# Enhancing Anomaly Detection in Time Series Data Using Hybrid Deep Learning Methods

MSc Research Project
Msc in Data Analytics

Wonderpaul pallikona
Student ID: X23218142

School of Computing
National College of Ireland

Supervisor:Musfira Jilani

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** ……. ………………Wonderpaul pallikona……………

**Student ID:** ……………………X23218142…………….

**Programme:** …………Msc in Data analytics…………… **Year:** …2024-2025

**Module:** …………Msc research project…………….

**Supervisor:** …………………Musfira Jilani…………………

**Submission Due Date:** 29/01/2025

**Project Title:** Enhancing Anomaly Detection in Time Series Data Using Hybrid Deep Learning Methods

**Word Count:** 6266 **Page Count: 21**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** …………Wonderpaul pallikona…………………………

**Date:** ………………29-01-2025…………………

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Enhancing Anomaly Detection in Time Series Data Using Hybrid Deep Learning Methods

**Abstract**

**Time series anomaly detection is a critical task for detecting irregularities or deviating patterns in the data which may indicate system failure, fraud, etc. This research focuses on detecting anomalies using various deep learning techniques, including LSTM, GRU, RNN based Autoencoders. The New York City taxi data is employed in the study to detect anomaly patterns in time series data. Deep learning models like RNN-Autoencoder and its variants such as LSTM and GRU based autoencoders are trained in unsupervised setting by minimizing the reconstruction error, and an anomaly is found when the reconstruction error is above certain specified threshold. Although the main goal of this research is not to make accurate predictions but predicting anomalies in the dataset. This research evaluates the various auto encoder variants based on MAE, MSE and RMSE based on reconstruction error on test data. The analysis of the results illustrates that the RNN Autoencoder has the lowest MAE, MSE, and RMSE scores of 0.0457, 0.0036, and 0.0600, respectively with the highest $R^2$ score of 0.8897. The LSTM Autoencoder also shows high performance, while the GRU Autoencoder, which has the advantage of having lower computational complexity provides relatively lower performance. The results of the proposed hybrid deep learning models are then compared with a more conventional approach, the Isolation Forest, to demonstrate the strengths of deep learning in identifying nonlinear temporal relationships in time series data. The results show that by using deep learning methods, anomaly detection performance can be enhanced by a huge margin that provides affordable solutions for real-time monitoring and decision-making in numerous domains including operational excellence, fraud prevention, and predictive maintenance.**

**Keywords: Anomaly Detection, LSTM, RNN, GRU-Autoencoders, Isolation Forest**

## 1. Introduction

Most businesses and industries including financial, health, manufacturing sectors, and cybersecurity collect a lot of time series data in the present data-driven world. Time series data consists of ordered sequence of data points collected at certain time intervals, typically patterns indicate normal behavior or abnormal, in the form of anomalies. Identifying such anomalies is always important because it can help identify major events such as fraud in financial activities, failure in industrial processes, or cyber-security incidents and breaches (Schmedl & Papenbrock et al., 2022). An anomaly detection task aims to find deviations that may be marginal and challenging to differentiate from normal data outliers.

The anomaly detection from time series data points out that more conventional approaches to anomaly detection, including statistical methods or simpler machine learning algorithms, are not efficient at treating complicated and high-dimensional time series data. These approaches usually depend on assumptions made on the data distribution or pre-set boundary conditions, which often result in high false positive values or do not model out the complex, non-linear interactions between samples which are common in sequential data (Ji & Fang, 2021). For a long time, time series data has increased in terms of both, the amount of data and the intricateness of the patterns it forms, and therefore, one needs to develop more versatile methods, which can capture temporal dependencies that are inherent to sequential data.

Various research have demonstrated that deep learning offers an improved solution to the limitations of traditional methods especially with models capable of learning temporal relations and patterns in the sequential data. From them, Recurrent Neural Networks (RNN), and its improved variant models of LSTM and GRU have achieved tremendous success in capturing sequential data. Additionally, Autoencoders, which are unsupervised neural networks trained to down sample the representations and reconstruct the data have been used in detecting outliers with the help of reconstruction error that is difference between the input data and the reconstructed data. Together, these models can potentially provide a strong basis for research work in detecting anomalies in time series data through the synergistic combination of sequence modeling and reconstruction error-based anomaly detection.

This research aims to use enhanced deep-learning model that combines RNN with auto encoders and also its variants such as LSTM Autoencoders and GRU Autoencoders to detect anomalous regions in time series data. Each model comes with different benefits to perform anomaly detection. LSTM Autoencoders work best with time series data because they can identify long-range temporal dependencies that are characteristic of complex, non-linear, time-varying patterns. RNN Autoencoders can be considered as another variant which are essentially simpler and may have advantages in conditions where the data is characterized by shorter temporal dependencies or when the speed of analysis is particularly important. And also GRU Autoencoders are midway between LSTMs and RNNs in terms of computation and effectiveness in addressing long range dependency. For large datasets or real-time anomaly detection, these models can be effective. This research, therefore, proposes to use these described models to detect anomalies in time series data especially in unsupervised setup. These models will enable the modeling of temporal dependencies needed to capture more complex dependencies and identify anomalies behaviors based on the Autoencoder's reconstruction errors.

## 1.1 Motivation

The inspiration for this research is given by the drawbacks of the usual anomaly detection methods that cannot identify the complicated nonlinear patterns of time series data. Simple techniques such as statistical thresholding or basic machine learning approaches are ineffective when confronted with high-dimensional datasets, or sequences where long-term dependencies appear. One of the best solutions to integrate LSTM, GRU, and RNN networks with Autoencoders to detect anomalies in time series data based on reconstruction-based approach. Such models can learn temporal dependencies and identify subtle anomalies and do in the absence of labeled data. This research predict the anomalies in time series using various variants of time series models with auto encoders and predicts anomalies using reconstruction error based approach.

### 1.2 Research Question

This research seeks to answer the following question:

***"How effectively can LSTM, GRU, and RNN Autoencoder based deep learning provide more accurate and efficient anomaly detection on time series data, especially on conditions that the data is not labeled/structured?"***

### 1.3 Research Objectives

The primary objectives of this research are:

- Leveraging the various hybrid deep learning approaches like LSTM, GRU, & RNN networks with Autoencoders to anomalies in time series data.
- Reconstruction-Based Anomaly Detection: To utilize Autoencoder's reconstruction loss for the detection of anomalies, the comparison of LSTM, GRU, and the traditional RNN-based Autoencoders in this task.
- Unsupervised Anomaly Detection: To establish a framework that can be used in time series analysis to detect anomalies other than using labeled datasets with the help of unsupervised learning methods.
- Performance Evaluation: The effectiveness of these hybrid models will be compared will be compared based on reconstruction error with various evaluation metrics

This research will contribute to the field of anomaly detection in time series data especially through the use of hybrid deep learning. LSTM, GRU, and RNN with autoencoders in which the research intends to use to improve efficiency and scalability of Anomalous Detection of time series datasets. The expected findings of this study are expected to assist organizations to embody real-time detection and identification of anomalies enhancing the dependability of their decisions and boosting organizational performance.

## 2. Literature Review

Anomaly detection in time series data is a critical task; detecting unpredictable patterns in the data, which could signal major events, for example, equipment malfunction, fraud, or security incidents. A number of methods ranging from the initial statistical methods to the modern deep learning approaches have been designed to address the issues associated with anomaly detection for time series data. This literature review aims to review these approaches, including their advantages and limitations and, more importantly, the developments made in these techniques in the recent past.

### 2.1 Traditional Anomaly Detection Methods

Standard deviation techniques were largely used in the traditional anomaly detection and are very useful when one is sure of the kind of data distribution. Statistical anomaly detection methods, for instance, z-score analysis, measure how far away from the norm is a certain data point (Jamshidi et al., 2022). The other method is control charts that are graphical displays of data in which the data point is plotted against time so that any deviations from expected can easily be discovered based on predefined limits (Montgomery, 2019). However, these methods work best when the environment is somewhat static and deal with the established patterns only and may not update with the new data distribution thus missing out on new anomalies (Montgomery 2019). Similarly, time series decomposition methods, such as Seasonal-Trend decomposition using LOESS (STL), et al., 2024) separate a time series into its constituent components: The consumer behaviour pattern is influenced by trend, seasonality and irregularity as depicted by (Yildiz et al., 2024). After decomposition, the analyst can investigate certain residual in order to observe certain anomalous. These methods are efficient in excluding spikes and dips from seasonal trends but tend to fail when dealing with datasets characterized by intricate interdependencies, which poses certain impracticality constraints (Sharif et al., 2022).

### 2.2 Machine Learning Approaches

The application of machine learning approaches has greatly enhanced anomaly detection making it possible to identify complicated data characteristics and work well in dynamically changing situations. Most Support Vector Machines (SVM) are popular due to their efficiency in high dimensional space. One-class SVMs, in turn, introduce the concept of anomaly detection as a binary classification problem where the aim is to define a hyperplane that separates the data into normal and anomalous data instances as far as possible (Schölkopf et al., 2001; Pang et al., 2022). While useful, SVMs tend to need sensitive parameterization and can also be problematic when dealing with large quantities of features, thereby increasing the computation time. Another type of machine learning method is ensemble learning which we also mentioned in the previous section including Random Forest and Isolation Forest and appears to be quite suitable for anomaly detection. These techniques are developed by combining sets of weak learners to form a tree and capture interactions in the data (Samriya et al., 2023; Suboh et al., 2023). Isolation Forests work by randomly isolating data into subsets, and the anomalies are easily detected based on the path length. Although ensemble methods show significant performance gains, they do not incorporate time dependency which is desired in time series analysis.

### 2.3 Deep Learning Techniques

Deep learning methods have advanced understanding and modeling beyond linear and simple temporal patterns that can be highly non-linear and more complex in the temporal domain. Many authors have used Recurrent Neural Networks (RNNs) and its enhancements, namely Long Short-Term Memory (LSTM) networks, for time series anomaly detection since their architecture preserves information across sequences. RNNs, however, suffer from long range dependency problems, and vanishing gradients, which makes it hard for the network to remember information from several preceding time steps. To overcome these problems, LSTM networks were invented with the help of gating mechanisms for controlling the flow of information between time steps, in a way that the model is capable of capturing long-term dependencies without experiencing gradient problems (Hochreiter & Schmidhuber, 1997). Traditional forms of recurrent neural networks, namely LSTMs, have been popular in time series anomaly detection because of their temporal relationship understanding and the identification of discrepancies from learned patterns (Terbuch et al., 2023). However, they are computationally intensive and may take considerable time and hyperparameter optimization to provide the best results more so when working on large and complex data structures (Al-selwi et al., 2023).

The GRUs are another variant of RNN, and the best alternative for LSTM. GRUs have fewer parameters, are computationally simpler, and are widely used in time series anomaly detection scenarios that require capturing long dependencies and are computationally constrained. Even though GRUs tend to be less accurate than LSTMs when it comes to identifying long range dependencies, they have been reported to perform equally well in many time series anomaly detection problems, especially where the data does not contain such dependencies (Schmidl & Papenbrock, 2022). For instance, in one of the studies, the Convolutional Neural Networks (CNNs), designed mainly for image processing, were employed for time series anomaly detection with an assumption that the sequential data is a 1D signal. CNN has a high ability to detect local patterns and trends in data and when combined with RNN, extraction of both spatial and temporal features enhances the performance of the models that detect anomalies (Wen et al., 2019).

## 2.4 Comparative Analysis of Recent Literature Review

In the rapidly evolving area of anomaly detection in the time series data, various methods have been suggested, which apply different approaches, and have their advantages and disadvantages. In Table 1 presents the method and result of the review of studies of similar subjects, to gain a better understanding of their approach and conclusions.

**Table 1: Comparative Review Analysis of Existing Research**

| Authors | Datasets Used | Methodology | Model Used | Metric Value | Limitations | Future Work |
|---|---|---|---|---|---|---|
| Sharif et al., 2022 | Daily Price Data for S&P 500 index | LSTM Autoencoder | LSTM | Precision: 0.89 | Requires large labeled datasets | Explore unsupervised approaches |
| Al-selwi, et al. 2023 | Synthetic datasets | Sequence modeling | RNN | Accuracy: 92% | Computationally intensive training | Investigate adaptive learning techniques |
| Terbuch. A, X., et al. 2023 | Energy consumption data | Hybrid approach | VAE and LSTM | Enn: 0.1997 | Complexity in model integration | Implement real-time anomaly detection |
| Samariya et al., 2023 | Various benchmark datasets | Statistical methods | Isolation Forest, Statistics etc. | Precision: 0.85 | Limited to linear anomalies | Combine with deep learning techniques |

| | | | | | | |
|---|---|---|---|---|---|---|
| Wang et al. 2023 | CIFAR-10 Dataset | AC Conditional GANs | Conditional AC GAN (DNN classifier) | Precision: 0.92 | May struggle with imbalanced classes | Enhance with anomaly threshold adjustment |
| Suboh et al. 2023 | High Dimensional Dataset | Ensemble methods | PCA, Random Projection OCP method | Accuracy: UPTO 88% | Difficult to interpret model outputs | Incorporate explainability methods |
| Schölkopf, B., et al. 2001 | High-dimensional datasets | Support vector machines | One-class SVM | AUC: 0.88 | Requires extensive parameter tuning | Explore kernel-based techniques |
| Zamanzadeh et al. 2022 | Benchmarks SOTA dataset | Hybrid deep learning | MAD and GAN | N/A | Complexity in parameter tuning | Investigate multi-modal data integration |

This research aims to compare the various techniques used by recent studies based on anomaly detection in time series data. Each paper offers different ideas and tools, some of which have better results in terms of measurements and drawbacks. Although the hybrid models including LSTM, RNN, and GRU with Autoencoders are promising, the problems include the lack of a large, labeled dataset, high computational cost, and interpretability of the results. More work needs to be done in learning algorithms that can be sensitive to the nature of the data being analyzed, the use of approaches where the target system is not labeled, and the improvement of models that can be easily explained.

## 2.5 Challenges

Even though there has been significant improvement in various methods, several issues still arise in anomaly detection methodologies. Most conventional techniques do not consider the temporal relations present in the time series data; on the other hand, most machine learning algorithms demand labeled data which may sometimes be scarce. The deep learning models are effective but may need more computation power and lots of hyperparameters to tune. Furthermore, Autoencoders-based methods do not have enough parameters to be adjusted to accommodate different data characteristics, which explains the need to develop other mixed methods capable of learning anomaly thresholds based on changes in data patterns (Chen et al., 2023). Further studies are needed to improve the combination of anomaly detection methods in a single model and improve the models' ability to process real-time information.

This paper has presented a literature review of anomaly detection techniques in time series data, and the advantages and disadvantages of the traditional methods, machine learning, and deep learning. Out of them, models that use LSTM Autoencoders, RNN Autoencoders, and GRU Autoencoders have gained prominence because they capture temporal dependencies and data structures to enhance the chances of anomaly detection. These models have the following advantages: Despite these advantages, issues like interpretability, real-time detection, and scalability are still an issue in these models.

Future work should be aimed at addressing these issues to generate new approaches for building more accurate, explainable, and easily scalable models optimized for real-time processing of various and dynamically changing time series data.

# 3. Methodology

The approach implemented in this research to improve anomaly detection on time series data using various deep learning models, namely LSTM, GRU, RNN, with Autoencoder as shown in figure 1. The present work aims to enhance the effectiveness of the anomaly detection activity by adopting the benefits of these models effectively. This study employs data from New York City taxi time series data which has significant amount of anomalies in data due unexpected events over the time which effects the number of taxi booked. This section describes pre-processing of the data, training of the models, identification of anomalies, and assessment of the models.

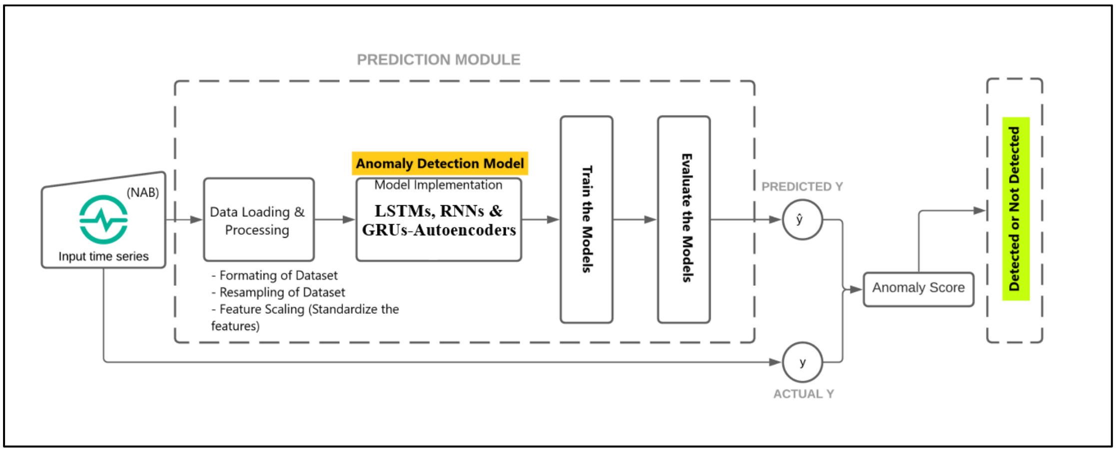


**Figure 1: Methodology for Anomaly Detection**

## 3.1 Data Collection

The work employs the NAB (Numenta Anomaly Benchmark) dataset and contains various time series data across different domains such as energy, traffic, and financial. The particular type of data selected for this research is the NYC Taxi dataset which records the number of passengers at various intervals.

```
# Display first few rows of the dataset
df.head()
```

| | timestamp | value |
|---|---|---|
| 0 | 2014-07-01 00:00:00 | 10844 |
| 1 | 2014-07-01 00:30:00 | 8127 |
| 2 | 2014-07-01 01:00:00 | 6210 |
| 3 | 2014-07-01 01:30:00 | 4656 |
| 4 | 2014-07-01 02:00:00 | 3820 |

```
# Display last few rows of the dataset
df.tail()
```

| | timestamp | value |
|---|---|---|
| 10315 | 2015-01-31 21:30:00 | 24670 |
| 10316 | 2015-01-31 22:00:00 | 25721 |
| 10317 | 2015-01-31 22:30:00 | 27309 |
| 10318 | 2015-01-31 23:00:00 | 26591 |
| 10319 | 2015-01-31 23:30:00 | 26288 |

**Figure 2: NYC Taxi Dataset Overview**

## 3.2 Data Preprocessing

The first step of the data preprocessing includes preparing the New York City taxi data which involves a set of data points over a given period. The time step column of data was converted to datetime type for chronological data analysis. Moreover, the main interest in the time series taxi ride count was determined, and their values were normalized using the MinMaxScaler. This scaling was done to normalize the data between zero and one so that deep learning models would run efficiently and be insensitive to the scale of the input data.

The obtained preprocessed data was then arranged into time sequences with the help of the rolling window method. This technique produces time windows (or sequences) from the dataset that work as input to the models. These sequences were then divided into the training set and the test set, with a usual 80/20 division, so there would be more than enough data to train the models and test their performance using reconstruction error. The sequences were also rearranged to sort it in the input needed for the deep learning models such as LSTM, RNN, and GRU based autoencoders.

### 3.3 Model Architecture and Training

#### LSTM Autoencoder

Deep learning-based LSTM Autoencoder was trained on the time series data. In this model, LSTM units are used, which are an adequate solution in the work with temporal sequences. It had an encoder-decoder architecture, where the encoder used LSTM layers to reduce the size of the input sequence into a lower dimensional space. The decoder then maps this latent space back to sequence. A bottleneck layer was incorporated into the system to reduce the reconstruction error during model training.

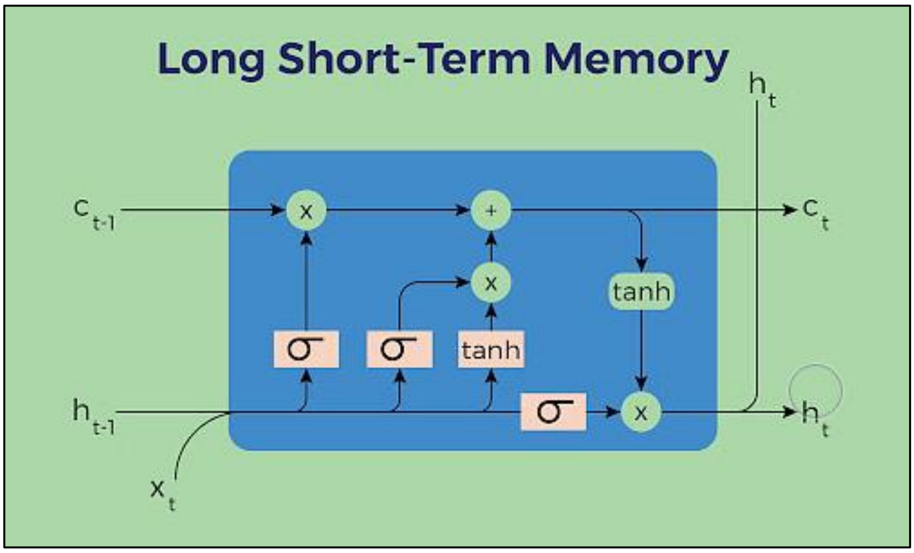


**Figure 3: Architecture of Long Short-Term Memory (LSTM)**

The model was trained by using the Adam optimizer and the Mean Squared Error loss function was selected to avoid large reconstruction errors. The model was trained for 30 Epochs with a batch size of 32. Reconstruction error was calculated during the training process and also for both the training and the test data set. The anomalies were detected based on reconstruction errors above a fixed error limit; the limit used was the

95th percentile of the reconstruction errors in the training set. This threshold approach makes it possible to identify cases of variation from the expected data outcomes.

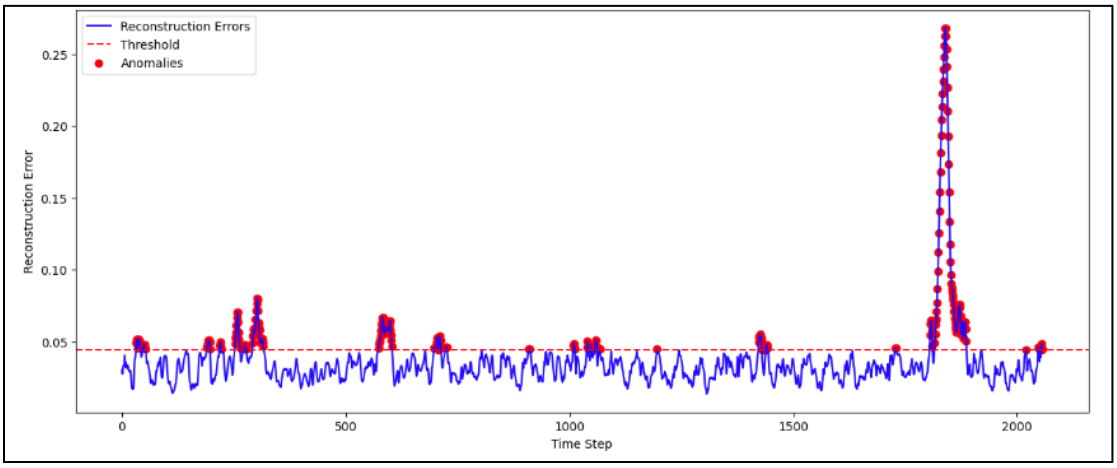


**Figure 4: Reconstruction errors on test data using LSTM-Autoencoder**

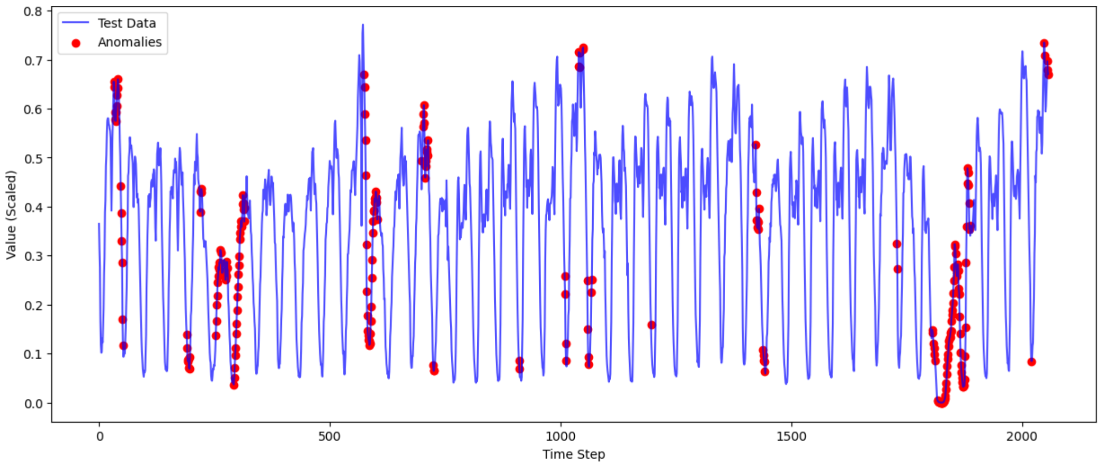


**Figure 5: Predicted anomalies using LSTM-Autoencoder**

## RNN Autoencoder

RNN Autoencoder model whose structure was similar to that of the LSTM Autoencoder model but uses RNN instead of LSTM. Although LSTM networks are built to reduce the impact of vanishing gradient problems, RNNs contain simpler structures and can still characterize short-term dependencies. The RNN Autoencoder was constructed analogously to the LSTM model with the RNN layers performing the role of the encoder part of the model and the decoder reconstructing the input sequence.

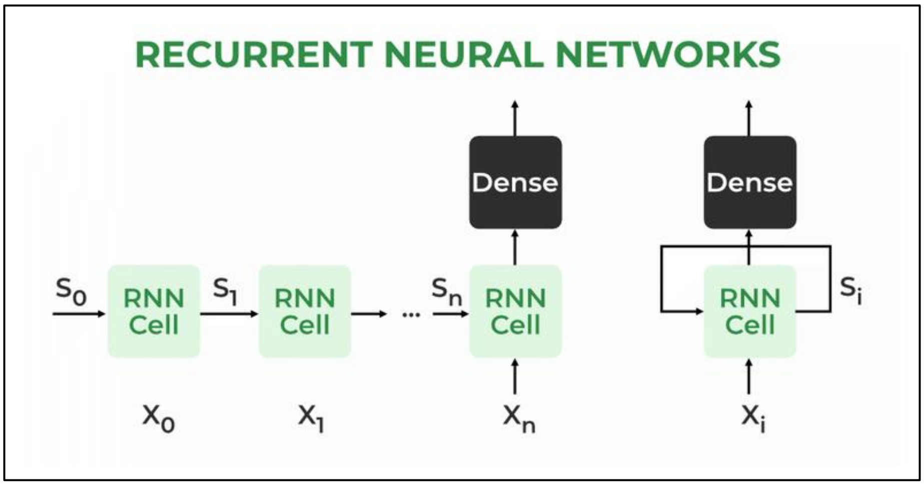


**Figure 4: Architecture of Recurrent Neural Network (RNN)**

The training of the RNN Autoencoder was also similar to that of the LSTM Autoencoder, the same optimizer and loss function settings were used. Anomalies were detected by identifying differences between the original input sequences and the reconstructed sequences, where the anomaly threshold was defined at the 95th percentile of the training reconstruction errors.

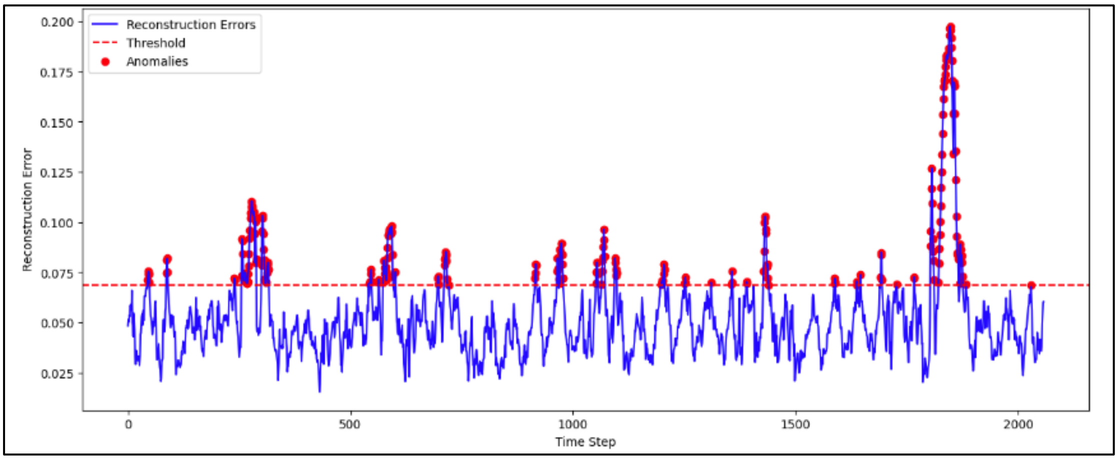


**Figure 6: Reconstruction errors on test data using RNN-Autoencoder**

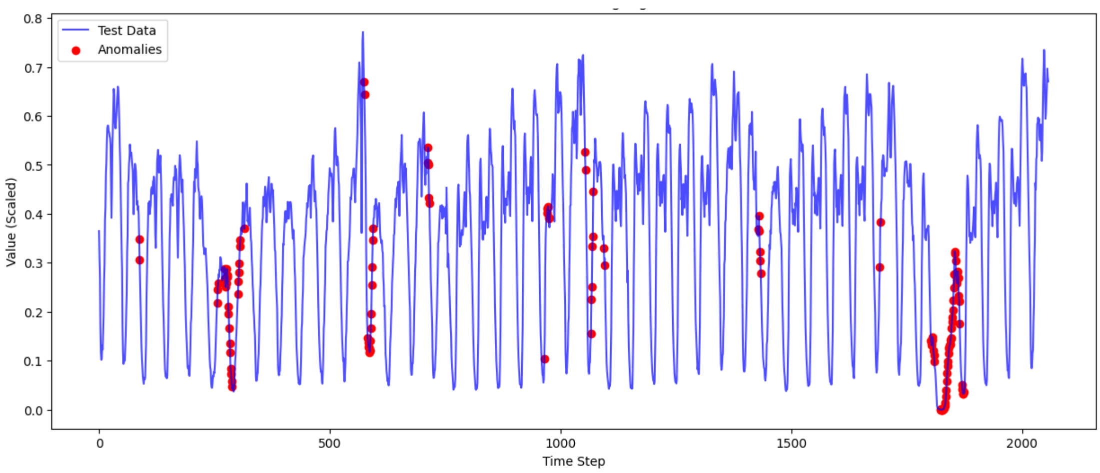


**Figure 7: Predicted anomalies using RNN-Autoencoder**

**GRU Autoencoder**

The third model used in the study was the GRU Autoencoder. Another type of the RNN is the Gated Recurrent Units (GRUs) which are developed to enhance the training process and the efficiency of the model with the help of gating mechanisms. LSTMs on the other hand have high computational costs while GRUs are optimised to retain their capability of capturing long-range dependency. The same encoder-decoder structure was applied to the creation of the GRU Autoencoder, with the GRU units instead of LSTM and RNN units.

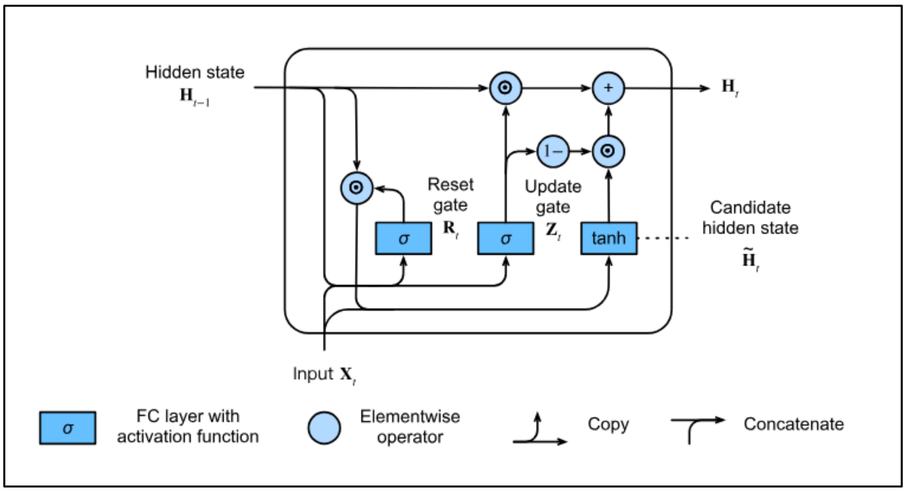


**Figure 8: Architecture of Gated Recurrent Unit (GRU)**

As with the earlier models, the GRU Autoencoder model was trained with Adam optimization and Mean Squared Error loss. The anomalies were determined based on the differences in the reconstruction errors for each sequence with the threshold being 95% of the training errors.

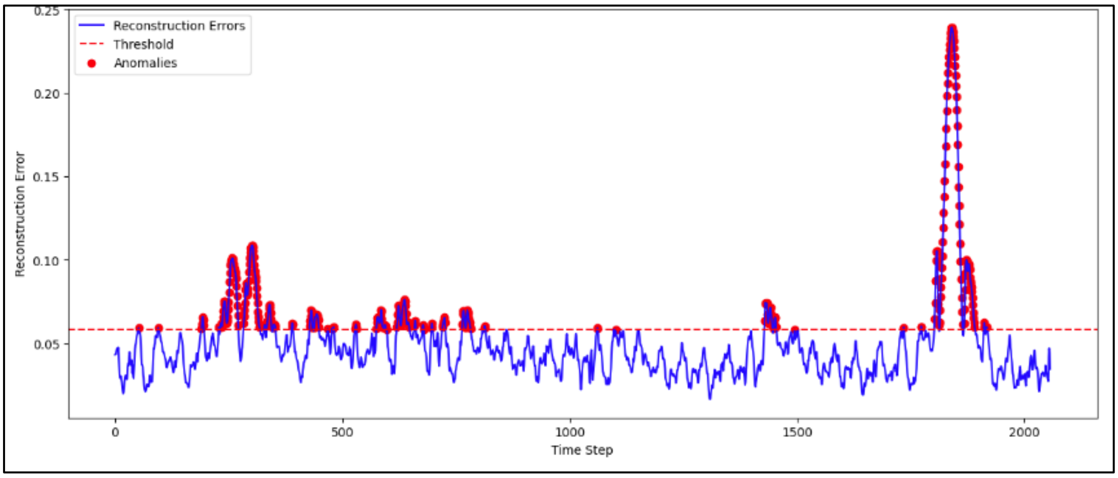


**Figure 9: Reconstruction errors on test data using GRU-Autoencoder**

## 3.4 Anomaly Detection and Evaluation

All the deep learning models LSTM, RNN, and GRU based autoencoders are trained under similar. The main goal of the model is to reconstruct input sequences, and a large difference between the input and the reconstructed sequences indicated possible anomalies. These errors were computed using basic statistical measures for error estimation including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Therefore, the coefficient of determination $R^2$ was also calculated to know the extent of variation dealt with by the model. These metrics were used on the training and test set offering an overall model performance.

These results from each model were then compared and visualized data from each model was plotted using reconstruction error plots and anomalies were indicated as outliers. Furthermore, to judge how effectively the models identify reconstruction, the original and reconstructed data were graphically represented.

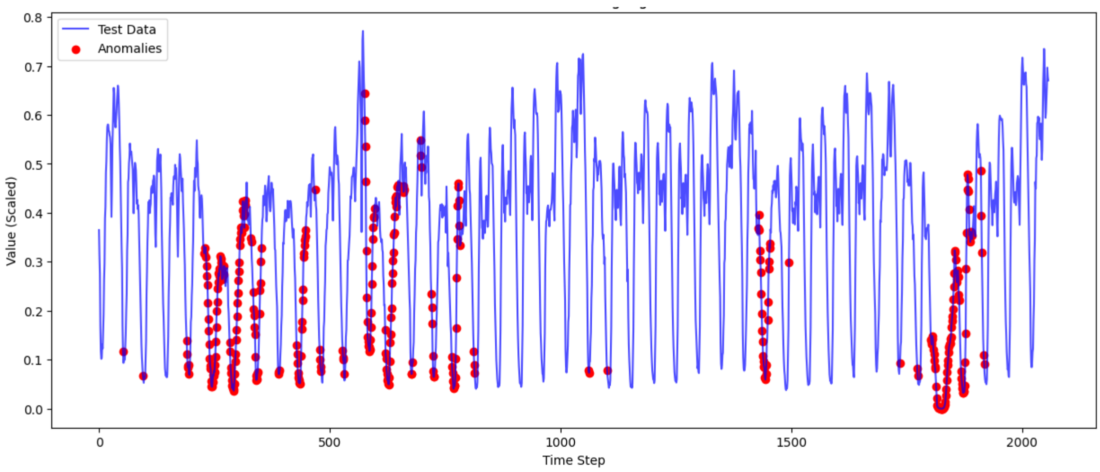


**Figure 10: Predicted anomalies using GRU-Autoencoder**

## 3.5 Benchmark Model: Isolation Forest

Besides the deep learning models, a benchmark traditional machine learning algorithm, namely Isolation Forest was also utilized in the study. One of the most known algorithms for anomaly detection is called Isolation Forest which isolates objects based on partitioning data into smaller subgroups and constructing random trees. The contamination parameter for Isolation Forest is 0.05 which means that nearly 5% of data points were considered to be outliers. The training phase was performed to train the Isolation Forest model, and the anomaly score of the test data was calculated. Like in the case of the deep learning models, the anomalies found by the Isolation Forest model have been visualized through the reconstruction error plots.

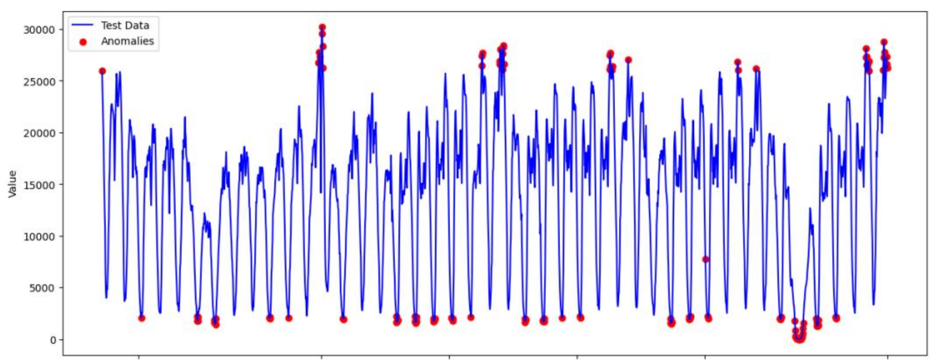


**Figure 11: Predicted anomalies using Isolation Forest model**

As deep learning models are able to reconstruct the data and based on the reconstruction error, able to predict many anomalies compared to isolation forest. From observation of figure 11 can see that only extreme anomalies can be predicted using isolation forest but unable to predict many anomalies in data compared to trainable deep learning models.

### 3.5 Model Comparison and Analysis

In this step, testing the performance of the various deep learning models, (LSTM, RNN, GRU) is computers using various metrics based on reconstruction error. This comparison was performed based on performance comparison of the detected anomalies in the reconstructed data. The idea was to show that deep learning models, especially the combined models, can be used to achieve a higher level of anomaly detection in time series data.

By means of this approach, the research intends to identify the anomaly detection in time series data and show the benefits of employing multiple deep learning models within a single framework. In this study, various models for real-time anomaly detection can be detected efficiently compared to traditional methods like isolation forest. Due to the capacity to improve on how time series data are analyzed to detect anomalies, this research can be employed to develop and refine theory and practice in organizations using time series data for forecasts and monitoring.

## 4. Model Evaluation Results & Discussion

The models LSTM Autoencoder, RNN Autoencoder, and GRU Autoencoder are evaluated by using of MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and $R^2$ Score with the NYC Taxi Dataset. These metrics were essential when trying to assess how well the models were able to reconstruct time series data and identify anomalies. Each of the models is described below and their advantages and disadvantages are then compared.

**Table 2: Performance Results of Models**

| Metric | LSTM Autoencoder | RNN Autoencoder | GRU Autoencoder |
|---|---|---|---|
| **Mean Absolute Error** | 0.0549 | 0.0457 | 0.0690 |
| **Mean Squared Error** | 0.0052 | 0.0036 | 0.0079 |
| **Root Mean Squared Error** | 0.0724 | 0.0600 | 0.0889 |
| **$R^2$ Score** | 0.8397 | 0.8897 | 0.7578 |

### 4.1 LSTM Autoencoder

The examined LSTM Autoencoder has proved efficient in reconstructing the time series data with a MAE value of 0.0549, MSE value of 0.0052, RMSE value of 0.0724. These results show a moderate level of reconstruction error which is common in real world time series data due to the presence of complicated non-linear dependency structures.

The acquired $R^2$ of 0.8397 shows the LSTM trained model was able to account for 84% of the variance in the data confirming the model's ability to capture the temporal dependencies in the data.

The comparatively high $R^2$ score shows that the LSTM appropriately captured long-periodical patterns which is one of the key strengths of the LSTM on a time series dataset. All the reconstruction errors are within reasonable range here, but they also show that the model might be slightly more accurate in finding smaller violations of the normal processes.

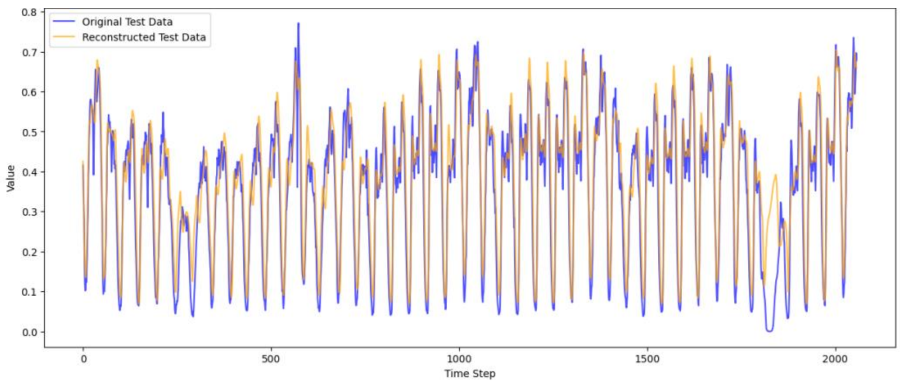


**Figure 12: Actual test data and Reconstructed test data using LSTM-Autoencoder**

### 4.2 RNN Autoencoder

RNN Autoencoder performed better than the LSTM model in this study with MAE of 0.0457, MSE of 0.0036, and RMSE of 0.0600. They show a greater reconstruction accuracy as the RMSE value is lower in comparison with LSTM model. This suggest that the RNN Autoencoder was better at reconstructing the time series data and had fewer large errors, which provided better modelling of the underlying characteristics. There was marginal improvement compared to LSTM where the $R^2$ score 0.8897, means that the RNN Autoencoder was capable of explaining about 89% of the variability in the data.  RNN is simpler model compared to LSTM. This might have averted the problem observed in the LSTM and performed marginally better than LSTM as it was able capture the short-range dependency in the data.

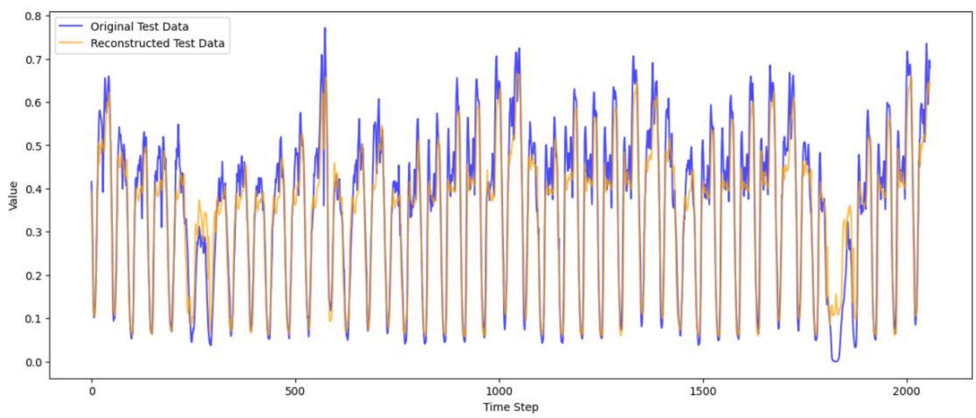


**Figure 13: Actual test data and Reconstructed test data using RNN-Autoencoder**

### 4.3 GRU Autoencoder

The GRU Autoencoder with a MAE of 0.0690, MSE of 0.0079 and RMSE of 0.0889 suggest that the GRU model had the highest reconstruction errors out of the three models, which signifying that the GRU model had the least performance in terms of reconstructing the time series data out of the three models. The use of the GRU has resulted in an $R^2$ score of 0.7578 meaning that the technique was able to account for variation of around 75% of the values. Even though GRUs are supposed to be less computation intensive than LSTMs, their inferior performance in the present study indicates that the model had difficulty in capturing the dependencies of the data. The number of parameters which are required in the GRU is less than that required in LSTM and the gating mechanisms used in GRU are less complex and hence the GRU may have failed to capture more complex dependencies in the series data. Nevertheless, the calculation indicates that one of the merits of this structure is its computational efficiency which would be an advantage in real time application where time is constrained. Nevertheless, GRU model results of reconstructions are not as efficient as other models, but it could still be used in time constrained environment.

### 4.4 Discussion

The analysis of the outcomes shows that there are some advantages and limitations of each of the discussed models in case of anomaly detection for time-series data. Mostly all models are able to detect the anomalies efficiently but GRU model with more error have results in more anomalies. Based on the comparison of the results, LSTM Autoencoder demonstrated reasonable performance, however, since the values for RMSE and reconstruction errors are slightly higher it could be assumed that there is a certain margin for the improvement of the model's precision in terms of anomalies detection.

Instead, the results obtained with the RNN Autoencoder show a better reconstruction performance than the LSTM model in terms of RMSE and $R^2$. This means that RNN was better suited to the identification of short-term dependencies of the taxi data than long-term dependencies, which could be less nuanced and easier to predict. It is suitable to be used in applications where higher performance and less complex data patterns are required. This makes it easier to train and use than the other models due to the simple architecture of the RNN making them suitable for real-time anomaly identification.

In fact, the GRU Autoencoder has a slightly lesser performance compared to the other two models but offers the following advantages in terms of computational complexity. It had the highest reconstruction errors but because it performs data processing faster than other algorithms, the GRU could be best suited for real time anomaly detection scenarios where the highest accuracy is not as much of a priority as speed. In some applications such as handling huge amounts of data in real-time the GRU's performance might be enhanced compared to its accuracy; however, it may require additional calibrations and adjustments.

These results emphasize the balance between model complexity and performance together with the computation time. Despite the rich set of deep learning models such as LSTM, RNN, and GRU which could be used for anomaly detection in time series data, the selection of model is largely dependent on the requirements of the particular application. For example, if the priority is accurate, and the data contain long temporal dependencies, LSTM is the most suitable choice. But if real time processing and speed is an issue, there could be enhancements with the RNN or the GRU.

By comparing with conventional methods, such as Isolation Forest, more complex and non-linear temporal features were learned by the deep learning models. Although the algorithm of the Isolation Forest allows detecting anomalies based on the provided contamination rate, it has a limitation of not being able to model the time series trend, or temporal dependency in observations. This makes the deep learning models appropriate for problems in which the temporal characteristics of the data are important.

# 5. Conclusion & Future Work

## 5.1 Conclusion

This study focused on the applying of hybrid deep learning techniques including LSTM, RNN and GRU Autoencoders to improve the identification of anomalies in time-series data using reconstruction errors. Specifically, the primary goal was to enhance the ability of the models detect anomalies in temporal patterns of the New York City taxi dataset. The performance of each model was evaluated based reconstruction error using various metrics

The RNN Autoencoder model emerged to be the most efficient with an MAE of 0.0457, MSE 0.0036 and RMSE 0.0600. Its $R^2$ score is 0.8897, thus it can reconstruct time series data and detect anomalies to an extent of 89%. The LSTM Autoencoder performed reasonably well with an MAE of 0.0549, MSE of 0.0052, RMSE of 0.0724 and, $R^2$ score of 0.8397, and was also capable of capturing long term dependencies, however it had higher reconstruction errors than the RNN. On the other hand, the performance of the GRU Autoencoder model was comparatively lower with the MAE of 0.0690, MSE of 0.0079, RMSE of 0.0889, whereas the R-squared score of the model was 0.7578.

On comparing with traditional anomaly detection methods like Isolation Forest, it became clear that deep learning models are more effective when it comes to modeling complex non-linear relationship that exist in the time series data. Although methods such as Isolation Forest can be used for finding outliers based on predetermined assumptions, the combined deep learning models turned out to be better suited in detecting more subtle and complex patterns characteristic to the time series data and thus providing more accurate results in practical applications.

Overall, this research offers strong evidence of the effectiveness of hybrid deep learning models, especially the RNN Autoencoder, to improve anomaly detection in time series data and therefore offers both theoretical and practical contribution to the field of operational monitoring, fraud detection, and decision making.

## 5.2 Future Work

Future work of this study aims to explore:

- Multivariate Time Series: The current research was conducted using univariate time series data whereby only one variable in the New York City taxi dataset was considered. Future work should expand these models to the multivariate time series data types which encompass multiple variables that interact in time. Finding outliers in multivariate data is even more difficult and it is the next step to develop models of the dependencies between variables for practical use, for example, in the monitoring of multi-sensor data or financial prediction.

- Evaluation on Diverse Datasets: To support the presented deep learning models, the future work should assess these models on other real-time data series originating from different domains, for example, energy usage data, financial transactions, and sensor systems. Thus, by applying the models to various types of time series data, it is possible to evaluate their transportability and influence of the approaches on their performance.

These future directions can be used as a guide to extend the current study, resulting in improved anomalous time series classification using deep learning approaches. These enhancements will not only advance the analysis of deep learning for anomaly detection based on development of theories, but also give more objective and effective solutions for the effective applications in real-time procedures in multiple fields.

# References

Sharif, M. H., Gupta, K., Mohammed, M. A., & Jiwani, N. (2022). Anomaly detection in time series using deep learning. *International Journal of Engineering Applied Sciences and Technology*, *7*(6), 296-305.

Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., & Muneer, A. (2023). LSTM inefficiency in long-term dependencies regression problems. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *30*(3), 16-31.

Terbuch, A., O'Leary, P., Khalili-Motlagh-Kasmaei, N., Auer, P., Zöhrer, A., & Winter, V. (2023). Detecting anomalous multivariate time-series via hybrid machine learning. *IEEE transactions on instrumentation and measurement*, *72*, 1-11.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Samariya, D., & Thakkar, A. (2023). A comprehensive survey of anomaly detection algorithms. *Annals of Data Science, 10*(3), 829-850.

Wang, H., Miller, D. J., & Kesidis, G. (2023). Anomaly detection of adversarial examples using class-conditional generative adversarial networks. *Computers & Security*, *124*, 102956.

Suboh, S., Aziz, I. A., Shaharudin, S. M., Ismail, S. A., & Mahdin, H. (2023). A Systematic Review of Anomaly Detection within High Dimensional and Multivariate Data. *JOIV: International Journal on Informatics Visualization*, *7*(1), 122-130.

Chen, X., Deng, L., Zhao, Y., & Zheng, K. (2023, February). Adversarial autoencoder for unsupervised time series anomaly detection and interpretation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (pp. 267-275).

Montgomery, D. C. (2019). *Introduction to statistical quality control*. John wiley & sons.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the Support of a High-Dimensional Distribution. Neural Computation, 13(7), 1443-1471.

Karadayı, Y., Aydin, M. N., & Öğrenci, A. S. (2020). A hybrid deep learning framework for unsupervised anomaly detection in multivariate spatio-temporal data. *Applied Sciences*, *10*(15), 5191.

Wen, T., & Keyes, R. (2019). Time series anomaly detection using convolutional neural networks and transfer learning. *arXiv preprint arXiv:1905.13628*.

Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., & Salehi, M. (2022). Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*.

Lee, C. K., Cheon, Y. J., & Hwang, W. Y. (2021). Studies on the GAN-based anomaly detection methods for the time series data. *IEEE Access*, *9*, 73201-73215.

Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. Proceedings of the VLDB Endowment, 15(9), 1779-1797.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ismail Fawaz, H., Forestier, G., Weber, J., & Keogh, E. (2020). Deep learning for time series classification: A review. Data Mining and Knowledge Discovery, 34(6), 1-39. https://doi.org/10.1007/s10618-020-00619-1

Ji, Z., Gong, J., & Feng, J. (2021). A novel deep learning approach for anomaly detection of time series data. Scientific Programming, 2021(1), 6636270.

Schmidl, S., Wenig, P., & Papenbrock, T. (2022). Anomaly detection in time series: a comprehensive evaluation. Proceedings of the VLDB Endowment, 15(9), 1779-1797.

Ali, Y. A., Awwad, E. M., Al-Razgan, M., & Maarouf, A. (2023). Hyperparameter search for machine learning algorithms for optimizing the computational complexity. Processes, 11(2), 349.

Pang, J., Pu, X., & Li, C. (2022). A hybrid algorithm incorporating vector quantization and one-class support vector machine for industrial anomaly detection. IEEE Transactions on Industrial Informatics, 18(12), 8786-8796.

Jamshidi, E. J., Yusup, Y., Kayode, J. S., & Kamaruddin, M. A. (2022). Detecting outliers in a univariate time series dataset using unsupervised combined statistical methods: A case study on surface water temperature. Ecological Informatics, 69, 101672.

Yildiz, C. Y., & Kundakcioglu, O. E. (2024). Optimization for Time Series Decomposition. In Encyclopedia of Optimization (pp. 1-9). Cham: Springer International Publishing.