USING AUTOENCODERS FOR THE DETECTION OF WORKOUT ANOMALIES

Term Project

Soumili Das #0815660

[Solo Project]

Project Summary

This project has been completed independently by me. In this project, I have worked on building a system for detection of irregular patterns in data sequences. The main aim of this project was to build a model that could identify anomalies in the heart-rate patterns, a sudden spike or drop in the heart-rate, indicative of serious health issues or sensor malfunctioning.

My contributions in this project are broken down into the following parts:

- Comprehending the Problem:

Firstly, the main issue was defined: identification of an irregular pattern by comparing it to a normal data pattern. This is an important part as it helps in monitoring health as well as to ensure that the fitness trackers are working effectively.

- Working with the data

The heart rate data was collected from FitBit and the preprocessing steps were performed in order to prepare the data for further analysis. I had cleaned the data for analysis by handling missing values, and performing standardization to make it easier for analysis. Furthermore, the data was resampled in order to ensure that every reading was recorded at a regular 1-minute interval and the values were standardized in order to make the scale more consistent. Since, there were no real-world anomalies in the dataset, synthetic anomalies were incorporated into it in order to test the model's capability to detect irregular patterns.

- Building the Models

In this project, I had built two advanced machine learning models to asses their capability of anomaly detection and compare them.

i. The LSTM Autoencoder was used to first compress the data into a smaller form and then reconstruct it into the original sequence. It is noted that LSTM Autoencoders learn the patterns with time.
ii. The Transformer Autoencoder focusses on both local and global patterns simultaneously. The special "attention" mechanism of Transformer Autoencoder helps to asses which part of the dataset is to be focussed on. Thus, this makes the Transformer Autoencoder sensitive to subtle changes and nuanced deviations as well.

- Testing the Models and Evaluating the Results

Both the models were trained using normal data sequences so that they could learn what a 'normal' pattern looks like. The models were then tested using a dataset containing both normal and irregular patterns. The results were then evaluated and visualized to compare their performance.

From understanding the data to testing and evaluating the models, every part of the project was done by me. Through this project, I have gained a valuable experience in working with time-series data and using advanced machine-learning algorithms for solving real-world problems. I plan to work further with this project and perform a detailed quantitative analysis on large data-sets with actual irregularities in it.

TABLE OF CONTENTS

1. Introduction

An anomaly is defined as a data point which is different from the remaining dataset. According to Hawkins, an anomaly is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism (Hawkins, 1980). The analysis and detection of anomalies is very a very crucial part as it helps us to obtain valuable insights into the characteristics and generation process of the data. Among the various anomaly detection techniques, the purpose of spectral anomaly detection technique is to obtain lower dimensional embeddings of the data, where we expect the anomalous and original data to be distinctive of each other. After finding these lower dimensional embeddings, we bring back these values to the original data space. This is called reconstruction of original data. B doing this, we expect to obtain the correct nature of the original data, devoid of various unnecessary noise and features. Reconstruction error can be defined as the error between the original data point and its low dimensional reconstruction. This reconstruction error is used as the anomaly scores to detect possible outliers in the dataset, mostly with the help of PCA (Principal Component Analysis) based methods. For the purpose of generalization, we have a scarcity of data-driven methods to automate the feature generation process. As of now, the features for anomaly detection are mostly chosen based on simple statistics (i.e., calculation of mean, median or standard deviation) or domain expertise.

In order to address this issue, autoencoders come into play. An autoencoder is a form of artificial neural network, typically designed for unsupervised learning tasks, that learns an efficient way of data representation. The main purpose of autoencoders is to perform dimensionality reduction by stacking up layers for the formation of deep autoencoders (An & Cho, 2015). It first compresses the input data into smaller dimensional spaces (known as encoding) and then reconstructs the original data from the compact representation (known as decoding). The main aim of autoencoders is the reduction of differences between the input and its reconstructed version, learning the vital features of the data effectively.

1.1 Project Goal

The main aim of this project was the development and evaluation of anomaly detection models for time-series heart-rate data, utilizing deep learning architectures. My goal was to identify anomalous patterns in the data that deviate from the original data, indicating physiological irregularities, unusual activities, or errors in the data. In this project, I have implemented two models – LSTM Autoencoder and Transformer Autoencoder for the detection of anomalies in the dataset. Furthermore, I have combined both the models in order to effectively detect anomalies in the dataset.

This project mainly focusses on the implementation of these algorithms, how one of them was better in detection of anomalies, along with their use in real-world scenarios, like fitness tracking applications and monitoring overall health. With systematic analysis of their performance, the optimum solution for anomaly detection based on their sensitivity to irregular features and overall computational efficiency has been recommended.

Summary of Results:

Both the LSTM Autoencoder and the Transformer Based Autoencoder were trained using a set of training data and were used to reconstruct normal patterns and detect anomalies based on the reconstruction errors.

- The LSTM Autoencoder has shown extreme deviations in the heart-rate patterns, thus, successfully detecting anomalies in the data. The model was computationally efficient and thus, appropriate to use in real-time fitness tracking applications.
- The Transformer Autoencoder model was comparatively more sensitive to understated and more distinctive irregular patterns in the data, thus, capturing both local and global patterns effectively. The model was more computationally exhaustive. However, this model is appropriate for scenarios that require a detailed analysis and prior detection of anomalies.

Both the models have indicative the same anomalous sequence, thus, proving their validation for reliability. The reconstruction error has proved to be an effective measure in differentiating between normal and irregular sequences of data.

Note: The dataset that I have used was enough to correctly demonstrate the capabilities of both the models for anomaly detection with regard to reconstruction errors. However, on increasing the size and diversity of the dataset, specifically with the incorporation of synthesized anomalies in the dataset, the overall performance of the model would enhance, ensuring their ability to simplify unseen data more effectively.

2. Tools and Algorithms Used

2.1 Dataset Description

I have used a dataset consisting of heart-rate time-series data that has been collected from a fitness tracking application (FitBit). The pre-processed FitBit dataset consisted of a total of 12,348 records and a total of 20 columns, specifying daily activities, heart rate, and sleep data.

2.2 Tools Used

1. **Programming Language:** Python was used as the primary language for the implementation of algorithms, preprocessing the data, and plotting visualizations.
2. **Frameworks and Libraries**:
   i. Deep Learning: I have used TensorFlow for the development, training, and evaluation of the LSTM Autoencoder and the Transformer Autoencoder. For construction and experimentation of the deep-learning models, I have used Keras.
   ii. Data Manipulation: I have used Pandas library for data cleaning, preprocessing, and manipulation. For the array operations and effective numerical computations, I have used the NumPy library.
   iii. Data Visualization: I have used Matplotlib for plotting effective visualizations for the time-series data, reconstruction errors and the results. I have used Seaborn enhanced statistical plots and visualizations.

iv. Evaluation Metrics: I have used Scikit-learn library for the calculation of evaluation metrics like precision, recall, and thresholds values. I have also implemented this library for preprocessing and splitting the dataset into training and testing sets.

3. **Supporting Tool:**

The entire project was done on Jupyter Notebook for interactive development of the code and its results.

2.3 Algorithms Used:

1. **LSTM Autoencoder:**

LSTM Autoencoder has been defined as an autoencoder where both the encoder and the decoder of the model are LSTM networks. LSTM Autoencoders are suitable for detecting anomalies in time-series datasets because they are able to learn patterns in the data over long sequences. The use of the LSTM cell is to capture temporal dependencies in multivariate data (Nguyen, Tran, Thomassey, & Hamad, 2021). Malhotra says that, an encoder-decoder model learned using only the normal sequences can be used for detecting anomalies in multivariate time-series (Malhotra et al., 2016). The main purpose of the encoder-decoder model was to only see normal instances during training the dataset and learn the process of reconstruction of the same. In case of anomalous sequences, it leads to higher errors and thus, a not-so-well reconstruction of the data.

Architecture of LSTM Autoencoder

- Encoding: In this process, the input time-series data is compressed into a concise latent representation by learning only the most important features.
- Decoding: In this process, the input data is reconstructed from the latent representation of the data. This process aims to represent the original data sequence.
- Reconstruction Error: In this process, the difference between the original input and the reconstructed data has been computed and used for the identification of anomalies in data.

LSTM Autoencoders are excellent in capturing of temporal dependencies. They are also very useful in the detection of sharp and extreme deviations in the patterns of data (Maleki, Maleki, & Jennings, 2021). LSTM Autoencoders are computationally very efficient, which makes them appropriate for real-time datasets.

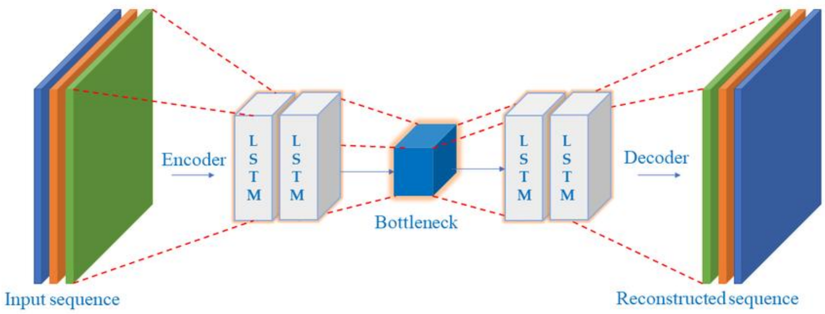


Fig 1. Basic Structural Representation of LSTM Autoencoder [6]

2. Transformer Autoencoder:

The Transformer Autoencoder model is a form of neural network architecture that is built in order to learn insightful representation of the input data and reconstruct the data based on these representations. The Transformer Autoencoder utilizes the strength of self-attention mechanisms, for the effective processing of sequential data.

Architecture of the Transformer Autoencoder:

1. Transformer Encoder: The primary purpose of the transformer encoder system is to encode the input sequence into a latent representation, capturing the most important features. It consists of three mechanisms:

    - Positional Encoding: As transformers do not process data sequentially by default, positional encoding is additionally implemented into the input data, thus, providing insights into the order of the elements in the sequence.
    - Multi-head relative attention mechanism: This is used to concentrate on the different parts of the sequence concurrently. Its main role is to compute the value of relation between each element in a sequence to the other. This helps the model to apprehend both short-term and long-term dependencies.
    - Position -wise fully-connected feed-forward network: By ensuring non-linear transformation of the data, the learned representation of the data is refined by the feed-forward network.

2. Latent Space: This is the intermediate layer where the concise, encoded representation of the input layer gets stored (Wang & Wan, 2019). Its main purpose is to remove noise and repeated information from the dataset, thus capturing its most relevant features.
3. Transformer Decoder: The main goal of the transformer decoder system is to reconstruct the original sequence of data from its latent representation. It again consists of three mechanisms:
    - Preparation of Input: In order to reintroduce the original sequence, in this process, the decoder takes the latent representation of data as input and adds positional encoding to the same.
    - Self-Attention and Cross-Attention Mechanism: In order to concentrate on the elements of the reconstructed sequence, the decoder implements self-attention mechanism (Liu & Liu, 2019). However, to align this with the encoded representation of data, the decoder uses the cross-attention mechanism.
    - Feed-forward Network: Similar to the Transformer Encoder, the feed-forward layer’s mechanism is to refine the reconstructed sequence of data before producing the output.

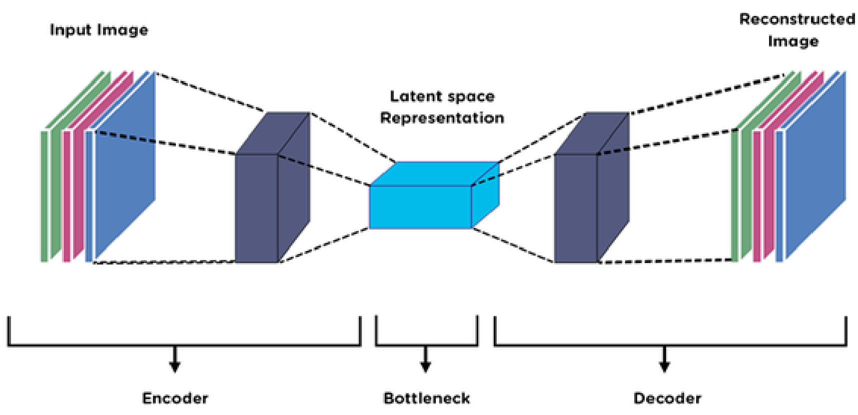


Fig 2. Diagrammatic Representation of Autoencoder Transformer

(Deepak Birla, 2019)

2.4 A Comparative Analysis of the Algorithms

FEATURE	LSTM AUTOENCODER	TRANSFORMER AUTOENCODER
STRENGTHS	It captures only temporal dependencies and is very computationally efficient	It captures both local and global patterns and is very highly sensitive to even subtle irregularities
WEAKNESSES	It may miss understated trends in the data	Its computational cost is high and the training process requires more resources
USE	It is used for anomaly detection of real-time datasets	It is used for a detailed data analysis and early detection of anomalies

Strengths and weaknesses aside, the comparison of LSTM Autoencoder and Transformer Autoencoder extended to show that the former had quite a considerable reduction in terms of computational overhead, fitting well with resource-constrained applications, while the latter fell short of capturing complex patterns (Liu et al., 2022). Transformer Autoencoders, instead, were very apt at catching subtle as well as strong anomalies by dwelling on the relationships across all sequences. However, the significant computational cost involved in Transformers means that training and inference require the use of GPUs.

Both methods uncovered unique use cases. LSTM Autoencoders were used when there was a strong need to monitor real-time anomaly detection, for example, to observe live streams of fitness data associated with an unusual heart rate (Mitiche, McGrail, Boreham, Nesbitt, & Morison, 2021). The applications of Transformer Autoencoders emerged while exploring history in reverse and making more general questions of anomalous activity patterns or sleep disturbances.

2.5 Performance and Assessment Indicators

The threshold for reconstruction error was set dynamically by analysing the distribution of errors in the validation dataset. While both models performed well, the Transformer Autoencoder outperformed the LSTM Autoencoder in terms of reconstruction errors and efficiency of anomaly detection, mainly due to its ability to learn complex dependencies. The practical implications of these findings for fitness monitoring and healthcare are tremendous. It can detect anomalies in heart rate or sleep patterns at real time to alert users about potential health concerns. It will analyse historical trends and anomalies to enable a comprehensive health check over time. This hybrid approach ensures decisions based on data are timely and detailed.

3. Experiment Design

3.1 Problem to be Solved:

In this project, the main issue addressed was the detection of irregularities in time-series heart-rate data that was obtained from a fitness tracker application called FitBit. The anomalies present in this data would help us to identify unnatural physiological occurrences, sensor issues, or irregular workout patterns. Identifying these consequences is a very crucial task for applications in healthcare monitoring, fitness tracking applications, or any wearable technology.

The goal of this project was to build a strong solution that was able to:

- Identify both extreme and subtle patterns in the dataset
- Have a proper understanding of the difference between gradual heart rate increase during the time of workout or sudden spikes or drops in the heart rate (genuine anomalies)
- Provide with high detection accuracy, thus, minimizing the false positives.

In order to address this issue, this project has implemented two deep learning architectures, namely, LSTM Autoencoder and Transformer Autoencoder. The goal of both of these models was to reconstruct the normal data sequences and the value of reconstruction error was used to detect anomalies in the data sequence.

3.2 Dataset Chosen and the Modifications made to the Data:

Dataset Chosen

- The chosen dataset consisted of the heart-rate time series data that has been collected from a fitness tracker application called FitBit.
- Every data point included the following:
    1. Value: This consisted of the beats per minute recorded at a timeframe
    2. Timestamp: The specific time of recording

3.3 Steps for Preprocessing the Data

In order to prepare the dataset for training and evaluation, the following steps have been performed:

- Data Cleaning: In order to maintain the quality of data, missing values were either removed or interpolated with assumed values.
- Data Resampling: In order to standardize the time intervals, the Timestamp data was resampled uniformly at 1-minute intervals.

- Normalization of the Data: Using MinMaxScaler, the heart-rate values were normalized into a range of [0,1].

Mathematical Formula Used:

X'=[X-min(X)]/[max(X)-min(X)]

This process of normalization was conducted in order to ensure that the data was uniformly processed without being affected by the differences in scaling.

Injecting Synthetic Anomalies in the Data

In order to test the anomaly detection capabilities of the model, synthetic anomalies were injected into the dataset. These anomalies were identified as:

- Sharp sudden spikes or drops in the heart rate
- Unusual oscillations beyond normal increase in the heart rate during the time of exercise.

3.4 Splitting the Dataset

For implementation of the autoencoder models and further analysis of the data, the dataset was split into training and testing sets:

- Training Set: The training set consisted only of the normal sequences, so as to make the model learn about how the normal patterns look like.
- Testing Set: The testing set consisted of both the normal sequences a well as the anomalous sequences of data in order to acknowledge the ability of the model to detect anomalies effectively.

3.5 Application of the Algorithms Used in the Project:

3.5.1 LSTM Autoencoder:

The implementation of LSTM Autoencoder was performed according to the following steps:

Step 1: Defining the Model Architecture of LSTM Autoencoder

- The LSTM Autoencoder was subdivided into two main parts:
    i. Encoder: The input sequence was compressed by the sequential LSTM layers into a very concise latent representation of the same, thus, capturing only the essential features.
    ii. Decoder: From the latent representation of the original input sequence, the sequential LSTM layer reconstructed the original sequence of data.

Step 2: Training the Model

- The model was trained only on the basis of normal sequences of data in order to make the model learn how the normal sequences look like. This helped them to better differentiate between normal and irregular sequence.
- The process of training the model taught the model to produce normal sequences effectively and accurately, thus, minimizing the reconstruction error.

Step 3: Testing the Model

- The trained model was now applied to the test data, that consisted of both normal and anomalous sequences.
- By testing the model, we were able to comprehend the model's capabilities to detect anomalies in the data.

Step 4: Calculation of Reconstruction Error

The value of reconstruction error was calculated for every sequence of data by calculating the difference between the original data point and the reconstructed data point.

Step 5: Anomaly Detection based on Threshold

- The 85th percentile of the reconstruction error from normal data was set as the threshold
- The reconstruction errors of the sequences that exceeded this threshold was considered to be anomalous in nature.

Step 6: Visualization

- The graph for Training Loss and Validation Loss is plotted in order to visually demonstrate how well the LSTM Autoencoder model has learned on its training data compared to the model's capability of generalizing unseen test data.
- Various other visualizations were also performed for the analysis of results.

3.5.2 Transformer Autoencoder:

Step 1: Defining the Model Architecture of Transformer Autoencoder

The Transformer Autoencoder consisted of the following components:

- Positional Encoding: Since, data is processed by Transformers non-sequentially, positional encoding is added to the input sequence in order to encode the order of elements.
- Encoder: The transformer encoder is a set of multi-head self-attention layers that capture the relationships between various elements in the sequence. This helps in the identification of local and global dependencies effectively.
- Decoder: Similar to LSTM Decoder, the Transformer Decoder reconstructs the original input sequence from the latent representation of data.

Step 2: Training the Model

- In order to minimize the reconstruction errors, the Transformer Model was trained only on the normal sequences of data
- The process of training the model is a very necessary step as it helps the transformer model to comprehend how the normal sequences actually look like

Step 3: Testing the Model

- During the process of testing, the trained model was now applied to the test dataset, that consisted of both normal sequences as well as anomalous sequences of data
- The Transformer Autoencoder model now reconstructed both the normal sequences and anomalous sequences, leading to reconstruction error values.

Step 4: Calculation of Reconstruction Error

The reconstruction error was calculated for every input sequence, both normal and anomalous, by calculating the difference between the actual data and the reconstructed data.

Step 5: Detection of Anomaly based on Threshold

- The 85th percentile was set as the threshold for calculation of reconstruction error.
- Similar to the LSTM Autoencoder, the reconstruction errors of the sequences that was more than this threshold was considered as anomalies.

Step 6: Plotting the Visualization

- Similar to the LSTM Autoencoder, the graph for Training Loss vs Validation Loss is plotted for assessing the capabilities of the Transformer Autoencoder model to implement unsupervised learning.
- Various other visualizations were performed to analyse the results.

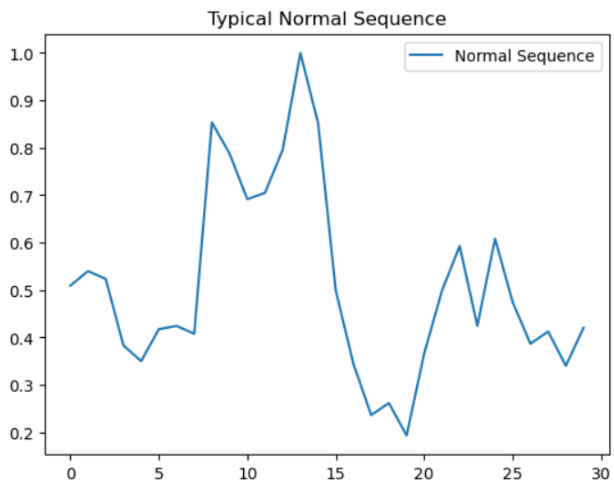
4. Results

In this project, firstly, the graphs for typical normal sequence and anomalous sequence had been plotted in order to note the key differences. The following were the results obtained:

4.1 Graphical Analysis

4.1.1 Typical Normal Sequence:

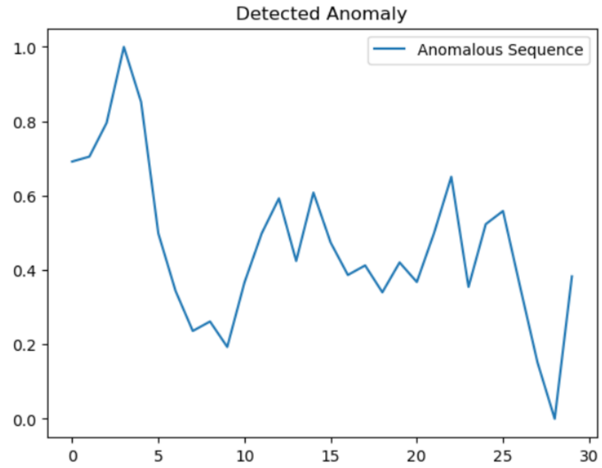
- The pattern of this graph is very smooth and balanced in nature with gradual rise and fall, without any sharp changes in it
- Throughout the sequence, the variations are moderate and consistent
- This indicates a steady heart-rate measurement during a normal physical activity



4.1.2 Anomalous Sequence:

- In this graph, we can see a sharp rise in the heart-rate followed by a steep and sudden drop. There has been a significant rise and fall in the graph in an abrupt manner

- In the end of the sequence, we can see a sudden spike, which diverted from the previous pattern
- This was indicative of either noise or measurement errors, or a potential health risk of the person

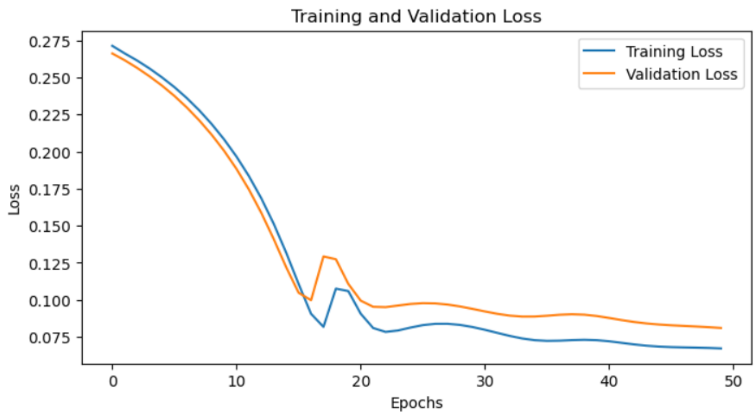


The results of the LSTM Autoencoder and the Transformer Autoencoder model are enlisted below:

4.2 Training and Validation Loss Graphs:

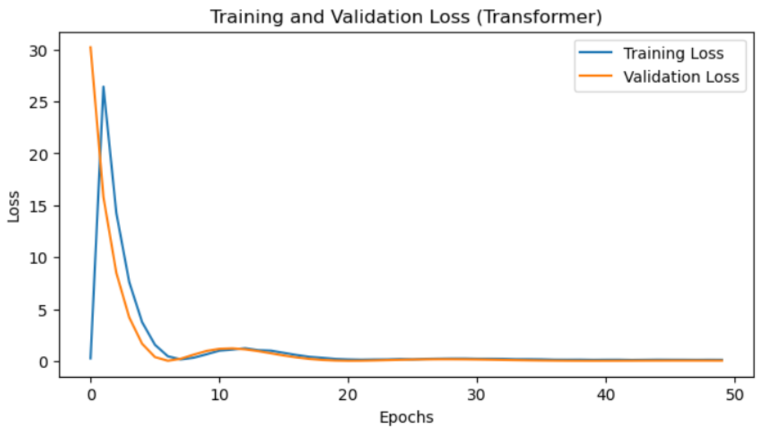
4.2.1 LSTM Autoencoder

- At first, there has been a steady decrease in the training and validation loss of the LSTM Autoencoder Model. Further, the model stabilized after 10 epochs, effectively learning to reconstruct input sequences.
- There was smooth convergence and minimal overfitting. This was indicated by the similar values of training and validation loss.



4.2.2 Transformer Autoencoder

- The training and validation loss values of Transformer Autoencoder model attained stability after 5-10 epochs, depicting fast convergence.
- The faster convergence of the Transformer model demonstrates that the model learns the patterns of data more efficiently as compared to LSTM Model.



4.3 Reconstruction Errors

4.3.1 LSTM Autoencoder

- According to the Reconstruction Error distribution of the LSTM Autoencoder model, it can be inferred from the broader range (0.06-0.09) of the model that the LSTM model was slightly less precise.
- The 85th percentile threshold (0.079) flagged 1 anomaly.

```
Reconstruction Error Threshold: 0.07998050592076347
Number of Anomalies Detected: 1
```

4.3.2 Transformer Autoencoder

- The Reconstruction Error Distribution range of the Transformer Autoencoder model was tighter (0.055-0.063) as compared to the LSTM Model, which is indicative of a more precise reconstruction of the input sequences.
- The 85th percentile threshold (0.062) identified 1 anomaly.

```
Reconstruction Error Threshold (Transformer): 0.06269691010896845
Number of Anomalies Detected (Transformer): 1
```

4.4 Analysis of the Results

Behaviour of the Anomalous Sequence:

- The anomalous sequence is identified by unusual transitions, which included sharp rise and fall, irregular oscillations, and a large variation in amplitude
- However, normal sequences generally tend to have smooth and more consistent patterns. Thus, this helps in distinguishing between normal and anomalous patterns more effectively.
- The probable reasons for this anomaly may include:
  i. A sudden physical exhaustion or a concern related to health
  ii. Errors in data collection or noise produced by the heart rate sensor of the tracking device
  iii. External factor that could affect the process of heart-rate measurement

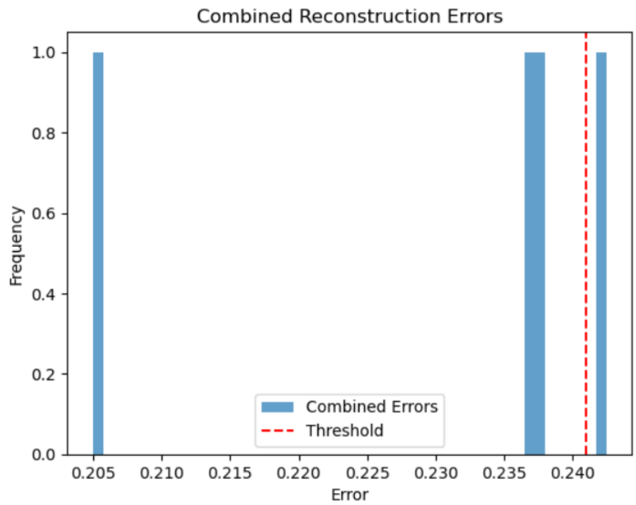
Key Observations:

- It has been observed that both the models have flagged the same data sequence as anomalous, thus, ensuring confidence in the process of detection
- Both the models showed their capabilities in unique ways:

a. The LSTM Autoencoder model detected sharp and extreme anomalies in the data sequence
b. The Transformer Autoencoder model was able to capture subtle irregularities and nuanced deviations more effectively

- The threshold of reconstruction error for the Transformer Autoencoder model (85th percentile or 0.0626) could effectively differentiate between anomalous and normal sequences due to its tighter range

Furthermore, I have combined both the models to check for anomalies in the same dataset in the most effective way possible. However, on combining both the models, I could not get fruitful results (Liu et al., 2022). Since, both the LSTM Autoencoder model and the Transformer Autoencoder model produced similar threshold values for reconstruction error, on combining them, we could not obtain additional results.



5. Conclusion

The main aim of this project was to address the crucial issue of anomaly detection in the time-series heart-rate data collected from a fitness tracker application called FitBit, leveraging advanced deep learning models in it. This project had employed the two critical algorithms for anomaly detection -LSTM Autoencoder and Transformer Autoencoder. With the help of this, the project could successfully help us comprehend the capability of machine learning to detect anomalies, both obvious and possible ones (Wang & Wan, 2019). Consequently, this could help us indicate anomalies in human physiology, measurement errors, or malfunctions of the sensor, or unusual physical activities.

5.1 Key Achievements of the Model:

- Developed a Robust Framework: This project helped us to develop a robust framework of the model, which encompassed data preprocessing, model training and evaluation. This was also followed by interactive visualizations. The preprocessing steps ensured that the data quality was maintained. Synthetic anomalies were also introduced in the dataset so as to evaluate the performance of the model in a controlled surrounding.
- Implementation of the Algorithms: The LSTM Autoencoder did well in the detection of sharp and noticeable irregularities. This proved that the LSTM

Autoencoder was capable of handling temporal dependencies in time-series data. On the other hand, the Transformer Autoencoder was able to capture both local and global patterns effectively through its mechanism of self-attention. This inferred that the Transformer Autoencoders were able to capture subtle and nuanced deviations as well.

- Model Evaluation: Based on the reconstruction errors, both the LSTM Autoencoder model and the Transformer Autoencoder model was able to distinguish between normal sequences and anomalous sequences of data. According to the visualizations, it can be inferred that, both the models had indicated the same data sequence as anomalous. This reinforced confidence in the models and contributed to their reliability and robustness.
- Real-life Implementations: The computation efficiency of LSTM Autoencoders make it suitable for implementing in real-life fitness-tracker applications and other wearable devices. Due to its ability to perform more intricate and detailed analysis and high sensitivity positions, Transformer Autoencoders can highly contribute to the healthcare monitoring systems in the real-world scenarios.

5.2 Contribution of this Project:

- This project has greatly contributed to the evolving field of anomaly detection by showing the effective application of deep learning architectures to time-series data, even when the ground truth labels are limited. Using reconstruction error in place of anomaly likelihood demonstrated how versatile unsupervised learning approaches are in real-world scenarios.
- The comparative analysis of both the models emphasizes on the trade-off between sensitivity of the models and their computational efficiency. LSTM Autoencoder suit well in real-time environments where the resources are limited (Cao, Lin, Guo, Xiong, & Jiao, 2023). On the other hand, Transformer Autoencoders are very extremely effective for in-depth pattern recognition.

5.3 Future Implications:

I would like to implement this deep learning architecture in my future works. However, there must be a few things that are to be considered in the future:

- Instead of synthesizing anomalies, if real-world anomalies are incorporated into the dataset from the diverse range of user populations, the overall generalization and robustness of the model would be improved.
- Instead of unseen data, if we collect labelled datasets containing anomalies validated from experts, we can calculate the precision and recall metrics, thus, enabling a more rigorous and informative evaluation.
- If attention mechanisms are introduced with the LSTM layers, The sensitivity of the model could be enhanced further with overhead computational costs.
- Optimization of the Transformer Autoencoder model for edge devices like routers, multiplexers, routing switches, etc. could lead to expansion of their application in the real-world scenarios.
- If we can integrate these models into the wearable technology platforms, it can reform health monitoring systems on a large scale.

Although, this project has focussed on the heart-rate data only, we can generalize the methodologies and frameworks to other forms of time-series data, such as IoT sensor data, environment monitoring systems, or real-time stock market data as well (Rassam, 2024).

5.4 Cost Analysis:

The cost analysis of this project depends on the type of work (whether academic or professional) we are performing.

- For academic or individual work, the cost of this project can be estimated between $30 to $500, provided we use open-source tools and personal hardware for the project.
- For commercial use in the real-world, using diverse datasets and commercial-standard infrastructure, the cost range can vary between $5000 to $15000. This also requires a good amount of time investment and potential expenditures for commercial hardware, datasets, and tools.

Reference List

Hawkins, D. (1980). Identification of outliers. **Available at:** https://link.springer.com/content/pdf/10.1007/978-94-015-3994-4.pdf

An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. Special lecture on IE, 2(1), 1-18. **Available at:** http://dm.snu.ac.kr/static/docs/TR/SNUDM-TR-2015-03.pdf

Nguyen, H. D., Tran, K. P., Thomassey, S., & Hamad, M. (2021). Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. International Journal of Information Management, 57, 102282. **Available at:** https://www.sciencedirect.com/science/article/pii/S026840122031481X

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff,G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148. **Available at: https://arxiv.org/abs/1607.00148**

Maleki, S., Maleki, S., & Jennings, N. R. (2021). Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering. Applied Soft Computing, 108, 107443. **Available at:** https://www.sciencedirect.com/science/article/pii/S1568494621003665

Said Elsayed, M., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2020, November). Network anomaly detection using LSTM based autoencoder. In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks (pp. 37-45). **Available at:** https://dl.acm.org/doi/abs/10.1145/3416013.3426457

Do, J. S., Kareem, A. B., & Hur, J. W. (2023). Lstm-autoencoder for vibration anomaly detection in vertical carousel storage and retrieval system (vcsrs). Sensors, 23(2), 1009. **Available at:** https://www.mdpi.com/1424-8220/23/2/1009

Liu, P., Sun, X., Han, Y., He, Z., Zhang, W., & Wu, C. (2022). Arrhythmia classification of LSTM autoencoder based on time series anomaly detection. Biomedical Signal Processing and Control, 71, 103228. **Available at:** https://www.sciencedirect.com/science/article/pii/S1746809421008259

Mitiche, I., McGrail, T., Boreham, P., Nesbitt, A., & Morison, G. (2021). Data-driven anomaly detection in high-voltage transformer bushings with lstm auto-encoder. Sensors, 21(21), 7426. **Available at:** https://link.springer.com/article/10.1186/s40064-016-2107-7

Wang, L., Zhao, X., Pei, J., & Tang, G. (2016). Transformer fault diagnosis using continuous sparse autoencoder. SpringerPlus, 5, 1-13. **Available at:**

Wang, T., & Wan, X. (2019, August). T-CVAE: Transformer-based conditioned variational autoencoder for story completion. In IJCAI (pp. 5233-5239). **Available at:** https://link.springer.com/chapter/10.1007/978-3-030-63031-7_8

Liu, D., & Liu, G. (2019, July). A transformer-based variational autoencoder for sentence generation. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE. **Available at:** https://ieeexplore.ieee.org/abstract/document/10252018/

Cao, X., Lin, H., Guo, S., Xiong, T., & Jiao, L. (2023). Transformer-based masked autoencoder with contrastive loss for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing. **Available at:**

Fan, C., Xiao, F., Zhao, Y., & Wang, J. (2018). Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. Applied energy, 211, 1123-1135. **Available at:** https://www.sciencedirect.com/science/article/pii/S0306261917317166

Rassam, M.A., 2024. Autoencoder-Based Neural Network Model for Anomaly Detection in Wireless Body Area Networks. IoT, 5(4), pp.852-870. **Available at:** https://www.mdpi.com/2624-831X/5/4/39

Sivan, K. M., Hu, S., Aslam, N., Chen, X., Sureephong, P., Wongsila, S., & Ahmed, S. Q. (2023, December). Heart rate anomaly detection using contractive autoencoder for smartwatch-based health monitoring. In 2023 15th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 118-123). IEEE. **Available at:** https://ieeexplore.ieee.org/abstract/document/10387334/

Abir, F. F., Chowdhury, M. E., Tapotee, M. I., Mushtak, A., Khandakar, A., Mahmud, S., & Hasan, A. (2023). PCovNet+: A CNN-VAE anomaly detection framework with LSTM embeddings for smartwatch-based COVID-19 detection. Engineering Applications of Artificial Intelligence, 122, 106130. **Available at:** https://www.sciencedirect.com/science/article/pii/S0952197623003147

Bajaj, R. K., Meiring, R. M., & Beltran, F. (2023). Co-Design, Development, and Evaluation of a Health Monitoring Tool Using Smartwatch Data: A Proof-of-Concept Study. Future Internet, 15(3), 111. **Available at:** https://www.mdpi.com/1999-5903/15/3/111

Websites

Deepak Birla (2019). Basics of Autoencoders - Deepak Birla - Medium. [online] Medium. **Available at:** https://medium.com/@birla.deepak26/autoencoders-76bb49ae6a8f