral analysis and system prediction. Anomalies in TS data are observations that deviate from standard behavioral patterns and are considered early indicators of potential system anomalies. These anomalies often signal critical events such as system failures, security breaches, or abnormal industrial, financial, and environmental behaviors. The increasing complexity of systems and data volume necessitates automated detection approaches. This study conducted a comparative analysis of four machine learning models: Regression, Categorical Boosting (CatBoost), Extreme Gradient Boosting (XGBoost), and Random Forest (RF) algorithms for anomaly detection in univariate TS data using the Python-based Darts Framework. The Darts Framework enables methodological standardization in TS analysis, allowing for reproducible and reliable results. The performance of these models was evaluated using accuracy, and the AUC was used to assess the models. In

experimental studies, metrics were conducted on four different univariate TS datasets for optimal anomaly detection. All models achieved a high accuracy rate above 90% across all datasets. Despite the change in the model's performance depending on the characteristics of the dataset, XGBoost and CatBoost showed superior performance, exceeding 96% in most datasets, followed by RF with similar results but less, varying between 92% and 97% in some datasets, and lastly the regression model. The findings of this study provide a quantitative framework for algorithm selection in real-time anomaly detection applications and present an effective approach capable of high-accuracy detection even in limited resource environments. This study contributes to model selection in real-time univariate TS anomaly detection. It offers reliability and the potential for adaptation to different application areas without the overhead of more complex architectures.

Key Words : Time Series, Univariate, Machine Learning, Anomaly Detection, XGBoost, CatBoost, Darts Framework.

Science Code : 92432

ÖZET

Yüksek Lisans Tezi

Tek Değişkenli Zaman Serisi Anomali Tespiti için Makine Öğrenimi Tekniklerinin Kullanımı

Mohamad ALKHOJA

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi. Kürşat Mustafa KARAOĞLAN
April 2025, 56 sayfa

Zaman serileri (TS), düzenli aralıklarla toplanan sıralı verilerdir ve sistemlerin davranışlarını anlamak için önemlidir. Zaman serilerindeki anomaliler, normal davranış kalıplarından sapan ve genellikle sistem sorunlarına işaret eden değerlerdir. Bu anomaliler sistem arızaları, güvenlik sorunları veya çeşitli alanlardaki beklenmeyen durumları gösterebilir. Sistemlerin karmaşıklığı ve veri miktarı arttıkça, anomalileri otomatik olarak tespit etmek zorunlu hale gelmiştir. Bu tez çalışmasında, tek değişkenli zaman serilerinde anomali tespiti için Python tabanlı Darts Framework kullanılarak Regresyon, CatBoost, XGBoost ve Random Forest (RF) olmak üzere dört farklı makine öğrenmesi modeli karşılaştırılmıştır. Darts Framework, zaman serisi analizinde metodolojik standardizasyon sağlayarak tekrarlanabilir ve güvenilir sonuçlar elde edilmesine olanak tanımıştır. Modellerin değerlendirilmesinde doğruluk (accuracy) ve AUC metrikleri kullanılmıştır. Dört farklı tek değişkenli zaman serisi veri kümesi üzerinde gerçekleştirilen deneysel çalışmalarda, tüm modeler tüm veri

kümelerinde yüksek doğruluk değerlerine %90'ın üzerinde yüksek doğruluk elde etti ulaşmıştır. Performansın veri setinin karakteristiğine bağlı olarak değişmesine rağmen, XGBoost ve CatBoost çoğu veri kümesinde %96'yı aşan üstün performans gösterdi, bunu benzer ancak biraz daha az sonuç veren RF modeli izledi, Bazı veri setlerinde %92 ile %97 arasında değişti ve son olarak regresyon modeli. Bu çalışmanın bulguları, gerçek zamanlı anomali tespiti uygulamalarında algoritma seçimi için nicel bir çerçeve sunmakta, sınırlı hesaplama kaynakları gerektiren ortamlarda dahi yüksek doğrulukta anomali tespiti sağlayan etkin bir yaklaşım ortaya koymaktadır. Bu çalışma, gerçek zamanlı tek değişkenli zaman serisi anomali tespiti uygulamalarında model seçimine katkı sağlamakta ve daha karmaşık mimarilere ihtiyaç duymadan farklı uygulama alanlarına uyarlanabilirlik açısından güvenilir bir alternatif sunmaktadır.

Anahtar Kelimeler : Zaman Serisi, Tek Değişkenli, Anomali Tespiti, Makine Öğrenimi, XGBoost, CatBoost, Darts Framework.

Bilim Kodu **:** 92432

ACKNOWLEDGEMENT

And say: My Lord! Increase me in knowledge [Ta Ha - 20:114]. First, Alhamdulillah for everything I have achieved. I'm glad to have completed my thesis, which is an important step at the start of my journey. Also, I would like to express my deepest appreciation and thanks to my supervisor, Assist. Prof. Dr. Kürşat Mustafa KARAOĞLAN, for his invaluable guidance and continuous support, which helped me to represent my work in the best way. Then, I would like to thank my beloved family for encouraging and supporting me with unconditional love and continued encouragement. Also, I appreciate my friends for their support during my challenging times.

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

SYMBOLS AND ABBREVITIONS INDEX

ABBREVITIONS

AI	: Artificial Intelligence
ARIMA	: Auto Regressive Integrated Moving Average
AUC	: Area Under the Curve
CatBoost	: Category Boosting
DL	: Deep Learning
GB	: Gradient Boosting
IoT	: Internet of Things
LSTM	: A Long Short-Term Memory
ML	: Machine Learning
NAB	: Numenta Anomaly Benchmark
NYC	: New York City
RF	: Random Forest
ROC	: Receiver Operating Characteristic Curve
STL	: Seasonal and Trend decomposition using Loess
TS	: Time Series
XGBoost	: Extreme Gradient Boosting

PART 1

INTRODUCTION

Time Series (TS) anomaly detection plays a crucial role in various industries [1]. This study uses Machine Learning (ML) methods to identify abnormal patterns in TS data and focuses on utilizing and evaluating the performance of ML models for univariate TS anomaly detection using the Darts framework, aiming to provide insights into their effectiveness and computational efficiency in real-world applications. In the first part, we start with general information about the research background, then we mention the problem statement and research objectives, and end with the scope of the study.

1.1. BACKGROUND

TS is any set of data that has been ordered in time and contains information recorded or gathered at fixed intervals [2]. TS data is a cornerstone of modern data analysis. The temporally structured nature of TS data allows research in terms of dynamic patterns that show developments such as trends, seasonality, and dependencies over time [3]. TS analysis is a major constituent of many applications, such as stock market prediction, energy demand forecast, climate modeling, and healthcare analytics [4]. Many of these applications require accurate forecasting together with anomaly detection to ensure that the system is optimized and operationally efficient. In particular, univariate TS, which involves observations of a single variable over time, is still used due to its simplicity and the wide availability of single-channel sensor or process data [5]. Univariate TS anomaly detection [6] focuses on analyzing individual variables to identify deviations from long-term patterns. In this approach, samples with significant differences from the overall distribution or the discriminator pattern are considered outliers.

In recent years, the rapid growth of data generation and its availability across various industries has led to the increased application of ML techniques, especially in anomaly detection [7]. Anomaly detection identifies data points or patterns that significantly differ from the anticipated behavior [8]. Often indicate something critically important, such as system failure, fraud, or other irregular activities [9]. The ability of anomaly detection lies in their early identification, which can prevent expensive failures, minimize the risks, and boost competent decision-making in real live applications [3], [10]. For instance, predictive maintenance in industrial systems employs anomaly detection to spot future downtime due to potential issues [11], [12]. The ability to identify and flag anomalous patterns in temporal data plays a critical role in fraudulent activities, identity or passport documents, insurance claims, and healthcare fraud [13], while in finance, it flags unusual market activities indicating fraud or instability.

However, recent developments in ML have transformed TS anomaly detection, enabling models to efficiently identify complex temporal patterns and nonlinear correlations within datasets [14]. While traditional statistical methods often face limitations in addressing these complexities, ML-based approaches leverage adaptive algorithms and sophisticated feature extraction techniques to effectively handle the intricate nuances of TS data across diverse domains [15]. Among these advanced methods, Ensemble methods such as Random Forest (RF) and Gradient Boosting (GB) algorithms like Categorical Boosting (CatBoost) and Extreme Gradient Boosting (XGBoost) have been widely applied to take advantage of their high resistance to noise and their flexibility. These models are designed to apply complex feature selection and weighting mechanisms, which enable the identification of anomalies, even in noisy and incomplete datasets [16]. Furthermore, it has been demonstrated that this kind of framework, which combines ML algorithms with some anomaly detection techniques, enhances performance by integrating domain-specific knowledge and improving TS modeling [17]. These models function particularly well when applied to real-world datasets, such as the Numenta Anomaly Benchmark (NAB) and Yahoo benchmarks with labeled datasets, which provide a benchmark for evaluating anomaly detection methodologies [18].

1.2. PROBLEM STATEMENT

Despite the advancements in ML techniques for anomaly detection in the context of univariate TS data, This field still accepts much research. Many real-world applications, especially in infrastructure monitoring, industrial logs, IoT sensors, and energy systems, still rely on univariate TS data anomaly detection, either due to hardware limitations or system design. These systems benefit from lightweight, interpretable models, making univariate anomaly detection practical and necessary in constrained environments. These systems rely solely on the past values of the TS itself and offer a more cost-effective and practical solution for real-time applications and dynamic control systems like solar energy prediction [19] [20], given the critical importance of TS anomaly detection across various domains. Although ML models like RF CatBoost and XGBoost have shown promise [21], there is a lack of systematic comparative studies assessing their performance for univariate TS anomaly detection. This research tries to fill this by utilizing and evaluating these models within the Darts framework in a comparative study exploring approaches that integrate statistical scoring techniques to improve accuracy and robustness in anomaly detection.

1.3. RESEARCH OBJECTIVES

The main objective of this study is to build and develop a complete framework for evaluating anomaly detection techniques for TS data under ML models. Specifically, we aim to achieve the following objectives:

1. Utilization and evaluation of regression, RF, CatBoost, and XGBoost models for anomaly detection in TS data to extract the results for accuracy and robustness assessment of these models within the Darts framework.

2. Explore and implement approaches integrating ML and anomaly detection statistical scoring methods for better results.

3. Propose and assess a comparative methodology for univariate TS forecasting and anomaly detection using various ML techniques.

1.4. SCOPE

This study focuses on implementing and evaluating four key models: Regression, RF, XGBoost, and CatBoost for anomaly detection in univariate TS data. The study employs three NAB datasets and one of Yahoo S5 anomaly detection Dataset, then leverages the Darts framework for its analysis and implementation. These kinds of methodologies are also presented; they integrate ML models with statistical scoring techniques such as NormScorer and WassersteinScorer to verify their efficiency in real-world scenarios. The scope is confined to evaluating model performance based on metrics such as accuracy and other model metrics like the Area Under Cover (AUC), which opens the door for academic research and applications in a multitude of domains. Restrict this research to univariate TS data, making it a limitation. Furthermore, the evaluation is limited to selected public datasets and may not generalize across all domains or production-level systems. Threshold selection for detection remains empirical and may require further tuning for domain-specific deployments.

1.5. SUMMARY

In this chapter, a brief introduction to anomaly detection in TS and its critical role in modern data-driven industries is given. This is followed by an introduction to some ML models, highlighting the advantages of ensemble models like RF and gradient-boosting algorithms such as XGBoost and CatBoost. Then, the evaluation of these models over benchmark datasets and using some techniques for persistent anomaly detection provides a comparative analysis. Also, the enhancements from the previous work that will be added to this study are given. The chapter ends with a description of the scope of the research. In the next chapter, we will introduce a literature review of the work, mention the works that are related to our model, and make a comparison between our model and theirs.

PART 2

LITERATURE REVIEW

This part of the study provides an overview of the fundamental concepts, challenges, and methodologies associated with TS anomaly detection. Starting with an explanation of TS data and anomalies in temporal patterns. Then, it mentioned the difficulties in detecting anomalies in univariate TS data. Followed by an exploration of relevant ML methods and the ML-based TS anomaly detection models. Lastly, a comparison of related works highlights the advantages, limitations, and practical applications of existing TS anomaly detection strategies across various domains.

2.1. TS ANOMALY DETECTION CONCEPT

The dataset is a collection of data instances or observations that have a unique pattern. The data can be a number, record, video, song, graph, image, event, and profile. Data used in anomaly detection needs to be transformed into general data types. So, the TS data refers to a sequence of data collected or recorded at successive points in time at uniform intervals [22]. This type of data captures how values change over time and often exhibits key characteristics such as trends (long-term increases or decreases), seasonality (recurring patterns at fixed intervals), and cyclic variations (patterns that repeat but not necessarily at fixed periods) [23]. TS data can be categorized based on the number of variables observed over time. In the case of univariate TS, each time point represents a single observed variable. Often used when monitoring a single specific sensor, metric, or stream of events. Univariate TS analysis may be straightforward but can reveal complex behavior nonetheless. It is invaluable when the interest is to monitor anomalies within a single measure over time.

The goal of TS data is to understand the past to predict the future based on historical data. However, unexpected deviations can mark significant events such as

system failures, fraudulent activities, or other abnormal behaviors. TS anomaly detection is widely used in many fields, such as finance for fraud detection, healthcare for identifying abnormal conditions, cybersecurity for detecting network intrusions, industrial maintenance for predicting equipment failures, and climate science for recognizing extreme weather patterns.

2.1.1. TS Anomaly Detection Challenges

Detecting anomalies in univariate TS data presents several challenges that can significantly impact the accuracy and reliability of anomaly detection models. One of these challenges is seasonality and trend components, which in Univariate TS data sometimes exhibit seasonality (regular, predictable patterns) and trends (long-term increases or decreases) [24]. Effectively distinguishing between these inherent patterns and original anomalies is complex, as failing to do so can lead to false positives or negatives [12]. Seasonality and trends are critical factors affecting TS anomalies. The second one is noise and data variability, which happens in TS data and is frequently distorted with random fluctuations. Distinguishing genuine abnormalities from this noise is a difficult task because, on the one hand, too high sensitivity can lead to false alarms, while on the other hand, too low sensitivity can cause you to miss big abnormalities [24]. Lack of labeled data is also considered a challenge because anomalies are rare and manual labeling is expensive; supervised anomaly detection techniques necessitate labeled datasets, which are frequently hard to come by in real-world situations [25]. Model training and validation are impeded by this scarcity, which may result in overfitting or poor performance. Concept drift is another challenge, which means the phenomenon that statistical features of TS data may change with time. Models trained on past data could lose efficacy as new trends show; that's why constant model updates are necessary to keep accuracy [26]. The last challenge is setting up suitable thresholds for anomaly detection, which is difficult but essential [27]. Overly lenient thresholds may have too many real anomalies, while too strict thresholds may mark normal variations as abnormalities.

2.2. ML IN ANOMALY DETECTION:

ML allows computers to modify their behaviors to improve their accuracy in solving problems. ML is to use artificial intelligence (AI) to enable computers to think and learn on their own. It has two learning approaches, which are supervised and unsupervised learning. Supervised learning is to use labeled data sets that are already designed to train the algorithm to predict the results correctly. Supervised learning can be divided into two categories, which are classification and regression [28]. Classification assigns the data to categories like foreigners and citizens, legal and illegal. Regression allows the machine to understand the relationship between dependent and independent variables. Regression methods are useful for predicting numerical values like stock price changes and sales revenue [29]. Unsupervised learning is to use ML algorithms to analyze and cluster unlabeled datasets. There are three main problems with using unsupervised learning, which are clustering, association, and dimensionality reduction. Clustering is a technique used to group unlabeled data depending on the similarities and differences. These algorithms can be used in fields like E-mail marketing, Retail marketing, and Streaming services [30]. Association algorithms are often used to define the relationships between two different variables in large datasets. It's widely used in market basket analysis to recommend that customers who buy a specific product or other products, like those who buy potatoes, will probably buy Ketchup. Dimensionality reduction algorithms are used when there is a large number of features in the dataset to reduce the number of data inputs so it will be easier to manage the data. ML has been used in many fields, such as neuroscience, computational complexity theory, Bayesian method, and control theory. ML proved efficiency in solving classification, regression, and anomaly detection problems, which we will focus on during the thesis, and this is under the regression type of supervised or semi-supervised learning.

2.2.1. ML Methods for TS Anomaly Detection

The early methods applied relied heavily on statistical methods like ARIMA (Autoregressive Integrated Moving Average) and seasonal-trend decomposition (STL) [31]. These techniques model series data by finding any of its underlying patterns and

trends, allowing deviation from expected behavior, that is anomaly detection. However, such simple models have inherent limitations with nonlinear behavior and high dimensions, largely prevalent in practical applications. The modern approaches in machine vision to solving these challenges include ML methods. Among the earliest applications of ML approaches in TS anomaly detection is the regression model [32], which has been used to model trends and deviations.

RF is strong among the many techniques employed in ensemble learning, and due to combining results from multiple decision trees, it provides better performance through its capacity to capture complex patterns [33]. It can decipher linear and nonlinear relationships, making it a friend in disguise for anomaly detection. Gradient-boosting methods like XGBoost and CatBoost have also gained tremendous fame owing to the fantastic predictive accuracies and efficacies they provide. As such, XGBoost becomes even more appealing in this case, coupled with the regularizing optimization provided by CatBoost in terms of categorical data, making it well-suited for many real-world anomaly detection tasks.

Ensemble methods, XGBoost and CatBoost, have become some of the most effective techniques in TS anomaly detection due to their robustness, which can be scaled to very large datasets [34]. These were configured as augmented decision trees by gradient-boosting methods, which led to improved accuracy in the tasks assuming a high-precision need. The models, indeed, have proven their superiority over the traditional methods of detection accuracy in such schemes with the feature selection and thresholding accepted. Meanwhile, challenges still exist in optimizing the models for real-time and managing datasets with multiple temporal dependencies.

Recent advancements that deal with TS anomaly detection are moving rapidly toward neural networks such as Long Short-Term Memory (LSTM) networks and Temporal Fusion Transformers (TFT) [35]. While these methods can capture temporal dependencies and complex patterns [36], they often require extensive computational resources and may not generalize well to smaller datasets. Furthermore, self-supervised learning (SSL) has emerged as a powerful alternative [37]. Consequently,

ensemble models like RF, XGBoost, and CatBoost remain attractive choices due to their balance of performance and computational efficiency.

This thesis builds upon these studies by comparing the accuracy of Regression, RF, XGBoost, and CatBoost models in detecting anomalies on multiple datasets from the NAB benchmark and Yahoo anomaly detection using the darts framework. The findings will contribute to the growing body of research by identifying optimal models or approaches for effective anomaly detection in time-series data.

2.3. RELATED WORK

Many studies have investigated various approaches for TS anomaly detection across different domains. The methods used in these studies range from traditional statistical techniques to modern ML and DL approaches. Providing a structured comparison. The related studies are mentioned with the title for each one, and the details are as follows.

2.3.1. Anomaly Detection in Railway Sensor Data Environments

This paper discussed the advancements in railway anomaly detection using real-time data analytics, ML, and the Internet Of Things (IoT). The study tried to minimize the challenges in the industry due to the vast infrastructure, the dynamic conditions, and the aging of infrastructure. The study covered three main objectives: identifying TS anomaly detection methods applied to railway sensor devices, recognizing the advantages and disadvantages of each method, and evaluating its effectiveness. The results demonstrate that the highest classification accuracy was achieved by CatBoost with 96% accuracy, followed by RF at 91% and XGBoost algorithms at 90%. Conversely, One-Class Support Vector Machines exhibited the lowest performance with merely 48% accuracy, while Local Outlier Factor and Isolation Forest methods yielded suboptimal results of 53% and 55%, respectively [38].

2.3.2. ML Methods for the Prediction of Wastewater Treatment Efficiency and Anomaly Classification with Lack of Historical Data

The study discovers ML techniques for regression tasks to predict wastewater treatment efficiency and classify anomalies, especially for preventing emergencies with limited historical data. The authors evaluate many models, assessing their performance in handling complex, multivariate TS data. The findings highlight the potential of these models in accurately predicting outcomes and identifying anomalies. CatBoost has shown competitive results, even when historical data is available [39].

2.3.3. An Improved Anomaly Detection Model for IoT Security Using Decision Tree and GB

In this study, the authors give an enhanced intrusion detection system model for securing IoT networks using ML techniques. The approach uses a GB algorithm, CatBoost, to improve detection accuracy and efficiency. The model is evaluated across multiple datasets. Experimental results, accelerated using GPU-based computation, show that the proposed model achieves high performance with accuracy, recall, and precision metrics reaching up to 99.9%, making it highly effective for detecting anomalies in IoT environments. Shows the potential of GB techniques in enhancing IDS frameworks, especially in the context of time-sensitive and resource-constrained IoT systems [40].

2.3.4. Anomaly Detection in Univariate TS: A Survey on the State-of-the-Art

The study focuses on anomaly detection in TS data. It categorizes its methods into methods that focus on statistical approaches, methods that use neural networks, and methods that use ML and Deep Learning (DL). The study covers over 20 methods of these categories. It provides a benchmark for them regarding the accuracy and the computation time for each method by using publicly available datasets to help select the most appropriate method for different types of time-series data [5].

2.3.5. ML Methods for Anomaly Detection in Industrial Control Systems

This paper investigates various ML approaches for anomaly detection in industrial control systems, particularly in the domain of cyber-attacks. Using models like RF,

GB machine, artificial neural networks, and LSTM classifiers. The study suggests that while all models perform well, RF, with optimized hyperparameters, outperforms others in accuracy [41]. Table 1 summarizes key related studies, focusing on the employed approaches, advantages, disadvantages, and additional details.

Table 1. Related works.

Paper	Approaches	Advantages	Disadvantages	Details
[38]	Supervised and Unsupervised ML methods - Hybrid statistical -ML approaches.	- Handles noisy sensor data. - Compares multiple methods.	- Requires labeled anomalies for supervised methods. - Computational cost	- Evaluates accuracy, F1-score, precision, recall. Using railway datasets (vibration, temperature, pressure sensors).
[39]	- Regression models, XGBoost. - Semi-supervised anomaly detection (One-Class SVM). - Transfer learning	- Works with limited historical data. - Interpretable results - Predicts efficiency + classifies anomalies.	- May overfit to specific plants. - Balancing prediction and anomaly detection tasks are complex.	- Uses chemical sensor data - Evaluate RMSE for efficiency prediction and AUC-ROC for anomaly classification.
[40]	-CatBoost (Gradient Boosting + Decision Trees) - Compared with SVMs and other boosting methods	- Achieves 99% accuracy, recall, and precision - Optimized for GPU for faster training	-Focuses mainly on classification, not TS -Tailored to intrusion detection, not general anomaly use	Uses NSL-KDD, IoT-23, BoT-IoT, Edge-IIoT datasets Highlights ML-based IDS potential in IoT
[5]	Statistical, ML and DL approaches for univariate TS anomaly detection.	Comprehensive comparative evaluation of 20+ methods; highlights strengths and limitations.	Was general didn't specify	Evaluate accuracy and AUC values Used different data characteristics
[41]	- RF - GB	- High performance on ICS attacks	- Focused on ICS only	Uses Augmented ICS dataset; RF with

	- ANN - LSTM	- Useful in cybersecurity for critical infrastructure	- Lack of Interpretability	hyperparameter tuning performed best.

The reviewed studies highlight various TS anomaly detection techniques, showing the strengths and balance of statistical, ML, and DL approaches. While statistical methods like ARIMA and Prophet are suitable for structured seasonal data, they often struggle with complex, high-dimensional datasets [42]. DL models like LSTM and Transformers are powerful in capturing temporal dependencies but are computationally expensive and require large, labeled datasets [4,43].

PART 3

PROPOSED APPROACH AND METHODOLOGY

In this section, full details of the study for anomaly detection in univariate TS data will be introduced, covering the entire workflow, starting with the structure of the proposed approach, from data preprocessing to model evaluation using metrics and comparative analysis methodology to assess the performance of the models. More details about the models and scoring mechanism were explained in the methodology. Then, the experimental setting explained the datasets and hyperparameters and moved to the evaluation methods in detail.

3.1. PROPOSED APPROACH

The proposed approach adopted in this study provides a structured framework for assessing how well ML models perform in detecting anomalies within univariate TS data. Since spotting anomalies in data that change over time can be quite challenging, it's crucial to have a solid experimental setup to ensure we can accurately compare the different models. This research focuses on four ML models: Regression, XGBoost, CatBoost, and RF, chosen for their capability to manage time-related patterns and their success in previous anomaly detection studies [44]. The experimental framework was implemented using the Darts library, which provides specialized tools and a unified interface for TS analysis [45]. The methodology consisted of three main phases, as illustrated in Figure 1:

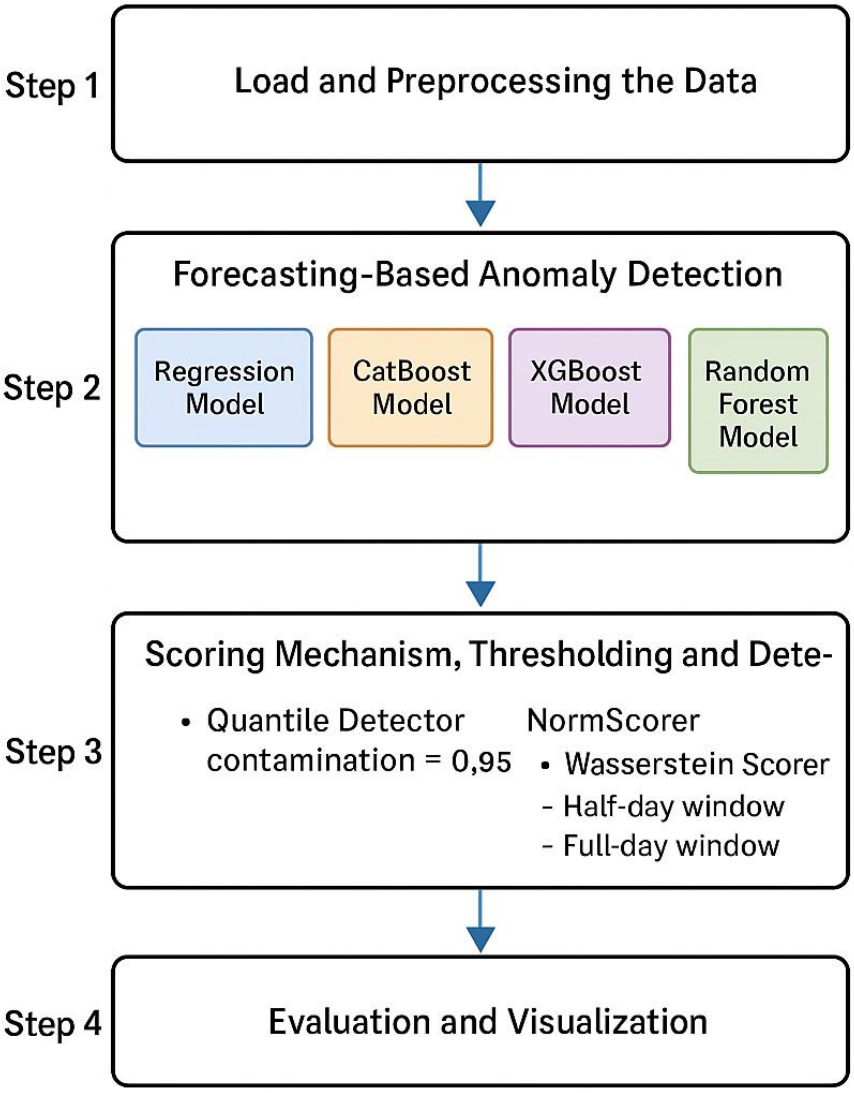

Figure 1. The proposed framework for TS anomaly detection using ML models.

3.1.1. Data Loading and Preprocessing

In step 1 of Figure 1, the publicly available NAB and Yahoo S5 Datasets were utilized for this study in the first step. The dataset consists of the timestamp recorded at regular frequency intervals (freq=30 min, freq=5 min, or freq=1 hour) and the value number. The dataset spans a defined temporal period and serves as a benchmark for evaluating anomaly detection methodologies. Preprocessing involves resampling the data to a regular frequency, like 5 min or 1 hour, starting at the original timestamp, then filling in the missing value if there is any. The next step is defining start and end dates for the known anomalies that we will use on the models. This is a crucial step in creating a series with binary anomaly flags. Lastly, plotting the data with the anomalies and

visualizing it for analysis to ensure it is clean and structured for effective ML model training. Then, split it into training and testing sets so the next step can be implemented without any problem. Table 2 shows the first five rows of the New York City (NYC) Taxi dataset, aggregating the total number of taxi passengers in New York City into 30-minute intervals ready to use [46].

Table 2. Sample records from the NYC Taxi dataset

Timestamp	Value
2014-07-01 00:00:00	10844.0
2014-07-01 00:30:00	8127.0
2014-07-01 01:00:00	6210.0
2014-07-01 01:30:00	4656.0
2014-07-01 02:00:00	3820.0

3.1.2. Forecasting-Based Anomaly Detection

As shown in Figure Step 2, the TS anomaly detection approach is based on forecasting techniques. Each ML model is trained to predict future TS values utilizing historical data and covariates. The value predicted and the value observed are considered on the same scale, which leads to anomaly detection. The implemented supervised ML models comprise Regression, CatBoost, XGBoost, and RF models, which are explained in the methodology section in detail.

3.1.3. Scoring Mechanism, Thresholding and Detection

Step 3 involves the quantification of deviations between model predictions and actual observations using the Forecasting Anomaly Model class from the Darts library. This helps detect anomalies using three scorers: one of the norm scorers to capture immediate anomalies and two of the WassersteinScorer for long-term and short-term to detect transient and persistent anomalies. For the thresholding, a Quantile detector was used to convert anomaly scores into a binary anomaly with the 95th percentile applied.

3.1.4. Evaluation and Visualization

The evaluation process incorporates accuracy and AUC metrics for comprehensive assessment, with results validated by visualizing predicted anomaly scores against the original TS data. The detected anomalies undergo rigorous comparison with predefined ground truth anomalies in the dataset.

3.2. METHODOLOGY

This chapter outlines the methodological framework developed for detecting anomalies in univariate time series data using ML techniques. Covered model explanations and selection in addition to the scoring and thresholding mechanisms.

This study utilized four forecasting-based anomaly detection models: Regression, CatBoost, XGBoost, and RF. Starting with the Regression model,
A traditional regression technique predicts the target variable based on lagged values, past observations, and external covariates, providing a flexible integration of ML algorithms into TS analysis through the Darts framework [47]. Selected due to handling nonlinear relationships. Unlike simple statistical models and works with covariates, it is fast and useful in labeled datasets. The second model is the CatBoost, a GB algorithm developed by Yandex, built on decision trees and particularly useful for TS forecasting and anomaly detection [48]. Selected for its capability to handle categorical characteristics and minimal preprocessing requirements, it demonstrates high efficiency in TS prediction. And XGBoost is a powerful ML algorithm based on decision trees [49]. Implementing a gradient-boosting ML approach generates weak sequential models that iteratively improve predictions through optimization techniques, effectively handling large, complex datasets [50]. The algorithm processes numeric, categorical, and missing values while an objective function is being optimized using gradient descent. Its efficiency has been demonstrated in various real-world ML applications, such as univariate and multivariate TS forecasting with integrated regularization for overfitting prevention. The fourth model is the RF model, an ensemble learning method based on decision trees; it builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting [51].

Exhibits advantages in both univariate and multivariate TS forecasting through its inherent characteristics: minimal sensitivity to noise and outliers, high accuracy, robustness, feature importance capabilities, adaptability, scalability, and overfitting reduction through tree averaging [52]. Table 3 below compares the explained models regarding Target Series Support: Univariate and Multivariate, Covariates Support: Past-observed and Future-known/ Static, and Probabilistic Forecasting: Sampled/ Distribution Parameters.

Table 3. ML models comparison

Model	Target Series Support: Univariate/ Multivariate	Covariates Support: Past-observed/ Future-known/ Static	Probabilistic Forecasting: Sampled/ Distribution Parameters
Regression Model	✓/✓	✓/✓	✗/✗
RF	✓/✓	✓/✓	✗/✗
CatBoost Model	✓/✓	✓/✓	✓/✓
XGB Model	✓/✓	✓/✓	✓/✓

In this study, four ML models, Regression, RF, CatBoost, and XGBoost, were selected for TS anomaly detection due to their effectiveness in handling structured temporal data. The models were evaluated one by one based on their target series support, covariate handling, and probabilistic forecasting capabilities. All selected models support univariate and multivariate TS, making them suitable for analyzing different types of temporal datasets. Also, they can incorporate the past observed and future covariates, allowing for a more contextualized anomaly detection process by using historical trends and external influencing factors. On the other hand, only CatBoost and XGBoost support probabilistic forecasting, which enables them to predict uncertainty and provide confidence intervals for anomaly predictions. Despite this, RF and standard Regression models are still valuable due to their interpretability and

robustness in capturing complex patterns without requiring extensive hyperparameter tuning. The combination of these models provides a balanced approach, ensuring both performance efficiency and scalability and making them good for anomaly detection in univariate TS data, which is the focus of this research.

After training each ML model to forecast values based on historical univariate time series data, the next phase involves identifying anomalies by analyzing the discrepancy between predicted and actual values. Using scoring mechanisms, each offering different sensitivity levels and temporal aggregation strategies, including the Norm Scorer, which calculates the L1 norm (absolute deviation) between predicted and actual values, and the Wasserstein Scorer, measuring anomalies based on Wasserstein distance. The implementation uses two temporal configurations: the first window for detecting localized deviations and the second window for identifying persistent anomalies through longer-interval aggregation. Also, The Quantile Detector applies a thresholding mechanism at the 95th percentile after the scoring, focusing on the most significant anomalies. This approach converts anomaly scores into binary predictions (1 for anomalies, 0 for regular observations), as shown in Figure 2, which are evaluated against ground truth data.

Figure 2. Function of the detector.

3.2.1. Summary

figure 3 illustrates a process summary of anomaly detection in TS data using ML models. It consists of multiple stages, starting with input TS data. The anomaly detection model, which uses an ML forecasting model, for example, XGBoost, is the core component that learns from past trends and predicts expected values. Then, The anomaly scoring step compares the predicted values to the actual values, calculating

residuals or probability-based thresholds to detect anomalies. Finally, the detected anomalies are flagged and visualized, allowing for interpretation and further action.

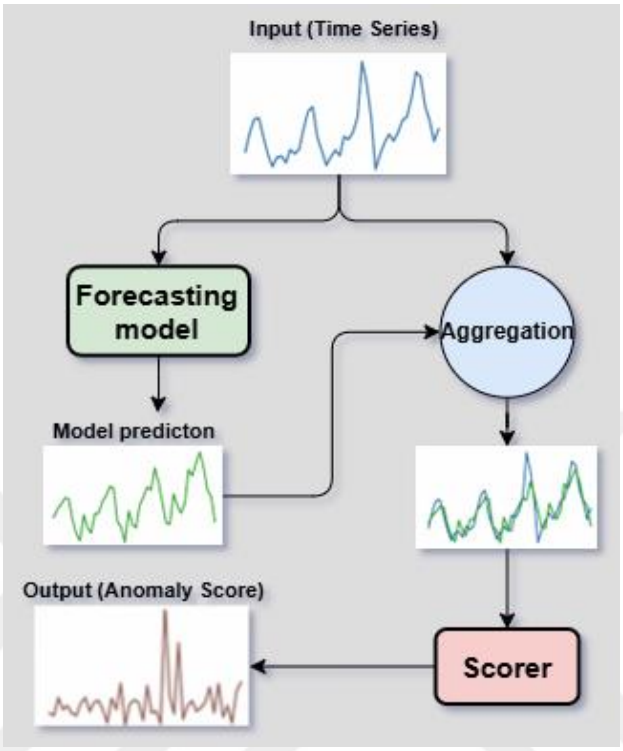

Figure 3. The structure of forecasting anomaly detection model.

3.3. EXPERIMENTAL SETTING

This section introduces the dataset used and preprocessing techniques applied to clean and transform the data in the study with details, then the hyperparameters of the developed model are explained, and information about the performance evaluation metrics is provided. This well-defined experimental framework enables a fair comparison of the selected models to ensure their effectiveness in anomaly detection is rigorously tested.

3.3.1. Dataset

The experiments in this study were conducted using four publicly available datasets, three from NAB: the NYC Taxi Dataset, the CPU Utilization Dataset, the Ambient Temperature Dataset [53], and one from Yahoo S5 anomaly detection labeled dataset

[54], which are widely used for evaluating anomaly detection algorithms. Each dataset is TS data with known anomalies, which can be considered the ground truth to evaluate the performance of the anomaly detection models. As such, these contain real-world and synthetic TS data with labeled anomalies, enabling accurate validation of models. The datasets have diverse characteristics, such as dissimilar seasonality, trend components, and noise levels; in this way, we can evaluate the generalization capacities of the models and maintain comparability. All datasets are preprocessed and standardized. The raw datasets were first converted into TS objects using the Darts framework to ensure compatibility with all models. Also, missing values were dealt with by forward-fill interpolation whenever needed in order to preserve TS continuity. Additionally, the datasets were resampled to regular time intervals where needed, ensuring that all timestamps were uniformly spaced. Feature engineering was done by adding the hour of the day and day of the week cyclic temporal features to capture seasonality and periodicity, encoded as covariates. The datasets were then split into training and testing sets. The training set fits the models, while the testing set is reserved for scoring anomalies and evaluation.

The NYC Taxi Dataset represents TS data of taxi passenger counts in New York City recorded at 30-minute intervals. The dataset includes known anomalies corresponding to events that significantly impact taxi demand, the NYC Marathon (November 2, 2014), Thanksgiving (November 27, 2014), Christmas (December 24-25, 2014), New Year's Eve (December 31, 2014 - January 1, 2015), and a Snow Blizzard (January 26-27, 2015) [55]. These anomalies are the ground truth for evaluating the performance of anomaly detection models, as shown in Figure 4. The dataset was then divided into a training set comprising the first 4,500 data points to ensure the dataset was anomaly-free, enabling the model to learn normal patterns without interference, and a test set containing the remaining data points.

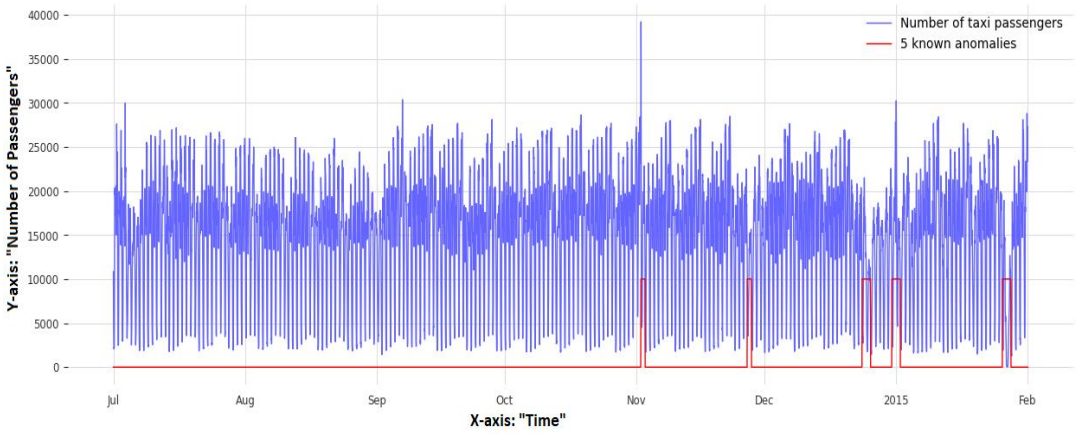

Figure 4. Temporal distribution of known anomalies in the NYC taxi dataset with highlighted special events.

The CPU Utilization Dataset tracks CPU utilization in a cloud-hosted application environment, where anomalies indicate high unexpected usage of CPU, with 5-min frequency. It also received anomalous events on (July 12, 2014, 2:04:00 AM) and (July 14, 2014, 9:44:00 PM) as seen in Figure 5. The dataset was split into a training set consisting of the first 9,000 data points and a test set with the remaining points to evaluate detection accuracy.

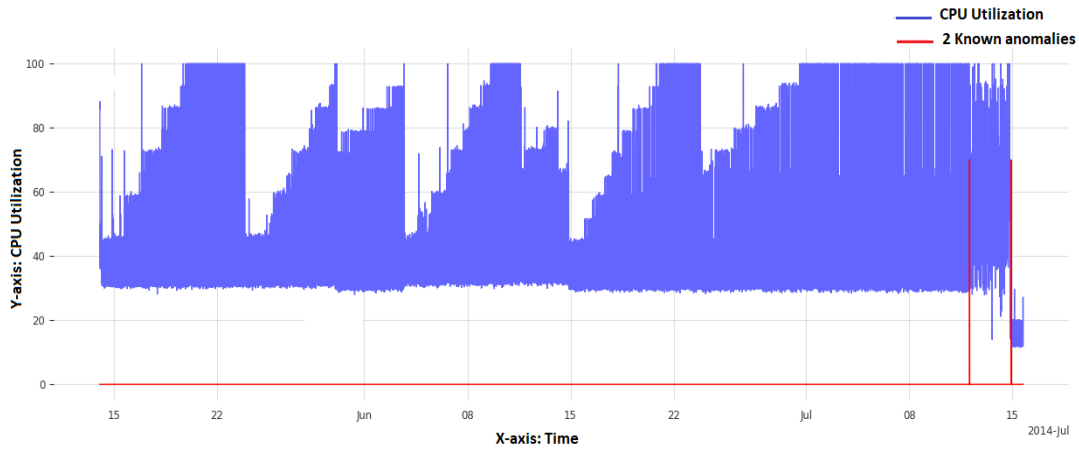

Figure 5. Temporal distribution of known anomalies in the CPU utilization ASG misconfiguration dataset with highlighted special events.

The Ambient Temperature Dataset represents ambient temperature readings from a system where anomalies signify potential failures or irregularities in the system. However, two anomalies in the data, December 22, 2013 (4:00 PM to 5:00 AM) and

April 13, 2014 (6:00 AM to 12:00 PM), are represented in Figure 6. The data was divided into a training set containing the first 3,500 points and a test set with the remaining points, with a frequency of 1 hour.

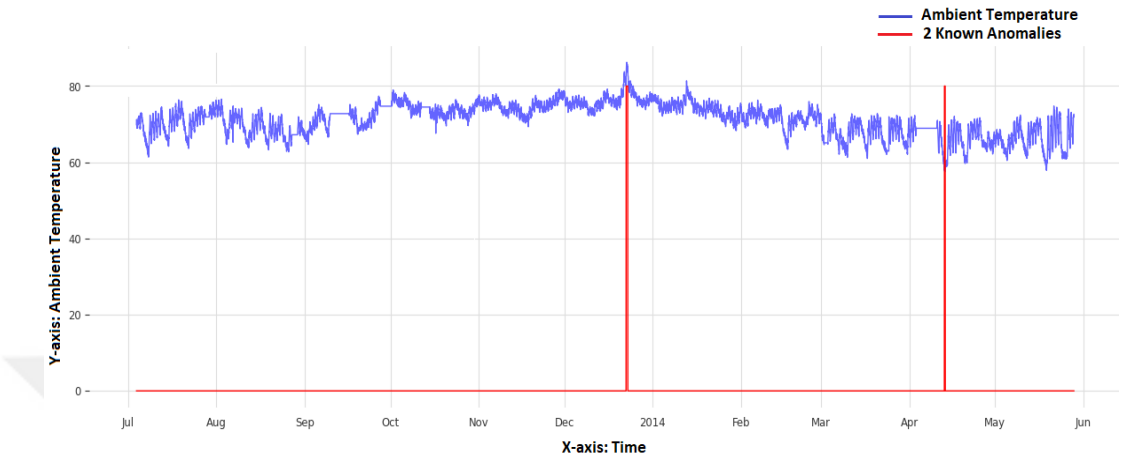

Figure 6. Temporal distribution of known anomalies in the ambient temperature system failure dataset with highlighted special events.

Yahoo S5 Synthetic Dataset: A synthetic dataset with known anomalies occurring at specific timestamps (e.g., July 13, 2014, 11:30 PM - 12:30 AM, July 14, 2014, 5:00 PM - 6:00 PM, and July 19, 2014, 2:30 PM - 3:30 PM) shown in figure 7. This dataset allows testing on controlled synthetic anomalies. Divided into 500 training and the rest testing set with a frequency of 30 min.

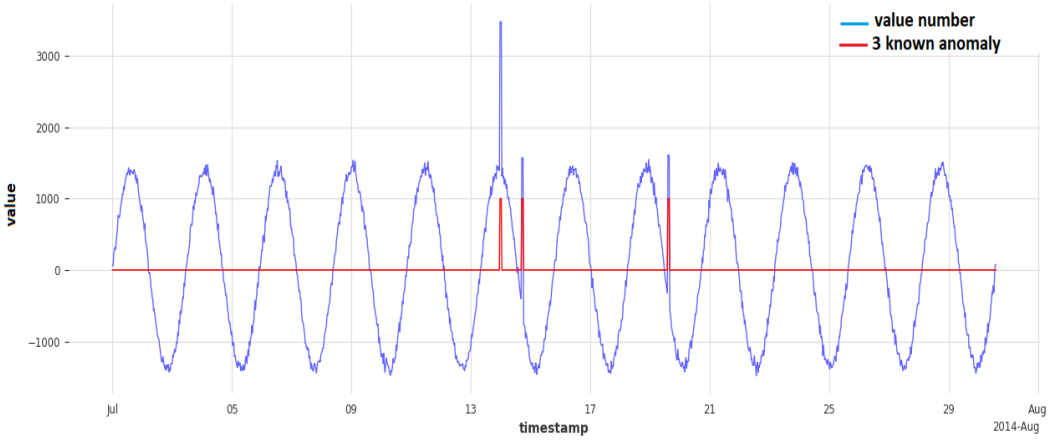

Figure 7. Temporal distribution of known anomalies in the Yahoo S5 dataset with highlighted special events.

3.3.2. Hyperparameters

In the experimental setup, hyperparameters, as Table 4 shows, are finely tuned for the models for each dataset to improve anomaly detection performance across the four datasets. Some models used a lag window of one week to capture temporal patterns effectively, and some used a lag window of one day regarding their unique patterns. The lag window was set to 7 days * 24 hours * 2 intervals (for 30-minute data) for the NYC Taxi dataset, 1 days * 24 hours * 12 intervals (for 5-minute data) for the CPU Utilization dataset, 7 days * 24 hours (for hourly data) for the Ambient Temperature and 1 days * 24 hours * 12 intervals (for 5-minute data) for the Yahoo S5 A2 dataset. Cyclic encoders are added to provide features like the hour and day of the week to reflect the repeating periodic pattern that the dataset reveals. The Regression Model similarly used covariates to maintain consistency. By adding the same temporal and cyclic covariates as the other models, the Regression model provided a baseline for comparison against the other sophisticated ensemble models. The XGBoost and CatBoost models were run with default settings for the boosting algorithms, and they included the same set of covariates and lag structure as the other models. The RF was set to run with 200 decision trees, finding a good compromise between computation time and the accuracy of prediction, while the criterion for splitting focused on minimizing absolute error because it was less sensitive to outliers and thus quite apt for anomaly detection.

Table 4. Hyperparameters of the models and their descriptions.

Hyperparameter	Type/Value	Description
lags	One week or one day	Calculation of one week or day (depending on the dataset's frequency intervals) specifies historical lags used for forecasting.
lags_future_covariates	[0]	Indicates future covariates (e.g., time-based features) used for prediction.
output_chunk_length	1	Defines the length of the output prediction chunk.

add_encoders	cyclic:{"future": ["hour," "day of the week"]}	Specifies encoders for additional features (e.g., categorical or temporal encodings).
n_estimators	200	Number of trees in Random Forest model.
criterion	absolute_error	The loss function used for the RandomForest model.

The hyperparameters were optimized to maximize model performance while ensuring computational efficiency. The implementation uses dataset-specific lag window sizes, allowing you to set a relevant context of historical events leading to the prediction, and cyclical encoders capture periodicity in the taxi demand dataset as an example. The analysis employs the Darts ForecastingAnomalyModel class for anomaly identification using comparative analysis of actual values and model predictions. The framework implements three different scoring mechanisms for residual analysis: one is the NormScorer, and two is the WassersteinScorer, which works differently from the NormScorer by considering the distribution of residuals over a window rather than point-wise errors. It is useful for detecting pattern anomalies over time rather than just isolated deviations. As follows:

1. NormScorer, calculating point-wise anomaly scores via L1 norm for immediate deviation detection.
2. WassersteinScorer with a certain window or period regarding the dataset's requirement. Used for anomaly detection and short-term temporal pattern sensitivity.
3. WassersteinScorer with window long period score aggregation and persistent anomaly identification.

In addition, a QuantileDetector system is used to translate continuous anomaly scores into binary predictions within the analytical framework. It uses a high quantile threshold (0.95), enabling the identification of significant deviations by flagging the top 5% of scores as anomalies while maintaining optimal false positive rate control. Integrating scoring mechanisms with detection thresholds establishes a comprehensive approach to anomaly detection within the temporal data structure.

3.3.3. Evaluation Methods

This study evaluated the performance of anomaly detection models using a combination of quantitative metrics and visual analysis. Two key metrics were employed: Accuracy and AUC, which provide complementary perspectives on the performance of the anomaly detection models across the datasets. Accuracy measures the proportion of correctly classified instances (both anomalies and normal observations) out of the total number of instances. It serves as a straightforward and interpretable metric for evaluating model performance, particularly when the dataset is balanced. It is calculated as given in equation 3.1, where true positives and true negatives are the correctly identified anomalies and normal data points.

$$\text{Accuracy} = \frac{True\ Positives + True\ Negatives}{Total\ Instances} \tag{3.1}$$

AUC (Area Under the Curve) is the second method we used for evaluation represented in Two AUC-based metrics were used to evaluate the quality of anomaly detection:
AUC-ROC (Receiver Operating Characteristic Curve): This measures the model's ability to differentiate between normal and anomalous instances by assessing the trade-off between the true positive rate (sensitivity) and the false positive rate. A higher AUC-ROC indicates better model performance. AUC-PR (Precision-Recall Curve): This evaluates the precision-recall trade-off, which is particularly valuable for imbalanced datasets where anomalies constitute a small proportion of the total instances. A higher AUC-PR reflects the model's capability to detect anomalies while minimizing false positives.

The visualization process was also a part of the evaluation here, using visual analysis to validate the detection results, where anomaly scores were plotted with the original TS data to validate the detection results. This provides an intuitive understanding of the model's performance and its ability to align with the ground truth anomalies. These plots serve multiple purposes, such as the verification of anomaly detection; by

overlaying the detected anomalies on the original series, one can visually assess whether the flagged anomalies correspond to significant deviations from normal behavior. Temporal Analysis, which is Visualization, helps in understanding how anomalies are distributed over time and whether the model captures both sudden and gradual changes. Model comparison represents Graphs that facilitate direct visual comparison between different models, revealing strengths and weaknesses that might not be immediately apparent through numerical metrics alone.

PART 4

DISCUSSION AND RESULTS

This part presents the results obtained from implementing ML techniques for univariate TS anomaly detection. And discuss the performance of the models used for univariate TS anomaly detection using multiple datasets by comparing the models across different evaluation metrics. It also presents the training and validation processes supported by graphical results. In addition, the performance results and evaluation of all models are based on the metrics. It also discusses the impact of dataset characteristics, hyperparameter tuning, and threshold selection on model performance.

4.1. FORECASTING MODEL IMPLEMENTATION

In this section, the predictions of the four models are observed over time. The plots visualize both the actual number of taxi passengers (black) and the model's predictions (green) over time. The forecasting approach relies on lags of one week, meaning that the forecasting model uses past values from a week prior to predicting the next step [56]. Figure 8 illustrates the prediction results of a basic regression model applied to one of our datasets, for example, the NYC taxi dataset.

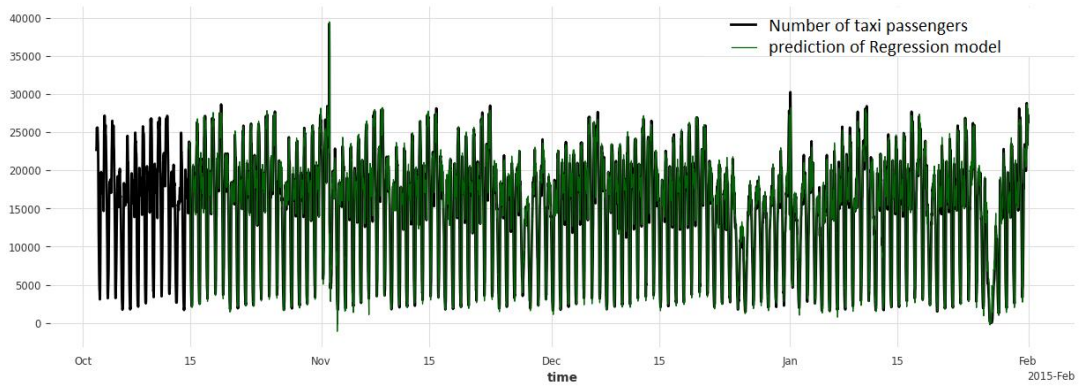

Figure 8. Forecasting regression model performance: predictions vs. actual values for the NYC taxi dataset.

Figure 9 displays the prediction results using the RF model and its forecasting future values capturing temporal dependencies.

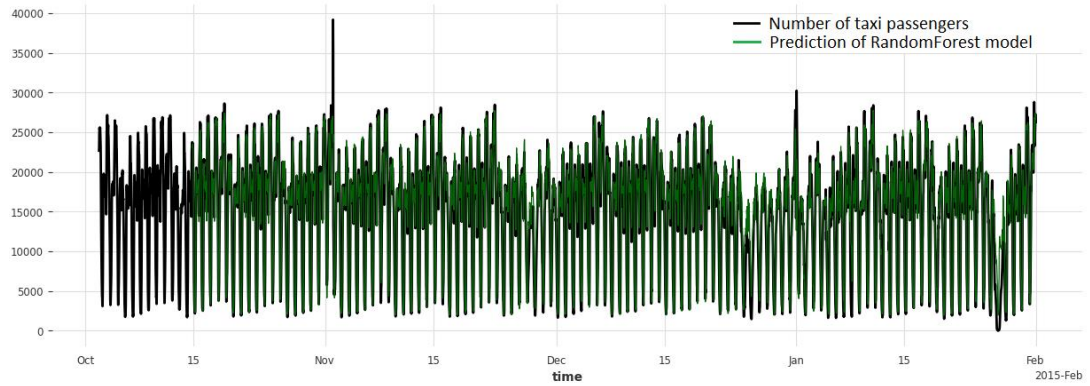

Figure 9. RF forecasting model performance: predictions vs. actual values for NYC taxi dataset.

Figure 10 illustrates the results of the CatBoost forecasting model. The predicted values follow the actual values closely, too.

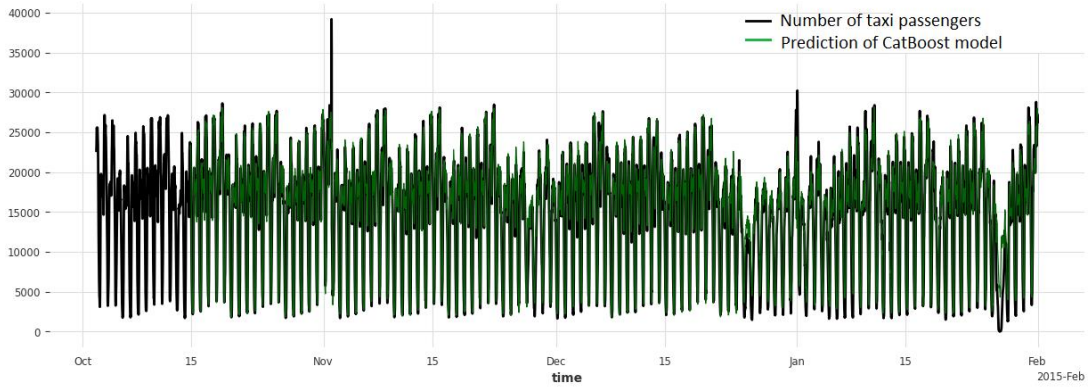

Figure 10. CatBoost forecasting model performance: predictions vs. actual values for the NYC taxi dataset.

The final Figure 11, shows the XGBoost model's predictive performance. Like CatBoost, RF gave a good predictive performance but not as good as the regression model.

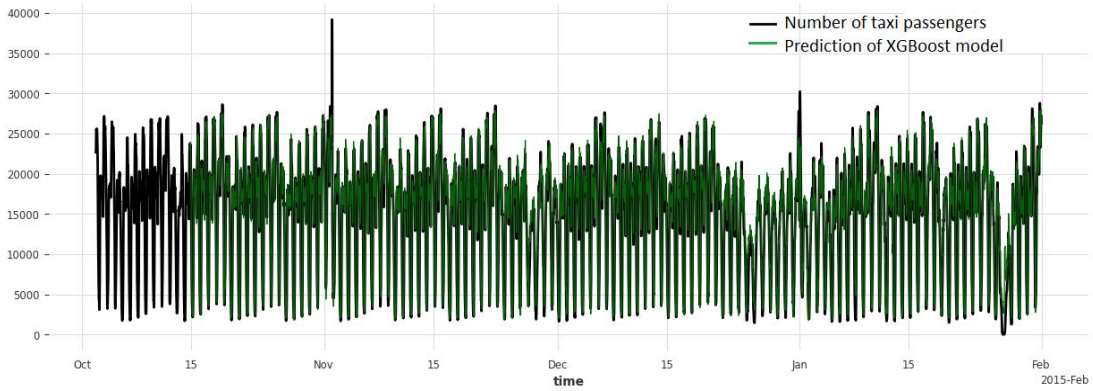

Figure 11. XGBoost forecasting model performance: predictions vs. actual values for NYC taxi dataset.

Visually, all models appear to track the seasonal patterns in the data reasonably well, with slight deviations around peaks. The slight variations between the actual data and the predictions indicate that these models are very powerful in handling TS data on the NYC taxi dataset and the other proposed datasets. But interestingly, regression models seem to produce smoother predictions in most of the datasets that closely follow the trend more than the other models like XGBoost and CatBoost, which sometimes display more variance or slightly delayed responses to sharp changes. We can see that Figure 12 shows the regression model performing well, capturing the actual value in a zoomed-in view on a spik.

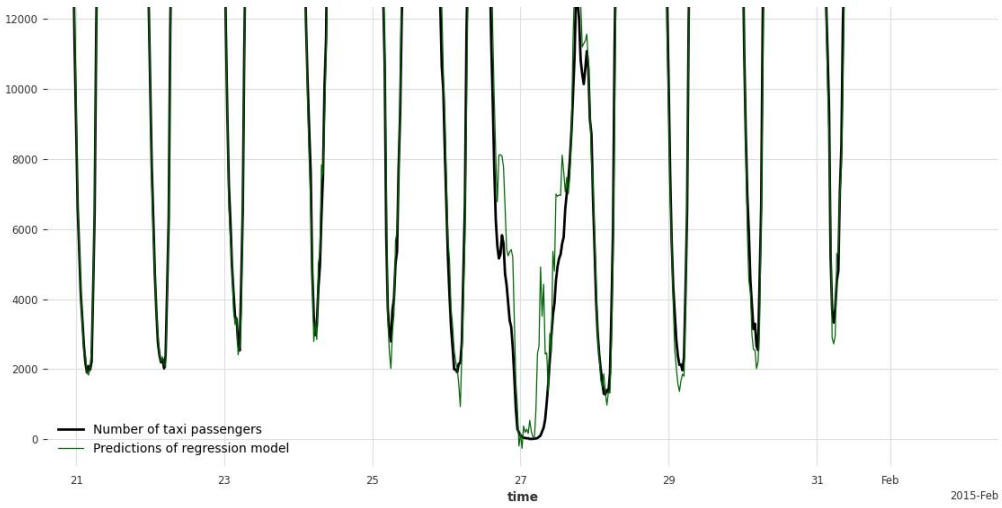

Figure 12. Zoomed-in view of a spike event: Actual vs. predictions for regression model on NYC taxi dataset.

Figure 13 shows the CatBoost model, which performed slightly less than the regression model. Noted, the other two models performed like CatBoost.

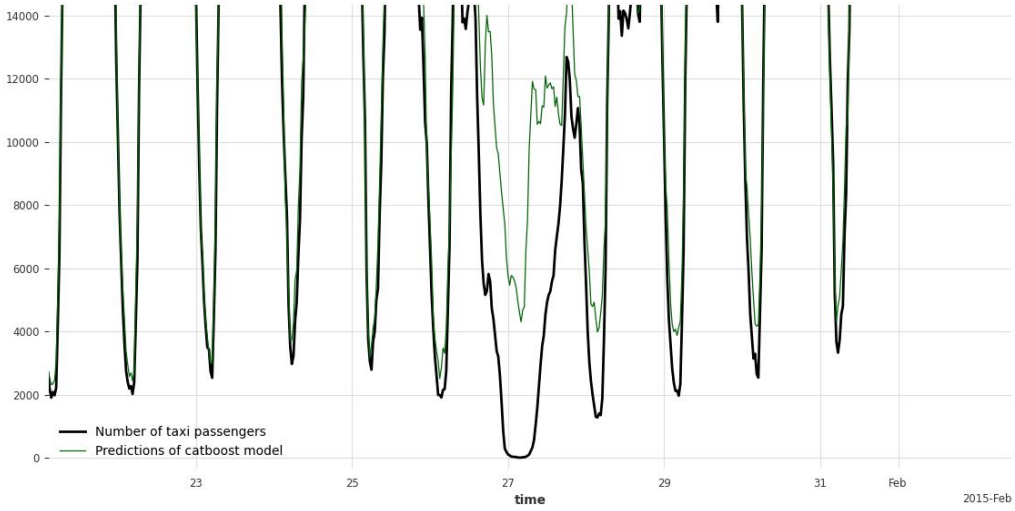

Figure 13. Zoomed-in view of a spike event: Actual vs. predictions for CatBoost model on NYC taxi dataset.

The regression model performed slightly better across some of the datasets, like NAB datasets, captured trends and seasonal patterns, and was closely aligned with the actual values. The data followed a well-defined seasonality and gradual trends over time. However, in the Yahoo dataset, RF and XGBoost performed better with strong periodic (cyclical) patterns in the spikes. This distinction can be attributed to the specific characteristics of the Yahoo S5 dataset. Figure 14 shows the forecasting performance of the regression model zoomed in on the spike.

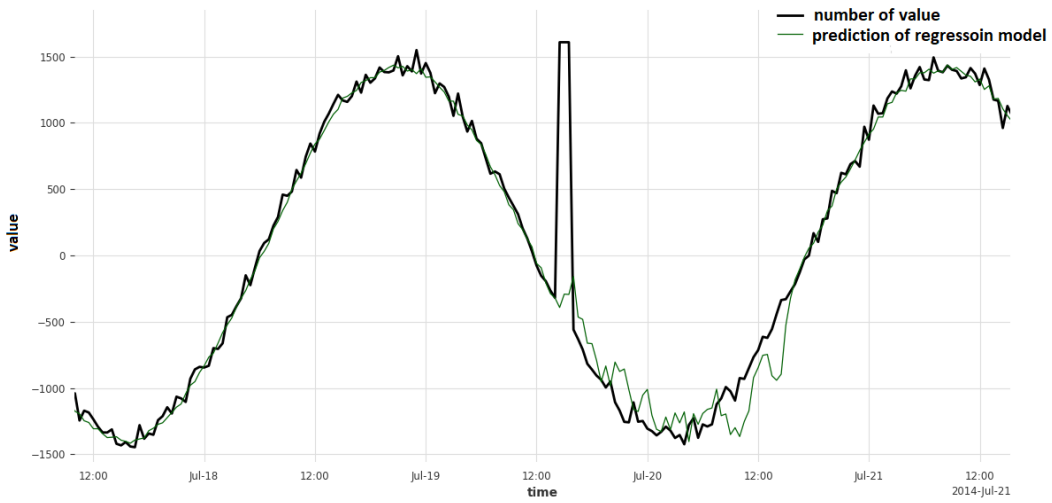

Figure 14. Zoomed-in view of a spike event: Actual vs. predictions for regression model on Yahoo S5 dataset.

Figure 15 shows in a zoomed-in view of a spike how the RF captured the actual value well.

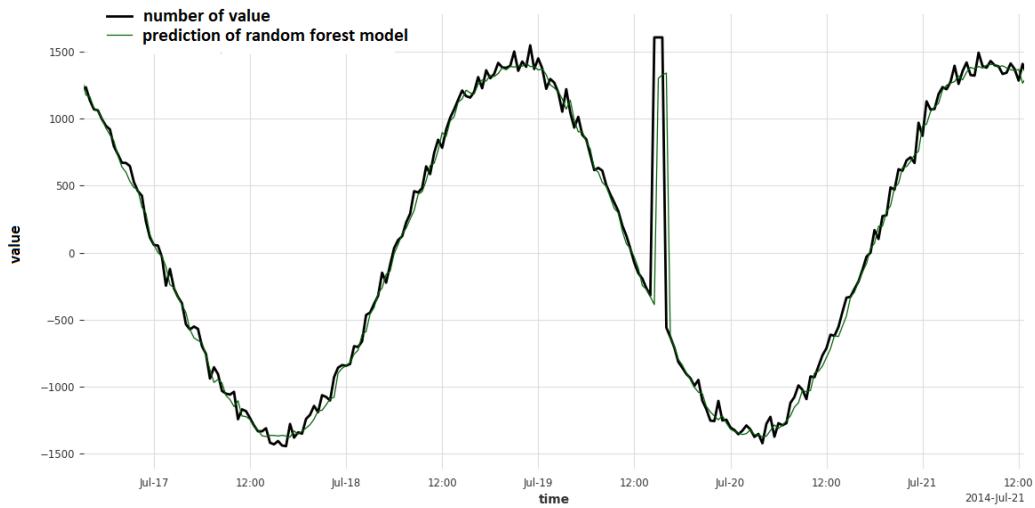

Figure 15. Zoomed-in view of a spike event: Actual vs. predictions for RF model on Yahoo S5 dataset.

4.2. AUC METRIC AND SCORER EVALUATION

AUC-ROC and AUC-PR metrics are used under three distinct scoring methodologies: Norm (order=1) with a unit window size, WassersteinScorer short-term window, and

WassersteinScorer with a long-term window. We evaluated the comparative analysis of four forecasting models, Regression, RF, CatBoost, and XGBoost, across the distinct datasets: NYC Taxi, CPU Utilization, Ambient Temperature, and Yahoo S2. The performance analysis reveals significant variations among the implemented models. GB methods (XGBoost and CatBoost) outperform across multiple evaluation metrics. While the Regression model demonstrates moderate initial performance under Norm scoring, its capability improves when leveraging larger temporal contexts through expanded window sizes. This underscores the impact of temporal dependencies in enhancing anomaly detection accuracy.

Table 5 shows the performance of the following models on the NYC Taxi and Ambient Temperature dataset. The models were assessed using AUC-ROC and AUC-PR metrics under Norm (order=1) with a unit window, WassersteinScorer with a 24-timestamp window, and WassersteinScorer with a 48-timestamp window for NYC Taxi, while a slightly different configuration was used for the Ambient Temperature dataset. Starting with the NYC Taxi dataset, the results show that bigger window sizes make the anomaly detection performance better, particularly for more complex models. The Regression model shows a moderate initial performance but improves when leveraging extended temporal contexts. Both RF and GB models (CatBoost and XGBoost) demonstrate strong adaptability, with performance peaking when using Wasserstein scoring with larger window sizes.

In contrast, the Ambient Temperature dataset presents different model behaviors, with RF and CatBoost achieving consistently high performance across all scoring methods. While the Regression model benefits from longer window sizes, its improvement is less pronounced compared to other datasets. XGBoost, on the other hand, exhibits robust performance, particularly under Wasserstein-based scoring, effectively capturing long-term dependencies in temperature variations.

Table 5. Performance comparison of models using different metrics for NYC taxi and ambient temperature datasets.

Model	Scorer	Dataset			
		NYC Taxi		Ambient temperature	
		AUC_ROC Metric	AUC_PR Metric	AUC_ROC Metric	AUC_PR Metric
Regression Model	Norm (ord=1)_w=1	0.658	0.215	0.571	0.010
	WassersteinScorer_w=24	0.884	0.609	0.567	0.015
	WassersteinScorer_w=48	0.950	0.687	0.746	0.030
RF Model	Norm (ord=1)_w=1	0.757	0.391	0.999	0.888
	WassersteinScorer_w=24	0.887	0.611	0.998	0.900
	WassersteinScorer_w=48	0.895	0.726	0.992	0.661
Cat Boost Model	Norm (ord=1)_w=1	0.768	0.403	0.999	0.906
	WassersteinScorer_w=24	0.904	0.648	0.999	0.926
	WassersteinScorer_w=48	0.918	0.778	0.993	0.683
XG Boost Model	Norm (ord=1)_w=1	0.742	0.328	0.998	0.860
	WassersteinScorer_w=24	0.911	0.653	0.998	0.868
	WassersteinScorer_w=48	0.926	0.788	0.992	0.684

In Table 6 below for the CPU Utilization dataset, models were assessed using AUC-ROC and AUC-PR metrics under three different scoring methodologies: Norm (order=1) with a unit window, WassersteinScorer with a 3-timestamp window, and WassersteinScorer with a 6-timestamp window. The regression model initially shows moderate performance under the norm scoring method but exhibits improvements when Wasserstein scoring is incorporated. This suggests that even short-term temporal dependencies contribute to better anomaly detection. RF and CatBoost kept straight performance across the different scoring methods, with notable improvements under Wasserstein-based scoring. XGBoost shows the most robust anomaly detection capability, particularly under Wasserstein-based scoring. This highlights its ability to capture short-term anomalies in CPU utilization. The results emphasize that leveraging short-term window sizes in Wasserstein scoring improves detection performance across all models, particularly for GB methods.

Table 6. Performance comparison of models using different metrics for CPU utilization datasets.

Model	Scorer	Dataset	
		CPU utilization	
		AUC_ROC Metric	AUC_PR Metric
Regression Model	Norm (ord=1)_w=1	0.917	0.008
	WassersteinScorer_w=3	0.977	0.199
	WassersteinScorer_w=6	0.959	0.124
RF Model	Norm (ord=1)_w=1	0.944	0.001
	WassersteinScorer_w=3	0.953	0.221
	WassersteinScorer_w=6	0.970	0.767
Cat Boost Model	Norm (ord=1)_w=1	0.881	0.004
	WassersteinScorer_w=3	0.951	0.167
	WassersteinScorer_w=6	0.943	0.113
XG Boost Model	Norm (ord=1)_w=1	0.982	0.011
	WassersteinScorer_w=3	0.971	0.350
	WassersteinScorer_w=6	0.960	0.241

In Table 7, the models were assessed on the Yahoo dataset, which is known for its synthetic anomalies, using AUC-ROC and AUC-PR metrics across three scoring approaches: Norm (order=1) and WassersteinScorer with 10 and 12-timestamp windows. Interestingly, the Regression model outperforms all other models on this dataset, achieving perfect scores under the Norm scorer and maintaining high performance even with Wasserstein-based scoring. This behavior may come from the simplified, synthetic structure of the dataset, which allows linear models to act well without using complex feature extraction. In contrast, RF and XGBoost show a drop in performance when moving from Norm to Wasserstein scoring, showing reduced effectiveness in detecting anomalies across longer temporal contexts in this dataset. The CatBoost model is still capable of doing well but not as well as the Regression model in Norm-based scoring. These findings suggest that, for synthetic datasets like Yahoo S5 A2, using simple models could be enough, and complex GB models may not provide additional benefits, especially when the anomaly patterns are straightforward and well-defined.

Table 7. Performance comparison of models using different metrics for Yahoo S5 A2 datasets.

Model	Scorer	Dataset	
		Yahoo S5 A2	
		AUC_ROC Metric	AUC_PR Metric
Regression Model	Norm (ord=1)_w=1	1	1
	WassersteinScorer_w=10	0.973	0.785
	WassersteinScorer_w=12	0.981	0.857
RF Model	Norm (ord=1)_w=1	0.998	0.743
	WassersteinScorer_w=10	0.642	0.547
	WassersteinScorer_w=12	0.591	0.413
Cat Boost Model	Norm (ord=1)_w=1	0.998	0.999
	WassersteinScorer_w=10	0.880	0.814
	WassersteinScorer_w=12	0.891	0.753
XGBoost Model	Norm (ord=1)_w=1	0.998	0.886
	WassersteinScorer_w=10	0.631	0.540
	WassersteinScorer_w=12	0.571	0.401

The analysis establishes the enhanced performance of GB methodologies, which is particularly evident with expanded window sizes. The WassersteinScorer implementation with a long-term window demonstrates consistent superior performance across all model configurations, highlighting the significance of comprehensive temporal context in TS anomaly detection. The approach uses the Darts QuantileDetector to transform continuous anomaly scores into binary classifications via a systematic evaluation process. This framework uses a high quantile threshold of 0.95, computed from the historical data distributions, to allow focused identification of significant deviations in the temporal patterns. The quantile-based classification methodology designates the upper 5th percentile of scores as anomalous observations, establishing an optimal balance between detection sensitivity and specificity. The analytical foundation comprises absolute residuals from forecasting models, with the long-term timestamp window WassersteinScorer configuration exhibiting superior performance through elevated AUC-ROC metrics.

4.3. ANOMALY DETECTION VISUALIZATION

In this section, we compare the detected anomalies with the actual static anomalies for each implemented model dataset, one by one. Experimental results confirm the effectiveness of the detection methodology in identifying temporal anomalies while maintaining low false positive rates.

Visual Figure 16 illustrates the anomaly detection performance of the regression model on the NAB NYC taxi dataset. Anomalies were identified based on large residuals between the model's predicted and actual passenger counts. The peaks of the predicted results often catch the labeled anomalies, such as extreme fluctuations in demand, indicating the model's ability to capture significant events. However, some minor anomalies may be missed or misrepresented due to the limitations of linear regression in capturing complex temporal dynamics.

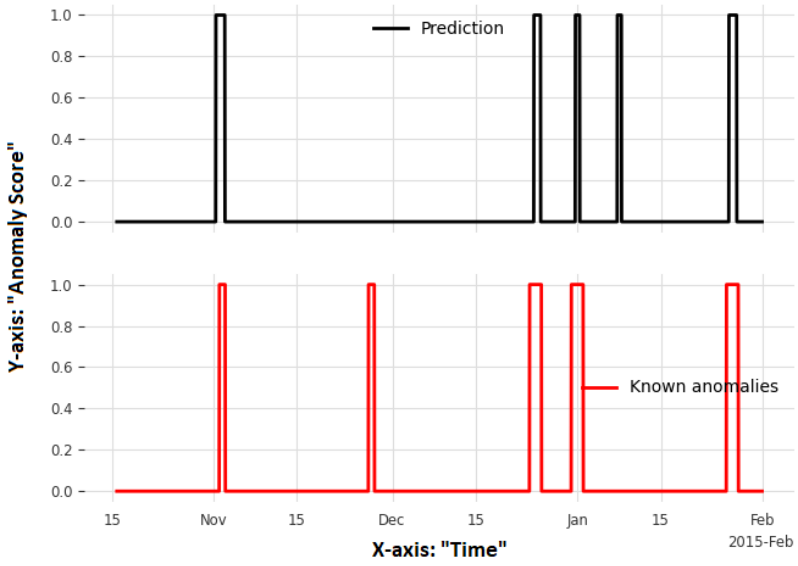

Figure 16. Detected anomalies vs. ground truth anomalies of regression model for NAB NYC Taxi dataset.

In the following Figure 17 shows the performance of the anomaly detection performance of the RF model on the same dataset as above; the model predicts the major anomalies and does exhibit a few false positives, flagging events that are not

labeled as anomalies in the ground truth. Overall, the results confirm RF's effectiveness in identifying temporal patterns and abnormal events.

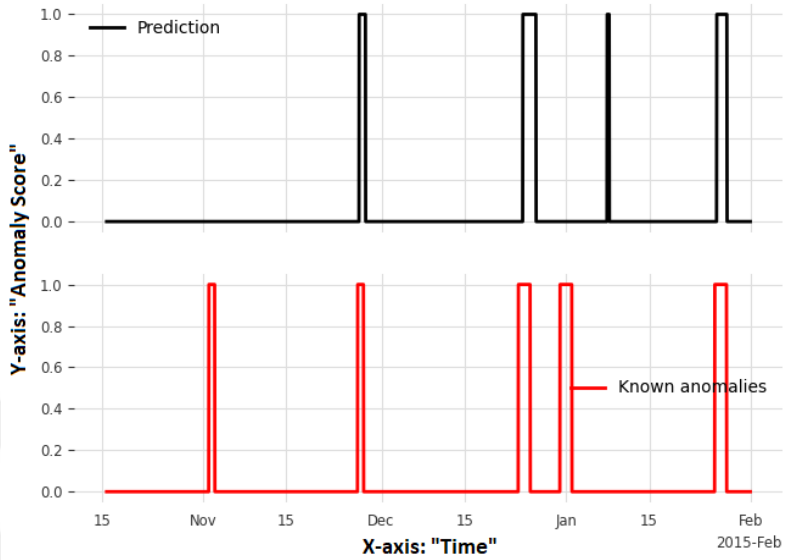

Figure 17. Detected anomalies vs. ground truth anomalies of RF model for NAB NYC Taxi dataset.

Figure 18 presents the CatBoost model's anomaly detection performance on the NAB NYC taxi dataset; also, this model shows a good alignment with ground truth anomalies during major events. While maintaining a low false positive rate, indicating a balanced sensitivity and specificity. The sharp spike patterns suggest that the model distinguishes between normal and abnormal behavior.

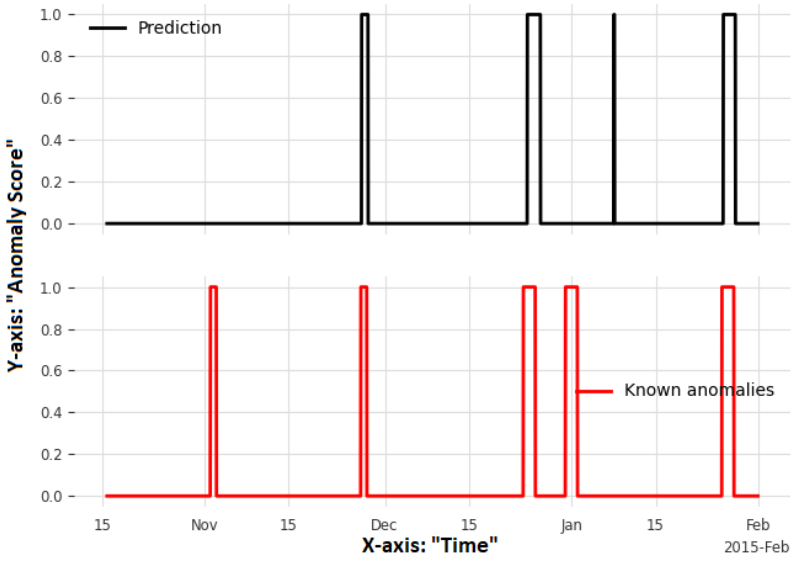

Figure 18. Detected anomalies vs. ground truth anomalies of CatBoost model for NAB NYC Taxi dataset.

Figure 19 visualizes the anomaly detection results of the XGBoost model applied to the NAB NYC taxi dataset. The model performed well and exhibited precise anomaly flagging during major deviations around the holiday season. Compared to other models, the XGBoost anomaly detection model was better.

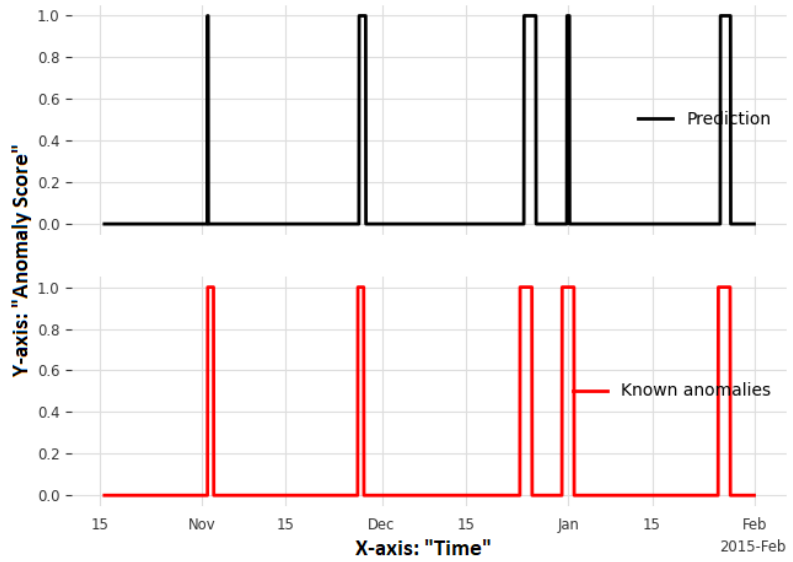

Figure 19. Detected anomalies vs. ground truth anomalies of XGBoost model for NAB NYC Taxi dataset.

Figure 20 displays the anomaly detection performance of the regression model on the NAB CPU utilization dataset. The model captured the main labeled anomalies. It produces several consecutive anomaly marks, which might suggest sensitivity to short-term fluctuations, potentially leading to an over-mark around actual anomaly periods.

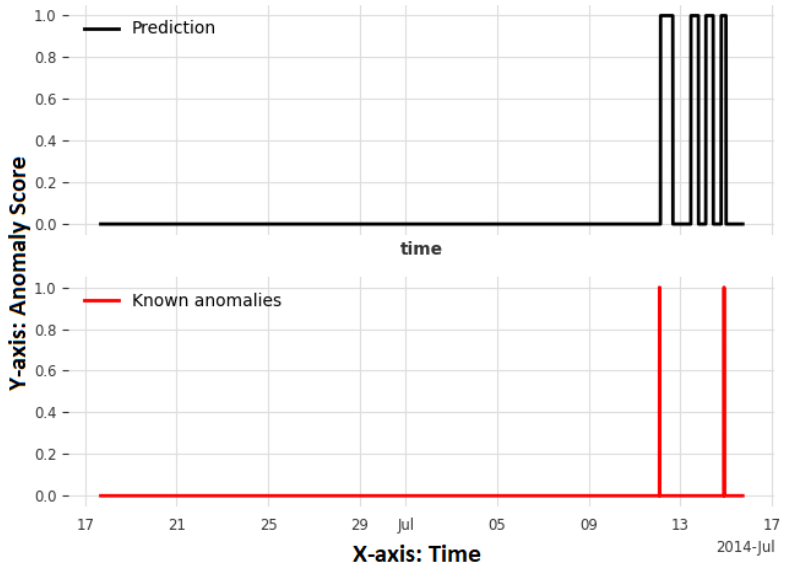

Figure 20. Detected anomalies vs. ground truth anomalies of regression model for NAB CPU utilization dataset.

In Figure 21, the RF anomaly detection model on the CPU utilization dataset detected a single spike in anomaly score. Compared to the Regression model, RF produces fewer false positives.

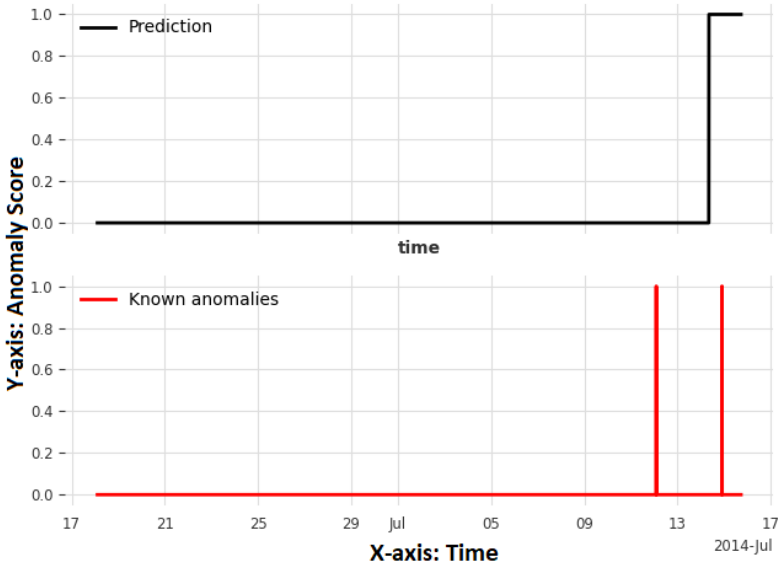

Figure 21. Detected anomalies vs. ground truth anomalies of RF model for NAB CPU utilization dataset.

Figure 22 presents the anomaly detection results using the CatBoost model on the NAB CPU utilization dataset. The model identified the key anomalous intervals, which means it captures deviations in system resource usage over time.

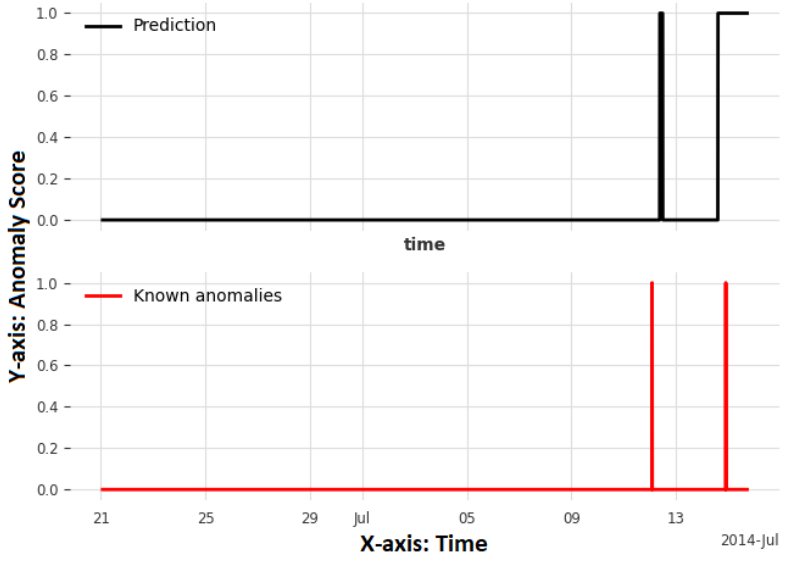

Figure 22. Detected anomalies vs. ground truth anomalies of CatBoost model for NAB CPU utilization dataset.

Figure 23 illustrates the XGBoost anomaly detection model performance applied to the NAB CPU utilization dataset. The prediction spikes start occurring just before or directly on the dates marked as true anomalies. This behavior suggests that XGBoost can capture abrupt shifts in CPU performance and mark them as deviations from expected patterns.

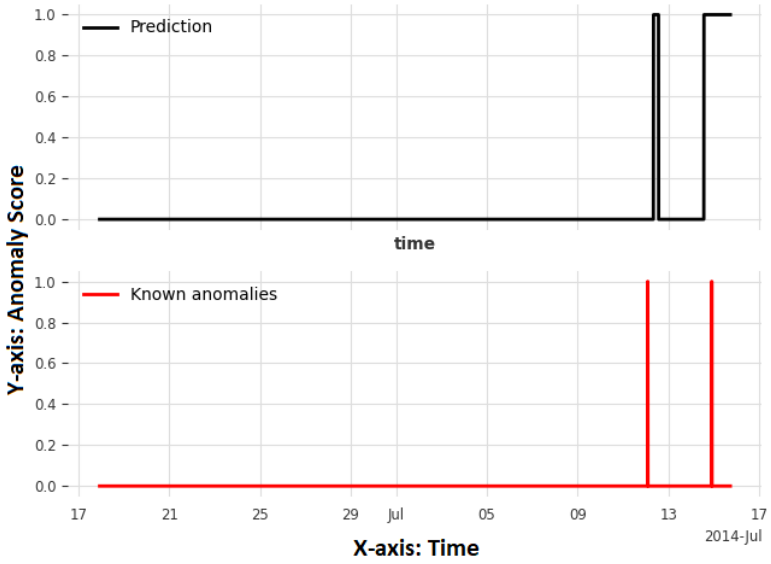

Figure 23. Detected anomalies vs. ground truth anomalies of XGBoost model for NAB CPU utilization dataset.

For the Ambient Temperature System Failure dataset, it appears that the models identified many of the true anomalies and missed some in some models due to the structure of this dataset.

Figure 24 shows the anomaly detection performance of the Regression model for the dataset mentioned above. The model flags deviations from the normal pattern but wasn’t perfect due to the dataset structure.

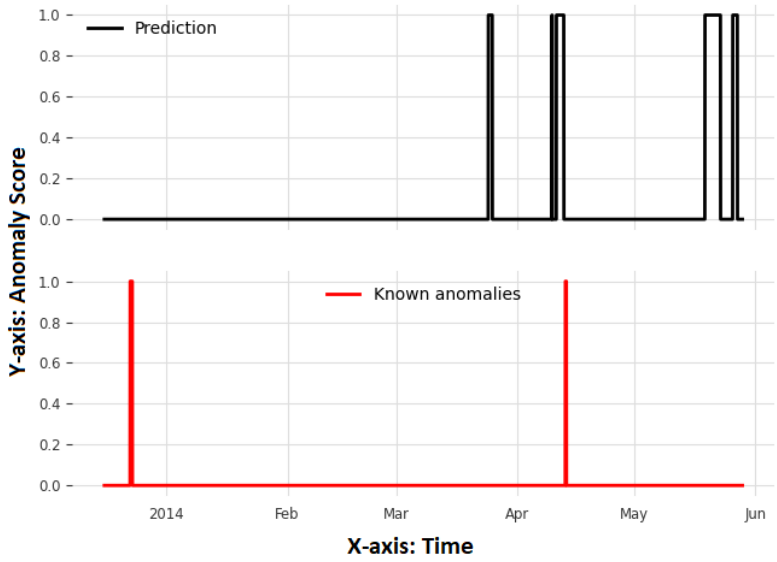

Figure 24. Detected anomalies vs. ground truth anomalies of regression model for NAB Ambient Temperature dataset.

Figure 25 presents the RF anomaly detection model that detects key deviations from expected temperature behavior.

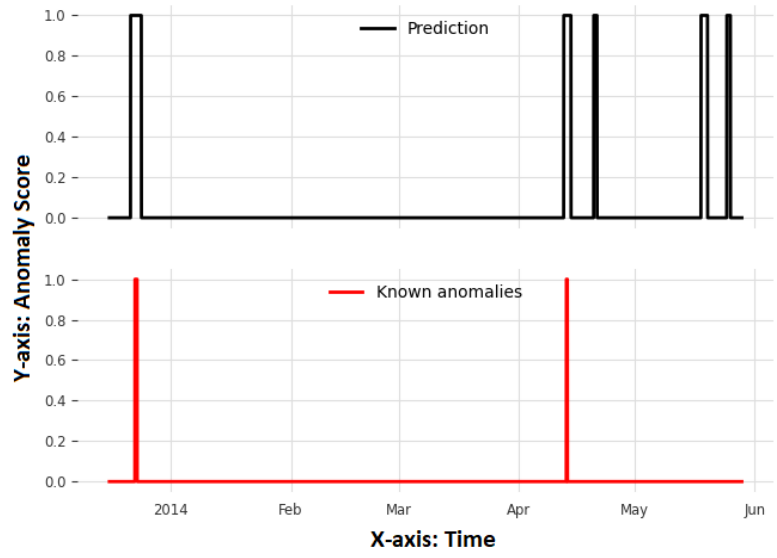

Figure 25. Detected anomalies vs. ground truth anomalies of RF model for NAB Ambient Temperature dataset.

Figure 26 shows the anomaly detection CatBoost model performance on the same dataset. The model identifies key anomalies in early January and April, demonstrating

strong alignment with the ground truth, better than the regression model and almost similar to RF and XGBoost, with a few isolated false positives observed, suggesting minor over-sensitivity in certain regions.

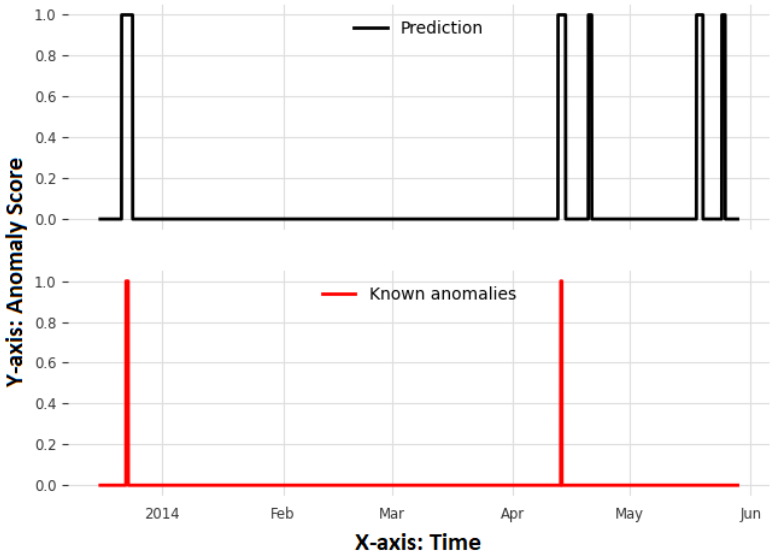

Figure 26. Detected anomalies vs. ground truth anomalies CatBoost model for NAB Ambient Temperature dataset.

Figure 27 displays the XGBoost anomaly detection model performance, which captures the major known anomalies in January and April, aligning well with the ground truth. Additionally, a few early and late spikes are observed in the prediction, which may represent potential false positives.

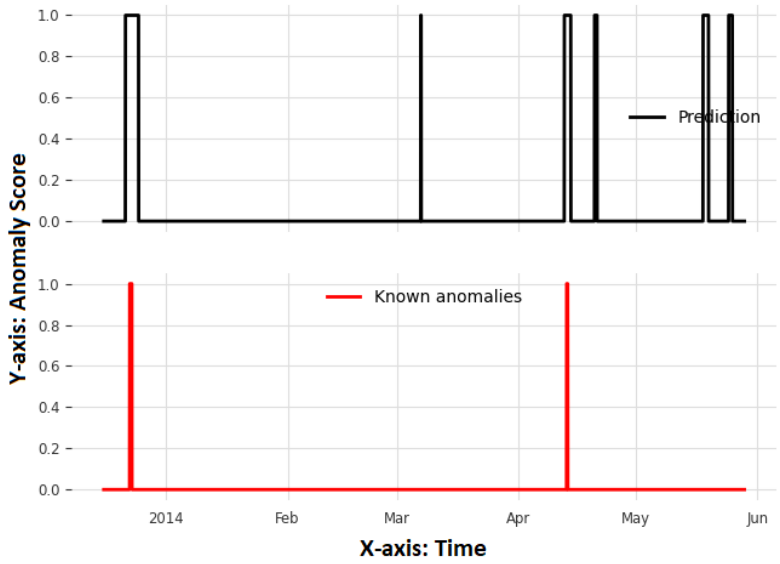

Figure 27. Detected anomalies vs. ground truth anomalies XGBoost model for NAB Ambient Temperature dataset.

For the Yahoo S5 dataset, all the models successfully performed well because of the simple periodic dataset used.

The regression model anomaly detection in Figure 28 detects most of the anomalies, particularly those occurring, closely aligning with the ground truth.

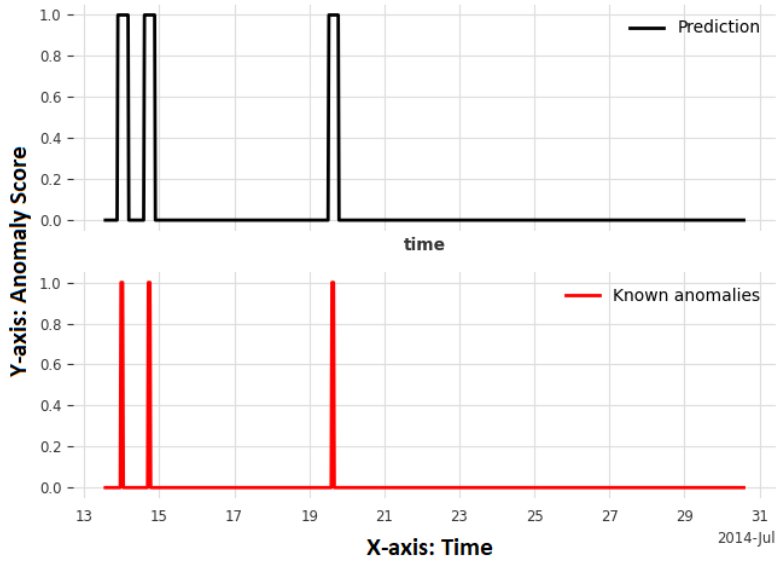

Figure 28. Detected anomalies vs. ground truth anomalies regression model for Yahoo S5 A2 dataset.

The performance of the RF model, as shown in Figure 29, successfully flags the major anomalies. It displays high sensitivity with early detections and minimal false positives.

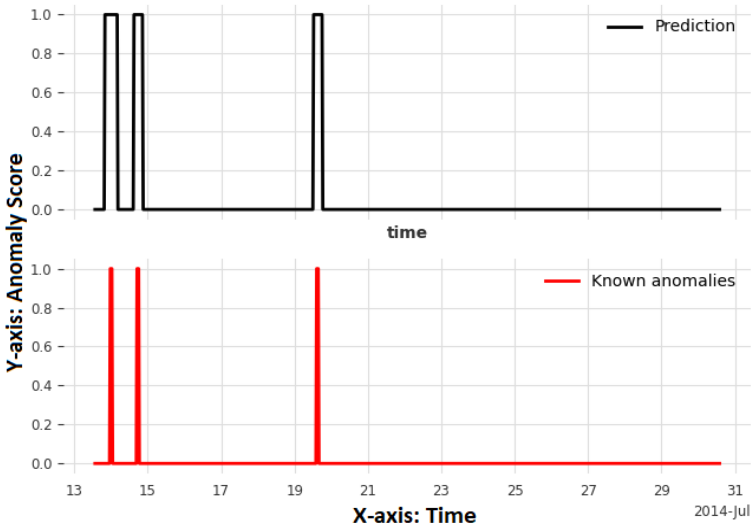

Figure 29. Detected anomalies vs. ground truth anomalies RF model for Yahoo S5 A2 dataset.

Figure 30 illustrates the anomaly detection CatBoost model performance, which accurately identifies the primary anomalies with minimal noise, similar to the other models.

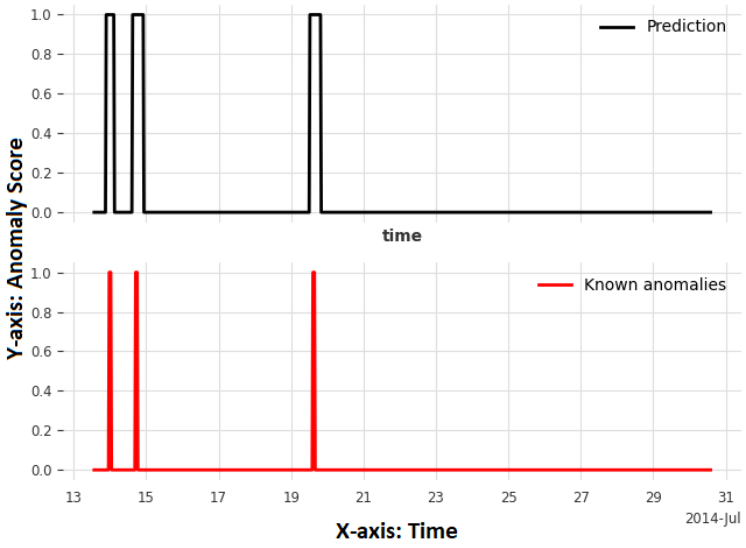

Figure 30. Detected anomalies vs. ground truth anomalies CatBoost model for Yahoo S5 A2 dataset.

Similarly to the other models for this dataset in Figure 31 below, the XGBoost model identified the anomalies, suggesting that XGBoost effectively captures sharp deviations and temporal anomalies within the TS.

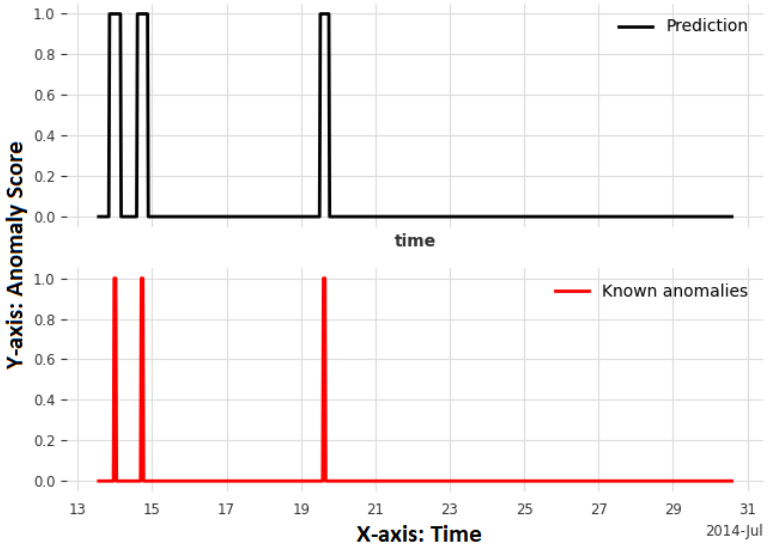

Figure 31. Detected anomalies vs. ground truth anomalies XGBoost model for Yahoo S5 A2 dataset.

4.4. ACCURACY METRIC

Table 8 presents the accuracy metrics for the four models implemented in this work: Regression, RF, CatBoost, and XGBoost across distinct datasets NYC Taxi, CPU Utilization (ASG Misconfiguration), Ambient Temperature (System Failure), and Yahoo S5 A2. The experimental results have shown very high detection accuracy across all implementations. The Regression model gives a baseline of 91% accuracy for the NYC Taxi dataset, 95% for the CPU Utilization dataset, 93% for the Ambient Temperature dataset, and 97% for the Yahoo S5 A2. Meanwhile, the RF implementation exhibits marginally enhanced performance at 92%, 92%, 97%, and 96% accuracy, respectively, across four datasets.

The GB implementations, CatBoost and XGBoost, both achieve superior and stable accuracy across datasets, with CatBoost achieving 93%, 96%, 97%, and 97% for the three datasets. The XGBoost model achieved 93%, 97%, 96%, and 96%. In addition,

no single model consistently outperforms others across all scenarios. The effectiveness of each model appears to depend heavily on the characteristics of the dataset. Also, the performance differences look small. However, the differences can have significant implications in practical applications, particularly in scenarios requiring low false positive rates. The consistently high accuracy rates across all applications demonstrate the robustness of the Darts framework for TS anomaly detection applications.

Table 8. Performance of accuracy metric for different datasets.

	Datasets			
Models	**NAB (NYC taxi)**	**NAB (Cpu utilization ASG misconfiguration)**	**NAB (Ambient temperature system failure)**	**Yahoo S5 A2**
Regression Model (linear)	0.91	0.95	0.93	0.97
RF Model	0.92	0.92	0.97	0.96
CatBoost Model	0.93	0.96	0.97	0.97
XGBoost Model	0.93	0.97	0.96	0.96

These observations underscore that model selection for anomaly detection should be guided by the nature of the dataset rather than assuming the universal superiority of a specific algorithm. The results demonstrate that even traditional methods like regression can achieve competitive performance when properly implemented within the Darts framework.

For a better understanding of the proposal model in the area of anomaly detection approaches, we compare it with the related work; the table presents a comparative overview of several key studies. The comparison considers factors such as model complexity, computational cost, ability to handle varying dataset sizes, and domain applicability. As the comparison table 9 illustrates, some models gain high accuracy in specific areas, such as railway monitoring or IoT security. These models often require extra computational requirements or limited scalability. On the other hand, the

proposed approach features low complexity and computational efficiency, offering broader applicability across multiple domains. This highlights its potential as a practical and adaptable solution for TS anomaly detection technology.

Table 9. Comparative overview of related work.

Model	Year Of Study	Complicity	Computational Cost	Handling small and Large Datasets	Domain Applicability
Anomaly Detection in Railway Sensor Data Environments	2024	High	High	✓	Railway Monitoring
ML Methods for the Prediction of Wastewater Treatment Efficiency and Anomaly Classification	2024	Low	Low	✗	Environmental Engineering
An Improved Anomaly Detection Model for IoT Security Based on Gradient Boosting and Decision Tree Algorithms	2023	Low	Medium	✓	IoT Security
Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art	2020	Low	Medium	✗	Multiple Domains
ML Methods for Anomaly Detection in Industrial Control Systems	2020	Low	Low	✗	Industrial Control Systems
Our Proposed Work	2025	Low	Low	✓	Multiple Domains

PART 5

CONCLUSION

This research presents a comprehensive study of ML methodologies for univariate TS anomaly detection on four datasets: three from the NAB (NYC Taxi, CPU Utilization, Ambient Temperature dataset) and one from the Yahoo S5 TS anomaly detection dataset. The background of TS data and the role of ML in anomaly detection are introduced. The second chapter provides more details about the concept of TS anomaly detection and its challenges, as well as more details about ML. A comparative review of relevant related works is also included. The third chapter in this work outlines the model architecture and provides a detailed step-by-step explanation of the experimental design, including the selected models, dataset descriptions, hyperparameter configurations, and evaluation methodologies. The final chapter discusses results supported by metric evaluations and visualizations.

The experimental findings demonstrate that no single model performs best across all datasets, emphasizing the importance of data characteristics in influencing detection accuracy. Notably, GB methodologies, XGBoost and CatBoost, achieved a high accuracy rate with structured or gradually evolving anomalies, while the regression model proved effective for synthetic or simple datasets with sharp anomaly signals. These empirical results establish the enhanced capabilities of contemporary ML algorithms in TS anomaly detection applications. The performance analysis through AUC-ROC and AUC-PR metrics under the WassersteinScorer with long-term window configuration yielded comprehensive insights. XGBoost demonstrated superior performance across all datasets, closely followed by CatBoost. The RF showed competitive performance, maintaining consistent and robust detection capabilities, while the Regression model showed significant improvement with extended window size. These results validate the effectiveness of incorporating broader temporal context through the WassersteinScorer methodology.

PART 6

FUTURE WORK

Future research directions will include extending the methodology presented to multivariate TS analysis and applying it to multiple data streams at the same time. This would allow a more comprehensive anomaly detection by considering correlations between different variables in complex systems. Another way of promising future direction is the development of hybrid methodologies that incorporate domain-specific knowledge bases with ML algorithms. Furthermore, continued experimental validation of the proposed approach across additional data sets and application domains will help to establish the generalizability of the method. Exploration of enriched scoring methods and temporal context integration methods also offers potential avenues for optimization in TS anomaly detection systems. Finally, incorporating explainability frameworks into detection models will be critical for adoption in sensitive or regulated industries such as finance and healthcare.

REFERENCES

1. Chandola, V., Banerjee, A., and Kumar, V., "Anomaly detection: A survey", ***ACM Computing Surveys (CSUR)***, 41 (3): 1–58 (2009).

2. Karaoğlan, K. M., Fndk, O., and Başaran, E., "Anomaly Detection in Meteorological Data Using a Hierarchical Temporal Memory Model: A Study on the Case of Kazakhstan", ***Frat Üniversitesi Mühendislik Bilimleri Dergisi***, 36 (1): 481–498 (2024).

3. Karaoğlan, K. M., "Zaman Serilerinde Anomali Tespiti Üzerine Genel Bir Bakış: Kavramlar, Teknikler, Güncel Yaklaşımlar, Zorluklar ve Fırsatlar", Disiplinlerarası Yapay Zekâ Araştırmaları, 188 (2023).

4. Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., and Salehi, M., "Deep Learning for Time Series Anomaly Detection: A Survey", ***ACM Computing Surveys***, 57: 42 (2024).

5. Braei, M. and Wagner, Dr.-I. S., "Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art", (2020).

6. Kim, T. Y. and Cho, S. B., "Web traffic anomaly detection using C-LSTM neural networks", ***Expert Systems With Applications***, 106: 66–76 (2018).

7. Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Targio Hashem, I. A., Ahmed, E., and Imran, M., "Real-time big data processing for anomaly detection: A Survey", ***International Journal Of Information Management***, 45: 289–307 (2019).

8. Karaoğlan, K. and Saka, F., "Detecting Anomalies in Dam Water Levels using Hierarchical Temporal Memory: A Case Study in Istanbul Province", (2023).

9. Hilal, W., Gadsden, S. A., and Yawney, J., "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances", ***Expert Systems With Applications***, 193: 116429 (2022).

10. Sgueglia, A., Di Sorbo, A., Visaggio, C. A., and Canfora, G., "A systematic literature review of IoT time series anomaly detection solutions", ***Future Generation Computer Systems***, 134: 170–186 (2022).

11. Min, M., Lee, J. J., Park, H., and Lee, K., "Detecting anomalous transactions via an iot based application: A machine learning approach for horse racing betting", ***Sensors***, 21 (6): 2039 (2021).

12. Choi, K., Yi, J., Park, C., and Yoon, S., "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines", ***IEEE Access***, 9: 120043–120065 (2021).

13. Terrades, O. R., Berenguel, A., and Gil, D., "A flexible outlier detector based on a topology given by graph communities", ***Big Data Research***, 29: 100332 (2022).

14. Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M., "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019", ***Applied Soft Computing***, 90: 106181 (2020).

15. Nassif, A. B., Talib, M. A., Nasir, Q., and Dakalbab, F. M., "Machine Learning for Anomaly Detection: A Systematic Review", ***IEEE Access***, 9: 78658–78700 (2021).

16. Panda, S. K. and Mohanty, S. N., "Time series forecasting and modeling of food demand supply chain based on regressors analysis", ***IEEE Access***, 11: 42679–42700 (2023).

17. Velasquez, D., Perez, E., Oregui, X., Artetxe, A., Manteca, J., Mansilla, J. E., Toro, M., Maiza, M., and Sierra, B., "A Hybrid Machine-Learning Ensemble for Anomaly Detection in Real-Time Industry 4.0 Systems", ***IEEE Access***, 10: 72024–72036 (2022).

18. Wu, R. and Keogh, E. J., "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress", ***IEEE Transactions On Knowledge And Data Engineering***, 35 (3): 2421–2429 (2023).

19. Majidpour, M., Nazaripouya, H., Chu, P., Pota, H. R., and Gadh, R., "Fast Univariate Time Series Prediction of Solar Power for Real-Time Control of Energy Storage System", ***Forecasting 2019, Vol. 1, Pages 107-120***, 1 (1): 107–120 (2018).

20. Chayama, M. and Hirata, Y., "When univariate model-free time series prediction is better than multivariate", ***Physics Letters A***, 380 (31–32): 2359–2365 (2016).

21. Huertas-García, Á., Martí-González, C., Maezo, R. G., and Rey, A. E., "A Comparative Study of Machine Learning Algorithms for Anomaly Detection in Industrial Environments: Performance and Environmental Impact", ***Trends In Sustainable Computing And Machine Intelligence (ICTSM 2023)***, 373–389 (2024).

22. Shaukat, K., Alam, T. M., Luo, S., Shabbir, S., Hameed, I. A., Li, J., Abbas, S. K., and Javed, U., "A Review of Time-Series Anomaly Detection Techniques: A Step to Future Perspectives", ***Advances In Intelligent Systems And Computing***, 1363 AISC: 865–877 (2021).

23. Garza, A., Challu, C., and Mergenthaler-Canseco, M., "TimeGPT-1", ***ArXiv Preprint ArXiv:2310.03589***, (2023).

24. Usmani, U. A., Aziz, I. A., Jaafar, J., and Watada, J., "Deep Learning for Anomaly Detection in Time-Series Data: An Analysis of Techniques, Review of Applications, and Guidelines for Future Research", ***IEEE Access***, (2024).

25. Tuli, S., Casale, G., and Jennings, N. R., "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data", ***Proceedings Of The VLDB Endowment***, (2022).

26. Liu, J., Yang, D., Zhang, K., Gao, H., and Li, J., "Anomaly and change point detection for time series with concept drift", ***World Wide Web***, 26 (5): 3229–3252 (2023).

27. Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., and Steger, C., "The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection", ***International Journal Of Computer Vision***, 129 (4): 1038–1059 (2021).

28. Sen, P. C., Hajra, M., and Ghosh, M., "Supervised Classification Algorithms in Machine Learning: A Survey and Review", ***Advances In Intelligent Systems And Computing***, 937: 99–111 (2020).

29. Toshniwal, A., Mahesh, K., and Jayashree, R., "Overview of anomaly detection techniques in machine learning", ***Proceedings Of The 4th International Conference On IoT In Social, Mobile, Analytics And Cloud, ISMAC 2020***, 808–815 (2020).

30. Ghosal, A., Nandy, A., Das, A. K., Goswami, S., and Panday, M., "A Short Review on Different Clustering Techniques and Their Applications", ***Advances In Intelligent Systems And Computing***, 937: 69–83 (2020).

31. Jiang, J.-R., Kao, J.-B., Li, Y.-L., Jiang, C. :, Kao, J.-R. ;, and Li, J.-B. ;, "Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning", ***Applied Sciences 2021, Vol. 11, Page 6698***, 11 (15): 6698 (2021).

32. Al-Amri, R., Murugesan, R. K., Man, M., Abdulateef, A. F., Al-Sharafi, M. A., and Alkahtani, A. A., "A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data", ***Applied Sciences 2021, Vol. 11, Page 5320***, 11 (12): 5320 (2021).

33. Zhang, Q., "Financial Data Anomaly Detection Method Based on Decision Tree and Random Forest Algorithm", ***Journal Of Mathematics***, 2022 (1): 9135117 (2022).

34. Yilmaz, M. N. and Bardak, B., "An Explainable Anomaly Detection Benchmark of Gradient Boosting Algorithms for Network Intrusion Detection Systems",

Proceedings - 2022 Innovations In Intelligent Systems And Applications Conference, ASYU 2022, (2022).

35. Demir, H. and Karaoğlan, K., "A Comparative Performance Analysis of LSTM Autoencoder and TimeGPT Models in Time Series Anomaly Detection", (2024).

36. Lindemann, B., Maschler, B., Sahlab, N., and Weyrich, M., "A survey on anomaly detection for technical systems using LSTM networks", ***Computers In Industry***, 131: 103498 (2021).

37. Abdulrazzaq, M. M., Ramaha, N. T. A., Hameed, A. A., Salman, M., Yon, D. K., Fitriyani, N. L., Syafrudin, M., and Lee, S. W., "Consequential Advancements of Self-Supervised Learning (SSL) in Deep Learning Contexts", ***Mathematics 2024, Vol. 12, Page 758***, 12 (5): 758 (2024).

38. Bałdyga, M., Barański, K., Belter, J., Kalinowski, M., and Weichbroth, P., "Anomaly Detection in Railway Sensor Data Environments: State-of-the-Art Methods and Empirical Performance Evaluation", ***Sensors 2024, Vol. 24, Page 2633***, 24 (8): 2633 (2024).

39. Gulshin, I. and Kuzina, O., "Machine Learning Methods for the Prediction of Wastewater Treatment Efficiency and Anomaly Classification with Lack of Historical Data", ***Applied Sciences 2024, Vol. 14, Page 10689***, 14 (22): 10689 (2024).

40. Douiba, M., Benkirane, S., Guezzaz, A., and Azrour, M., "An improved anomaly detection model for IoT security using decision tree and gradient boosting", ***Journal Of Supercomputing***, 79 (3): 3392–3411 (2023).

41. Tai, J., Alsmadi, I., Zhang, Y., and Qiao, F., "Machine Learning Methods for Anomaly Detection in Industrial Control Systems", ***Proceedings - 2020 IEEE International Conference On Big Data, Big Data 2020***, 2333–2339 (2020).

42. Oswal, S., Hadawle, S., and Khangar, A., "Anomaly Detection in Time Series Data by Forecasting Using Facebook Prophet", ***Communications In Computer And Information Science***, 1782 CCIS: 205–220 (2023).

43. Nakashima, T. and Yairi, T., "Assessing the Performance of Transformer for Time Series Anomaly Detection", ***PHM Society Asia-Pacific Conference***, 4 (1): (2023).

44. Alkhoja, M. and Karaoğlan, K. M., "Performance Evaluation of Machine Learning Models for Time Series Anomaly Detection using Darts Framework", ***Anadolu 16th International Congress of Applied Sciences*** (2024).

45. Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. Van, Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M.,

Petrik, M., and Grosch, G., "Darts: User-Friendly Modern Machine Learning for Time Series", ***Journal Of Machine Learning Research***, 23 (124): 1–6 (2022).

46. Sperl, R. E. and Chung, S. M., "Two-Step Anomaly Detection for Time Series Data", ***Proceedings Of 2019 International Conference On Data And Software Engineering, ICoDSE 2019***, (2019).

47. Zhang, Y., Chen, Y., Wang, J., and Pan, Z., "Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals", ***IEEE Transactions On Knowledge And Data Engineering***, 35 (2): 2118–2132 (2023).

48. Beskopylny, A. N., Stel'makh, S. A., Shcherban', E. M., Mailyan, L. R., Meskhi, B., Razveeva, I., Chernil'nik, A., and Beskopylny, N., "Concrete Strength Prediction Using Machine Learning Methods CatBoost, k-Nearest Neighbors, Support Vector Regression", ***Applied Sciences (Switzerland)***, 12 (21): 10864 (2022).

49. Liu, Y., Liu, L., Yang, L., Hao, L., and Bao, Y., "Measuring distance using ultra-wideband radio technology enhanced by extreme gradient boosting decision tree (XGBoost)", ***Automation In Construction***, 126: 103678 (2021).

50. Yilmaz, M. N. and Bardak, B., "An Explainable Anomaly Detection Benchmark of Gradient Boosting Algorithms for Network Intrusion Detection Systems", ***Proceedings - 2022 Innovations In Intelligent Systems And Applications Conference, ASYU 2022***, (2022).

51. Douiba, M., Benkirane, S., Guezzaz, A., and Azrour, M., "Anomaly detection model based on gradient boosting and decision tree for IoT environments security", ***Journal Of Reliable Intelligent Environments***, 9 (4): 421–432 (2023).

52. Marzak, Z., Benabbou, R., Mouatassim, S., and Benhra, J., "Forecasting Multivariate Time Series with Trend and Seasonality: A Random Forest Approach", ***Communications In Computer And Information Science***, 2373 CCIS: 128–144 (2025).

53. Lavin, A. and Ahmad, S., "Evaluating Real-time Anomaly Detection Algorithms - the Numenta Anomaly Benchmark", ***Proceedings - 2015 IEEE 14th International Conference On Machine Learning And Applications, ICMLA 2015***, 38–44 (2015).

54. Thill, M., Konen, W., and Back, T., "Online anomaly detection on the webscope S5 dataset: A comparative study", ***IEEE Conference On Evolving And Adaptive Intelligent Systems***, 2017-May: (2017).

55. Hasani, Z., "Robust anomaly detection algorithms for real-time big data: Comparison of algorithms", ***2017 6th Mediterranean Conference On***

Embedded Computing, MECO 2017 - Including ECYPS 2017, Proceedings, (2017).

56. Ru, V. I., Bernstein, A., Ru, A. B., Ru, E. B., Nazarov, I., Gammerman, A., Vovk, V., Luo, Z., and Papadopoulos, H., "Conformal k-NN Anomaly Detector for Univariate Data Streams", ***Proceedings Of Machine Learning Research***, 60: 1–15 (2017).

RESUME

Mohamad Alkhoja is a computer engineer with a Bachelor's degree in Computer Engineering (2016–2021) from Karabük University, Türkiye, where he specialized in web development, data science, artificial intelligence, and Android application development. His undergraduate thesis focused on the Backyard Alarm System Stand On Animal Detection. Currently, he is pursuing a Master's degree in Computer Engineering at Karabük University, with a strong foundation in programming principles across multiple platforms. Fluent in Arabic, English, and Turkish, he is motivated to learn. His academic and professional interests include artificial intelligence and data science.