Eötvös Loránd University

Faculty of Informatics

Dept. of Data Science

Time Series Anomaly Detection in Industrial Control Systems

Supervisor:
Mahmud Jiyan Salim
Data Science researcher

Author:
Oumaima DHAIDAH
Data Science MSc

Budapest, 2023

Contents

abstract – Anomaly detection plays a crucial role in various domains, including network security, financial fraud detection, and industrial process monitoring. In this paper, we propose a novel approach for anomaly detection in time series data using V Autoencoders. We compare our method with the baseline Autoencoder model and evaluate their performance on a real-world dataset. Our experimental results show that the V Autoencoder model achieves a higher accuracy of 93.51% compared to the baseline Autoencoder's accuracy of 92.62%. Additionally, we analyze precision, recall, and F1-Score as additional evaluation metrics. Our V Autoencoder model achieves a precision of 27.81%, recall of 4.60%, and F1-Score of 7.90%. These results indicate the effectiveness of our method in identifying anomalies, although with relatively lower recall and F1-Score compared to precision. Our study contributes to the field of anomaly detection by introducing a novel approach utilizing V Autoencoders. Further research can focus on refining the architecture, exploring threshold selection techniques, and improving recall and F1-Score to enhance the performance of anomaly detection systems.

Chapter 1

Introduction

Industrial control systems (ICS) are widely used in various industries to monitor and control critical processes. However, these systems are often targeted by cyber attackers who seek to disrupt or damage the industrial process. Anomaly detection is an effective way to detect abnormal behavior in ICS and prevent cyber-attacks.

Time series anomaly detection is a type of anomaly detection that analyzes the temporal patterns of the system data to detect anomalies. It's a complex task that requires advanced techniques to identify patterns and anomalies in large datasets.

The problem of time series anomaly detection in ICS has gained increased attention due to the growing concern for cybersecurity in critical infrastructure. Traditional methods for anomaly detection, such as statistical methods or rule-based approaches, may not be effective in capturing subtle and evolving anomalies in time series data. Therefore, there is a need for advanced techniques that can effectively detect anomalies in ICS time series data.

In this project, we propose to use a VAutoencoder, a type of autoencoder that incorporates the principles of VAEs, for time series anomaly detection in ICS using the SWAT dataset. The SWAT (Secure Water Treatment) dataset is a well-known benchmark dataset for ICS cybersecurity research, containing time series data from a simulated water treatment plant system. The objective of this study is to develop an effective anomaly detection method using V-Autoencoder for ICS and compare its performance with traditional Autoencoder-based techniques.

Chapter 2

Literature review

In [1] the authors propose an unsupervised framework for anomaly detection in Cyber-Physical Systems (CPSs) using an Attention-based Spatio-Temporal Autoencoder (STAE-AD). The framework combines a convolutional autoencoder with an attention-based ConvLSTM encoder-decoder. The study focuses on physical layer attacks and uses the "Physical" subdirectory of the SWaT dataset, which contains 51-time series generated by sensors and actuators on a per-second basis. The model is trained on a dataset that includes 496,800 records collected under normal conditions and 449,919 records collected during various cyber-attacks. The model uses mini-batch stochastic optimization with the AMS-Grad optimizer and Mean Squared Error (MSE) as the objective function. The best performance of the model is achieved when the input and output sequence length is set to four, and the dropout rate in the recurrent layers is set to 0.3. STAE-AD is compared to five baseline methods from four different categories, and it outperforms the other deep learning models in terms of prediction accuracy, precision, recall, and F1 score.

The authors in [2] introduce a novel deep learning-based anomaly detection approach called DeepAnT for time series data. DeepAnT can detect various anomalies such as point anomalies, contextual anomalies, and discords. DeepAnT consists of two modules: the time series predictor and the anomaly detector. The time series predictor uses a deep convolutional neural network (CNN) to predict the next time stamp, while the anomaly detector tags the corresponding time stamp as normal or abnormal. The approach is evaluated on 10 different datasets and compared with 15 anomaly detection methods, including several state-of-the-art methods. The results show that DeepAnT outperforms the state-of-the-art anomaly detection methods in most cases while performing on par with others. The proposed approach is effective for detecting both point anomalies and contextual anomalies in time series data, including those with periodic and seasonal characteristics, and can also be applied to discord detection in time series. However, poor data quality and high levels of contamination may negatively impact the system's performance.

The authors in [3] propose two unsupervised machine learning methods, Deep Neural Network (DNN) and Support Vector Machine (SVM), for anomaly detection in a water treatment system (Reference). The authors use the SWaT dataset, which consists of network traffic, sensor data, and actuator data collected over 11 days of continuous operation. The DNN approach includes a layer of Long Short-Term Memory (LSTM) architecture followed by feedforward layers, while the SVM is a commonly used technique for anomaly detection. The performance of the proposed methods is evaluated in terms of precision and recall scores of detected anomalies in the attack log. The DNN approach shows slightly better overall F-measure and precision, while the SVM has slightly better recall.

The authors in [4] focus on detecting cyber-attacks in Cyber-Physical Systems (CPS) using an unsupervised learning approach based on Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN) and Cumulative Sum (CUSUM) method. The authors propose a prediction model that uses LSTM-RNN to predict the expected sensor data based on historical data. The deviations between the actual sensor data and the predicted outputs are then calculated, and the CUSUM method is applied to detect anomalies by computing the cumulative sum of sequence predictions. The authors evaluate their approach on a dataset obtained from a Secure Water Treatment (SWaT) testbed, which includes regular continuous operation and attack scenarios. The results demonstrate that the proposed method can detect 9 out of 10 attacks, although some false positives and unexplained anomalies are observed.

The authors in [5] propose a novel approach called TadGAN for time series anomaly detection using Generative Adversarial Networks (GANs). The authors address the challenge of unsupervised anomaly detection in time series data by leveraging the power of GANs, a type of deep learning model that consists of a generator and a critic. The generator learns to generate synthetic data samples that are like real data, while the critic learns to distinguish between real and synthetic samples. One of the key contributions of this paper is the use of LSTM Recurrent Neural Networks as base models for both the generator and the critic, which allows capturing the temporal correlations present in time series data. The generator is trained with a cycle consistency loss, which ensures that the generated data can be reconstructed back to the original data, making the generated data samples more realistic and representative of the real data distribution. The paper also proposes several novel methods to compute reconstruction errors and combine them with the critic outputs to compute anomaly scores. The authors conduct extensive experiments on 11 datasets with a total of 492 signals from various application domains and compare TadGAN with 8 baseline anomaly detection methods. The results show that TadGAN outperforms all baseline methods by achieving the highest averaged F1 score across all datasets and demonstrating superior performance in 6 out of 11 datasets.

Chapter 3

Background

in this section, we introduce some definitions related to the project.

3.1 Anomaly types

To understand more the problem, it's essential to understand the various types of anomalies that can occur. Anomalies can be classified into three categories: point anomalies, contextual anomalies, and collective anomalies.

-Point anomaly: This is the simplest anomaly category, also called global anomaly and individual anomaly, which refers to data points that deviate significantly from the expected behavior of the time series.in other words, if one object can be observed against other objects as an anomaly, this is a point anomaly.

-contextual anomaly: Also called a conditional anomaly, it happens when data points display abnormal behavior within a specific context or condition.in other words, if the object is anomalous in some defined context, it's a contextual anomaly.

-Collective anomaly: also known as group anomaly or pattern anomaly, involve a group of data points that together exhibit anomalous behavior, while individually, they may appear normal.in other words, if some linked objects can be observed against other objects as an anomaly, it's a collective anomaly.

3.2 anomaly detection techniques

Many methods can be used for anomaly detection in time series data. some of the techniques are:

1- Statistical methods: involve using mathematical models and measures to identify anomalies, such as z-score, standard deviation, and mean absolute deviation, those methods are used to define a threshold for identifying data points that deviate significantly from the expected behavior.

2- Machine Learning-Based Methods: it's one of the most used techniques in time series anomaly detection, supervised learning involves training a model on labeled data, and unsupervised learning aims to discover patterns and anomalies without prior labeling. machine learning algorithms most used are: autoencoder, LSTM, support vector machine(SVM), isolation forest

3- Time Series Decomposition: This technique gives you the ability to split your time series into its constituent components, such as trend, seasonality, and residuals.

4- Bayesian Methods: these approaches utilize probabilistic models to detect anomalies in time series data. These methods consider the prior knowledge and incorporate it with the observed data to estimate the likelihood of anomalies. Bayesian techniques can be beneficial when dealing with uncertain or limited labeled data for anomaly detection.

3.3 Time series anomaly detection

Time series anomaly detection refers to the process of identifying abnormal patterns, outliers, or deviations from the expected behavior within a sequence of data points ordered in time. Time series data is prevalent in various domains, including finance, cybersecurity, manufacturing, and sensor networks. Detecting anomalies in such data is crucial for identifying critical events, outliers, or potential anomalies that may indicate unusual behavior or indicate potential issues.

The goal of time series anomaly detection is to distinguish between normal patterns and anomalous instances. Anomalies can manifest in different forms, such as sudden spikes or drops, shifts in trends, unexpected patterns, or abnormal variations in seasonal or cyclic behavior. By detecting these anomalies, organizations can take appropriate actions, investigate the root causes, and potentially prevent undesirable outcomes.[6]

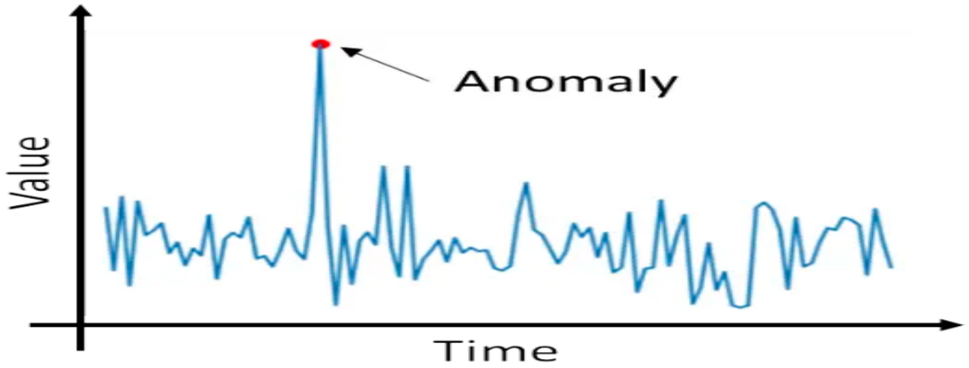
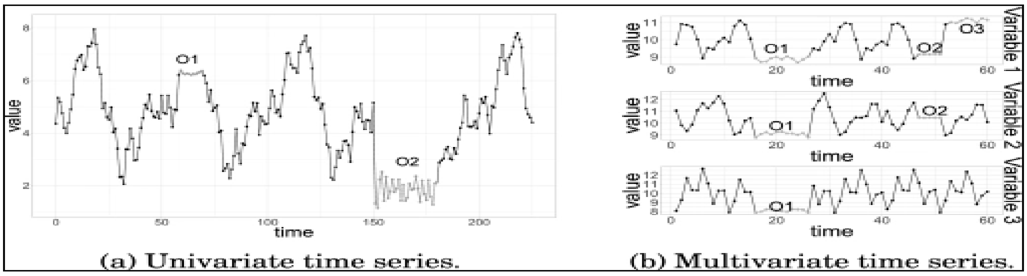


Figure 3.1: Time series anomaly detection

3.4 uni and multivariate time series anomaly detection

Time series anomaly detection can be performed on both univariate and multivariate time series data.



(a) Univariate time series. (b) Multivariate time series.

Figure 3.2: univariate/ multivariate Time series

- univariate time series anomaly detection Univariate time series anomaly detection focuses on analyzing and detecting anomalies in a single time series variable. In this approach, anomalies are identified based on deviations from the expected behavior of the individual variable over time. Various techniques can be applied like statistical methods, moving averages. . .

- multivariate time series anomaly detection Multivariate time series anomaly detection involves analyzing multiple variables simultaneously to detect anomalies. This approach considers the relationships and dependencies between different variables in the time series. Various techniques can be applied like Multivariate Statistical Methods, Clustering, and Dynamic Bayesian Networks...

Chapter 4

Methodology

4.1 Variational autoencoder

In this experiment, we proposed the use of a Variational Autoencoder (VAE) as the method for anomaly detection in the SWaT dataset. A VAE is a type of deep learning model that combines the power of both autoencoders and variational inference. It is well-suited for unsupervised anomaly detection tasks, as it can capture the complex patterns and dependencies present in the data.

The V-Autoencoder architecture consists of three main components:

Encoder: The encoder network maps the input time series data into a lower-dimensional latent space representation. It employs recurrent neural network (RNN) layers, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), to capture the temporal dependencies effectively.

The encoder equations can be represented as:

$$z = f_{\text{enc}}(X)$$

Here, X represents the input time series data, and f_{enc} denotes the encoder function. The encoder function can consist of multiple layers of fully connected neural networks with appropriate activation functions.

Latent Space: The latent space represents a compressed and abstract representation of the input data. It serves as an intermediate bottleneck layer between the encoder and decoder. The dimensionality of the latent space is a tunable hyperparameter that affects the model's capacity to capture meaningful patterns.

Decoder: The decoder network reconstructs the input data from the latent space representation. It aims to reconstruct the normal patterns accurately while highlighting any discrepancies that may indicate anomalies. The decoder typically mirrors the architecture of the encoder but in reverse.

The decoder equations can be represented as:

$$X' = f_{\text{dec}}(z)$$

Here, X' represents the reconstructed time series data, and f_{dec} denotes the decoder function. The decoder function can also consist of multiple layers of fully connected neural networks with suitable activation functions.

During the training process, the VAE learns to encode the normal patterns and structures of the SWaT dataset and reconstruct it accurately. Anomalies can be detected by comparing the reconstruction error of a new data point to a predefined threshold.

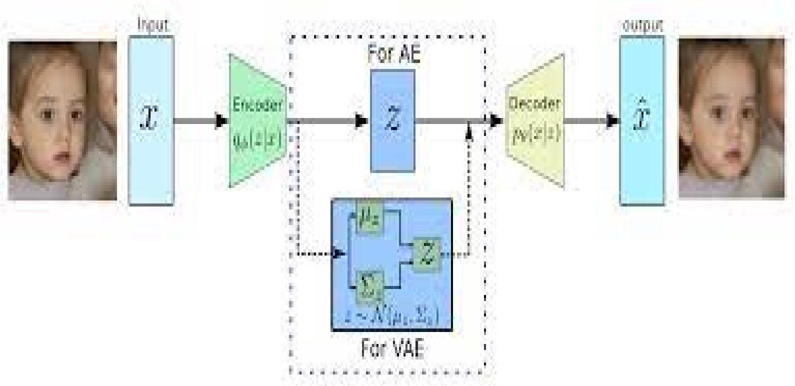


Figure 4.1: Variational Autoencoder (VAE)

4.2 dataset

In this experiment, we used SWaT Dataset [7].SWaT is a fully operational scaled-down water treatment plant that can produce 5 gallons/minute of filtered water. SWaT is a six-stage filtration process that mimics a large modern water treatment plant. It's designed to ensure the quality of the dataset by capturing both normal and attack data. The dataset consists of seven days of normal continuous operation and four days of attack. The SWAT dataset consists of multiple features, including but not limited to water flow rate, water pressure, pH level, temperature, and valve positions. The time series data is captured at regular intervals, providing a rich source of sequential information for anomaly detection algorithms. By utilizing the SWAT dataset, we aim to assess the effectiveness of the V-Autoencoder for time series anomaly detection in an industrial control system setting.

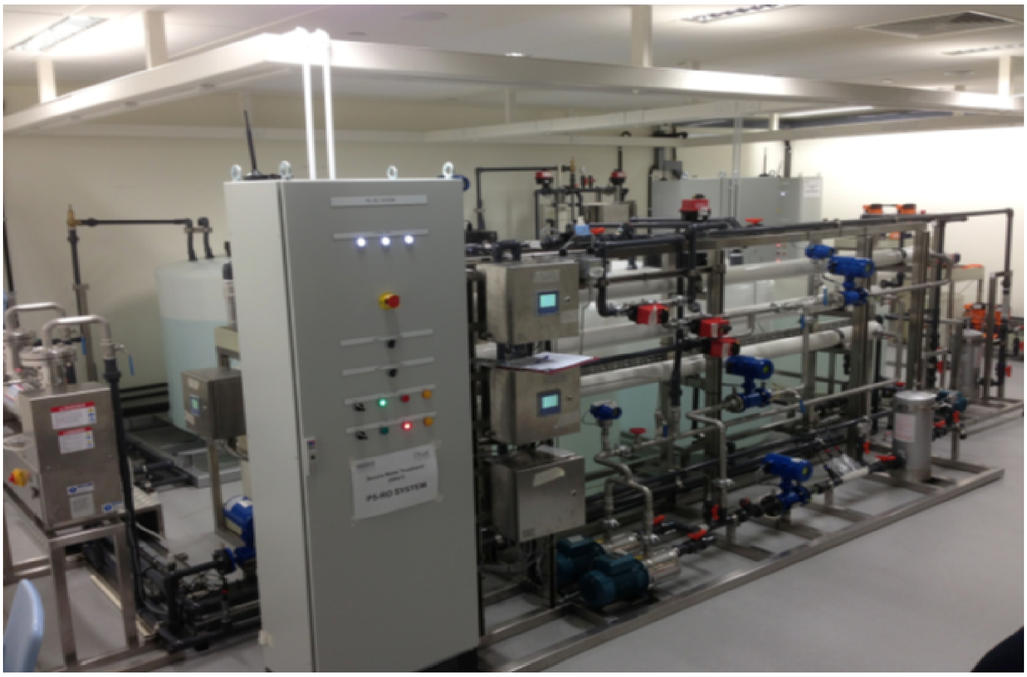


Figure 4.2: Actual Photograph of SWaT testbed

4.3 Preprocessing and Feature Selection

In this experiment, we performed several steps to prepare the dataset for anomaly detection, including handling the missing values, converting the Timestamp column to a datetime format, and removing duplicated entries and features with a unique values.

Next, we continued with the process of feature selection, taking into account the different types of features in the dataset. For numerical columns, we utilized min-max scaling to normalize the data. This scaling approach guarantees that all numerical features have a comparable range, avoiding any individual feature from overpowering the analysis.

In addition, we employed one-hot encoding to handle categorical features. This technique converts categorical variables into binary vectors, creating separate columns for each unique category.

To further reduce the dimensionality of the dataset, we implemented principal component analysis (PCA). By creating a PCA object with only 15 components, we transformed the data into a reduced feature space. This transformation is useful for capturing the most important patterns and variability in the dataset while minimizing the impact of less significant features.

Chapter 5

Experiments

In this section, we present the experimental setup, configuration, and evaluation metrics used to compare the performance of the baseline Autoencoder with the proposed V Autoencoder method. We also discuss the dataset used for the experiments.

5.1 Experimental Setup and Configuration

We conducted the experiments on a dataset consisting of time series data. The dataset was split into two separate subsets to facilitate the evaluation of our models: a training set and a test set. The training set consisted of 496,800 instances and was used to train the models. On the other hand, the test set, which contained the remaining portion of the dataset, was reserved for assessing the performance and generalization capability of the trained models.

To establish a baseline, we trained an Autoencoder model using the training set. The Autoencoder had an encoding dimension of 10. We then trained a V Autoencoder model, also with an encoding dimension of 10, using the same training set. Both models were trained for 10 epochs using the Adam optimizer with a batch size of 64. The mean squared error (MSE) loss function was used to measure the reconstruction error between the input and the output for both models.

5.2 Evaluation Metrics

To evaluate the performance of the models, we used the following metrics:

Mean Squared Error (MSE): The MSE was calculated between the reconstructed outputs and the original inputs for the test set. It provides a measure of the reconstruction quality.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\text{Original Input}_i - \text{Reconstructed Output}_i)^2$$

Threshold Selection: We determined a threshold value for classifying anomalies based on the MSE values. A higher MSE indicates a higher likelihood of an anomaly.

Accuracy: We calculated the accuracy of anomaly detection by comparing the predicted anomalies with the ground truth labels from the target values.

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

Precision: Precision measures the proportion of correctly identified anomalies among all the instances predicted as anomalies. It is calculated as the ratio of true positives (TP) to the sum of true positives and false positives (FP).

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

Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of actual anomalies that were correctly identified. It is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN).

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

F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance by considering both precision and recall. The F1-score is calculated as 2 times the product of precision and recall divided by the sum of precision and recall.

$$\text{F1-score} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}}$$

Chapter 6

Results

In this experiment, the performance of the baseline Autoencoder model was evaluated on the test set, yielding a Mean Squared Error (MSE) of 0.1192. Additionally, the anomaly detection accuracy achieved by the baseline Autoencoder was 92.19%. Further analysis revealed a precision of 0.2054, indicating that among the instances predicted as anomalies, approximately 20.54% were correctly identified. The recall, or true positive rate, was measured at 0.1020, implying that the baseline Autoencoder successfully identified 10.20% of the actual anomalies. The F1-score, which considers both precision and recall, was calculated as 0.1363.

In contrast, the proposed V Autoencoder method demonstrated improved results. The test MSE obtained for the V Autoencoder model was 0.1755. The accuracy of anomaly detection increased to 93.51%, indicating enhanced performance compared to the baseline model. The precision for the V Autoencoder was 0.2781, signifying that 27.81% of the instances predicted as anomalies were correctly classified. However, the recall for the V Autoencoder model was relatively lower at 0.0460, indicating that only a small proportion of the actual anomalies were identified. The F1-score for the V Autoencoder model was 0.0790, indicating a balance between precision and recall.

These results highlight the improved performance of the V Autoencoder compared to the baseline Autoencoder, particularly in terms of accuracy and precision. However, there is a trade-off with lower recall, suggesting that the V Autoencoder may miss some anomalies.

Model	Autoencoder	V-Autoencoder
Accuracy	0.9219	0.9351
Precision	0.2054	0.2781
Recall	0.1020	0.0460
F1-Score	0.1363	0.0790

To visualize the reconstruction error and identify anomalies, we plotted the reconstruction errors for both models on a line graph. Anomalies were marked above a threshold value determined based on the MSE values.

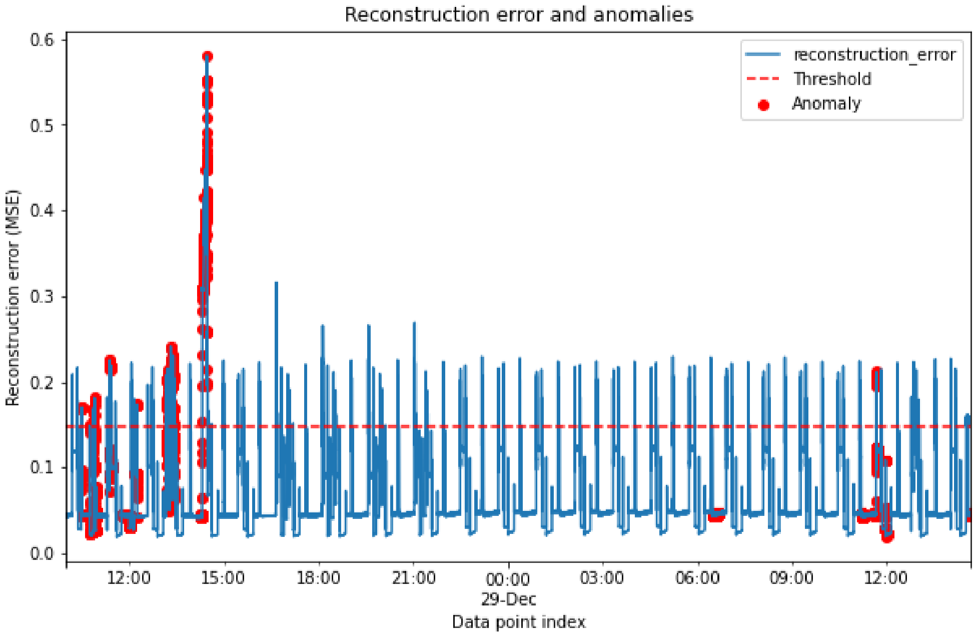


Figure 6.1: Autoencoder

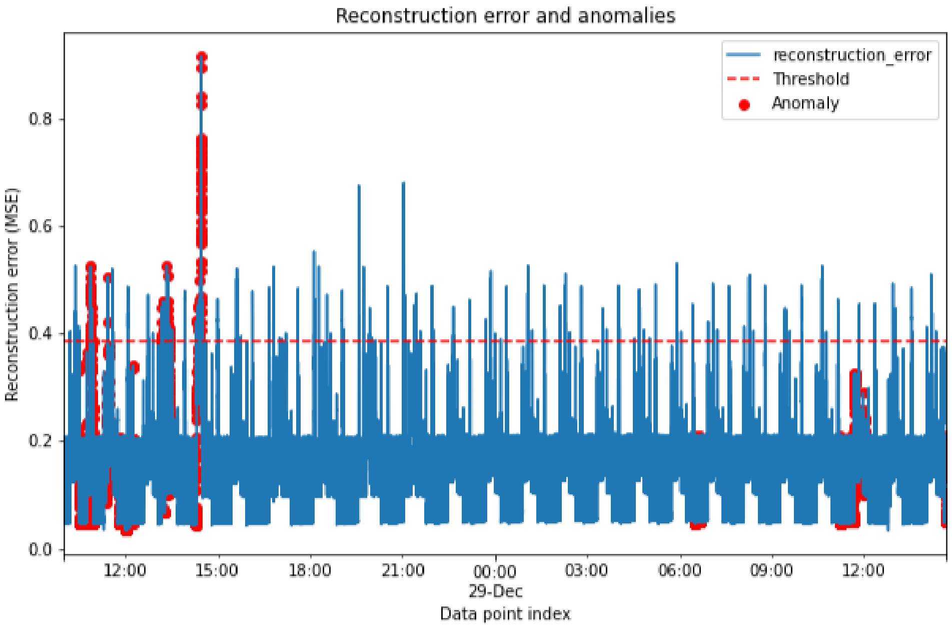


Figure 6.2: V-Autoencoder

The graph clearly shows the higher reconstruction errors corresponding to the identified anomalies.

Chapter 7

conclusion

In this study, we proposed a novel approach for anomaly detection in time series data using V Autoencoders. We compared our method with the baseline Autoencoder model and evaluated their performance on a real-world dataset.

The experimental results demonstrated that our V Autoencoder model achieved a Test MSE of 0.1754, slightly higher than the baseline Autoencoder's Test MSE of 0.1189. However, it is important to note that the accuracy of our V Autoencoder model was higher, reaching 93.51%, compared to the baseline Autoencoder's accuracy of 92.62%.

Furthermore, we analyzed additional evaluation metrics such as Precision, Recall, and F1-Score. Our V Autoencoder model achieved a Precision of 27.81%, Recall of 4.60%, and F1-Score of 7.90%. These results indicate that our method successfully identified anomalies in the time series data, although with relatively lower recall and F1-Score compared to precision.

Overall, our V Autoencoder model demonstrates promising potential for anomaly detection in time series data. The higher accuracy achieved by our model suggests its effectiveness in accurately identifying anomalies. However, further improvements are needed to enhance recall and F1-Score to achieve a more balanced performance in identifying true anomalies.

This study contributes to the field of anomaly detection in time series data by introducing a novel approach that utilizes V Autoencoders. Future research can focus on refining the V Autoencoder architecture, exploring different threshold selection techniques, and investigating strategies to improve recall and F1-Score.

List of Figures

Bibliography

[1] Chunming Wu Mayra Macas, ed. *An Unsupervised Framework for Anomaly Detection in a Water Treatment System.* No. 38 Zheda Road, Hangzhou 310027, China, 2019.

[2] ANDREAS DENGEL MOHSIN MUNIR SHOAIB AHMED SIDDIQUI and SHERAZ AHMED, eds. *DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series.* Germany, 2019.

[3] Yuqi Chen Christopher M. Poskitt Jun Inoue Yoriyuki Yamagata and Jun Sun, eds. *Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning.* Ikeda, Japan Singapore, Singapore, 2017.

[4] Marcus Tan Jonathan Goh Sridhar Adepu and Lee Zi Shan iTrust, eds. *Anomaly Detection in Cyber Physical Systems using Recurrent Neural Networks.* Singapore, 2017.

[5] Sarah Alnegheimish Alfredo Cuesta-Infante Alexander Geiger Dongyu Liu and Kalyan Veeramachaneni, eds. *TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks.* 2020.

[6] Yuncong Cheny Xinyang Fengz-Cristian Lumezanuy Wei Chengy Jingchao Niy Bo Zongy Haifeng Cheny Nitesh V. Chawlax Chuxu Zhangx Dongjin Songy, ed. *A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data.* USA, 2018.

[7] Khurum Nazir Junejo Jonathan Goh Sridhar Adepu and Aditya Mathur, eds. *a Dataset to Support Research in the Design of Secure Water Treatment Systems.* Singapore, 2016.