SOFTWARE REQUIREMENT SPECIFICATION

Project Title: Transformer for Time Series Anomaly Detection

---

1. Introduction

1.1 Purpose

The purpose of this project is to create an effective solution for detecting anomalies in time series data using Transformer models. Time series data often hides irregular patterns or rare events among normal data, making anomalies challenging to spot, especially without labelled examples. The solution will improve reliability, automate monitoring, and support better decision-making. It applies to industries like industrial monitoring, financial services, healthcare, and space exploration.

1.2 Scope

The project will analyse time series data (both single-variant and multi-variant) to identify unusual patterns that may indicate issues. The system will leverage advanced models like the Anomaly Transformer to handle data efficiently and accurately.

Key deliverables include:

- A trained model for anomaly detection:

  The project will deliver a fully trained anomaly detection model based on Transformer architecture, fine-tuned for time series data. This model will be capable of identifying irregular patterns in both univariate and multivariate datasets. Using self-attention mechanisms, it will effectively distinguish normal data points from anomalies, even in cases with subtle deviations

- User-Friendly Tool or Interface:

  The project will include a simple and intuitive tool for deploying the anomaly detection model and visualizing its results. Users can upload their time series data through a web-based interface and immediately analyse it for anomalies. The tool will provide real-time anomaly detection with timestamps, historical trend analysis, and customizable

visualizations such as graphs and heatmaps. Users can interact with the data by zooming in on specific time periods or filtering for particular anomaly types. Additionally, the tool will allow exporting results in CSV formats for reporting and further analysis.

- Documentation of Findings and Performance:

  The documentation will detail the model's design, data insights, and performance metrics. It will include an overview of the Transformer architecture, key features of the data, and evaluation metrics such as precision, recall, F1-score, and latency. Case studies of detected anomalies will highlight the model's real-world applications. The documentation will also provide a user guide for deploying the tool, troubleshooting common issues, and interpreting results effectively.

This solution is built to empower industries by enabling informed decision-making, reducing operational downtime, and boosting overall system reliability. By quickly identifying anomalies in time series data, organizations can proactively address issues before they escalate into major problems. This leads to improved efficiency, lower maintenance costs, and a more reliable operational workflow.

Applications of Time series anomaly detection using transformers

- Finance and Banking:

  Fraud Detection: Identify unusual patterns in transaction data to detect fraudulent activities.
  Stock Market Analysis: Detect anomalies in stock price movements or trading volumes.
  Risk Management: Monitor risk factors in financial markets or portfolios.

- Healthcare:

  Patient Monitoring: Detect irregularities in vital signs like heart rate, blood pressure, or glucose levels.

  Medical Equipment: Identify malfunctions or unusual usage patterns in medical devices.

  Disease Outbreak Detection: Spot anomalous trends in disease incidence or test results.

- Manufacturing:

  Predictive Maintenance: Identify early signs of equipment failure based on sensor data.

  Quality Control: Detect defects in production processes by analysing operational metrics.

  Energy Usage: Monitor energy consumption for unusual patterns.

- Energy and Utilities:

Grid Monitoring: Identify anomalies in electricity or water consumption patterns.

Renewable Energy: Detect irregularities in wind, solar, or hydroelectric power generation.

Oil and Gas: Monitor pipeline sensor data for leaks or failures.

- Transportation:

  Vehicle Monitoring: Detect anomalies in vehicle performance metrics like speed, fuel efficiency, or engine health.

  Traffic Management: Identify unusual traffic patterns or incidents.

  Fleet Management: Monitor operational metrics across a fleet of vehicles.

- Cybersecurity:

  Intrusion Detection: Spot unusual network traffic or login patterns to prevent cyber-attacks.

  Log Analysis: Identify suspicious activities in system logs.

  Endpoint Security: Detect anomalies in device usage patterns.

- Retail and E-commerce:

  Demand Forecasting: Identify sudden surges or drops in product demand.
  Supply Chain Management: Detect irregularities in inventory levels or delivery times.
  Customer Behaviour Analysis: Spot unusual purchasing patterns.

- Space and Earth Observation:

  Satellite Monitoring: Detect anomalies in satellite data for early fault detection.
  Environmental Monitoring: Spot irregular patterns in climate or weather data.
  Geophysical Events: Identify precursors to earthquakes, volcanic activity, or other natural phenomena.

- IoT and Smart Systems:

  Home Automation: Detect irregularities in smart appliance usage patterns.
  Smart Cities: Monitor city-wide systems like lighting, traffic signals, or waste management.
  Industrial IoT: Identify unusual behaviours in factory sensor networks.

- Telecommunications:

  Network Monitoring: Detect anomalies in call volumes, data usage, or network latency.
  Service Reliability: Monitor for outages or degradation in service quality.
  Fraud Prevention: Identify unusual calling patterns indicative of scams.

1.3 Definitions, Acronyms, and Abbreviations

- Transformer: A deep learning model designed for sequential data, initially used in NLP but now applied to time series.
- Anomaly Detection: Identifying patterns that don't fit the usual behaviour in data.

- Time Series: A sequence of data points arranged in time order.
- Self-Attention Mechanism: A neural network architecture component that allows a model to focus on relevant parts of the input sequence when making predictions. It enables better understanding of relationships within the data for more accurate predictions.
- Association Discrepancy: A metric used in anomaly detection to compare the association of a data point to its adjacent time points against its association to the entire time series. This helps differentiate normal and abnormal data points.
- Gaussian Kernel: A mathematical function used to give higher importance to closer data points in association modelling, facilitating the detection of local anomalies. It helps the model focus on the finer details of local patterns and anomalies by treating nearby points as more significant. Essentially, it highlights small, subtle deviations from the norm that might be missed by simply looking at distant data points.
- KL Divergence: Kullback-Leibler Divergence, a measure of how one probability distribution diverges from a second expected probability distribution. This is a way of measuring how one set of data (or probability distribution) differs from another, expected set. In the context of anomaly detection, KL Divergence helps identify how "out of place" a data point is compared to what the model expects based on the rest of the data. A high KL Divergence suggests that the point is quite unusual and may be an anomaly.
- MSE: MSE is a measure of how well a model's predictions match the actual values. It works by squaring the difference between the predicted and actual values, which helps to emphasize larger errors. The result is averaged over all data points, giving an overall idea of how accurate the model is. Lower MSE values indicate better performance, as the model's predictions are closer to the actual outcomes.

1.4 References

- Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. "**Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy**." *ICLR 2022*.

- NASA (National Aeronautics and Space Administration) & SMAP (Soil Moisture Active Passive), PSM (Pooled Server Metrics), SMD (Server Machine Dataset), SWaT (Secure Water Treatment) and MSL (Mars Science Laboratory) datasets.
- Vaswani et al., "**Attention is All You Need**," *NeurIPS 2017*.
- Solving Transformer by Hand: A Step-by-Step Math Example by “Fareed Khan”.
- Build an Artificial Neural Network from Scratch by “Nagesh Singh Chauhan”.
- Anomaly Detection in Time Series: A Comprehensive Evaluation by” Sebastian Schmidl, Phillip Wenig, Thorsten Papenbrock”.
- IS IT WORTH IT? COMPARING SIX DEEP AND CLASSICAL METHODS FOR UNSUPERVISED ANOMALY DETECTION IN TIME SERIES by “Ferdinand Rewicki , JoachimDenzler , Julia Niebling ”.
- ANOMALY DETECTION IN UNIVARIATE TIME-SERIES: A SURVEY ON THE STATE-OF-THE-ART by “MohammadBraei and Dr.-Ing. Sebastian Wagner”.

---

2. System Overview

2.1 Overall Description

The system will ingest time series data, preprocess it, and apply transformer-based models for anomaly detection. It will provide intuitive visualizations and support real-time monitoring through a deployment-ready pipeline.

- System Interfaces

  Input: CSV files are given as input.

  Output: Annotated time series data with identified anomalies, timestamps, and anomaly scores, in the same format as the input.

- User Interfaces

  File Upload: Drag-and-drop or select files for anomaly detection.
  Visualization Dashboard: View detected anomalies on interactive plots and download annotated results.

- Hardware interfaces:

  Minimum: 16 GB RAM, 4-core CPU, 1 GPU (e.g., NVIDIA Tesla T4)
  Recommended: 32 GB RAM, 8-core CPU, multiple GPUs (e.g., NVIDIA V100)

- Software Interfaces:

  Libraries:

  PyTorch for transformer model training and inference.

  Matplotlib for visualizing anomalies.
  NumPy and Pandas for data preprocessing.
  Environment: Python 3.8+.
  External Interfaces: API-based integration for real-time anomaly detection or communication with other monitoring systems.

2.2 Key Features

→Automated Data Preprocessing: Cleaning, normalizing, and handling missing data.

→Transformer Model Integration Uses attention mechanisms to detect anomalies.

→Real-time Monitoring: Processes streaming data for continuous anomaly detection.

→Interactive Dashboards: Visual representation of anomalies.

→Scalable Deployment: Ability to run on cloud or edge devices.

---

3. Operating Environment

3.1 Software Requirements

Programming Languages:

→Python: The primary programming language used for model development, data manipulation, and deployment.

Libraries:

→PyTorch: A deep learning library for building and training neural network models, including the Transformer model for anomaly detection.

→NumPy: Used for handling large arrays and matrices of data, essential for numerical computations and data preprocessing.

→ Pandas: Used for handling large arrays and matrices of data, essential for numerical computations and data preprocessing.

→Matplotlib: A library for plotting the graphs

→Scikit-learn: A library that provides simple tools for machine learning tasks, such as model evaluation and data preprocessing.

Visualization Tools:

→Plotly: A visualization library used for creating interactive and dynamic graphs that help users explore data and model results.

→Dash: A web application framework built on top of Plotly for creating interactive dashboards, which allow users to visualize and interact with anomaly detection results.

Deployment:

→Docker: A platform that packages applications and their dependencies into containers, ensuring consistent deployment across different environments.

→Kubernetes: An open-source system for automating the deployment, scaling, and management of containerized applications, used for scaling the anomaly detection solution.

Cloud:

→AWS: **(Amazon Web Services)**: Provides scalable cloud infrastructure and services, useful for hosting the model, storing data, and deploying the solution.

→Azure: Microsoft's cloud platform offering similar capabilities for hosting and scaling machine learning models.

→GCP: Google's cloud platform, which offers machine learning tools and scalable cloud infrastructure for deploying the solution.

3.2 Hardware Requirements

Minimum: 16 GB RAM, 4-core CPU, 1 GPU (e.g., NVIDIA Tesla T4)

Recommended: 32 GB RAM, 8-core CPU, multiple GPUs (e.g., NVIDIA V100)

---

4. Functional Requirements

4.1 Data Collection and Storage

Source:

→Public datasets: The project will use publicly available datasets, such as those on platforms like Kaggle, which provide rich time series data for training and testing the anomaly detection model. These datasets cover a wide range of domains like finance, healthcare, and industrial processes.

→IoT devices: Data from Internet of Things (IoT) devices can be used, especially for applications involving real-time monitoring and anomaly detection in sectors like manufacturing or environmental monitoring.

→Financial systems: Time series data from financial transactions, stock prices, or other financial metrics can serve as valuable sources for detecting anomalies in the financial domain, such as fraud or market irregularities.

Storage: Data will be stored in a cloud-based database such as AWS S3 or a local database like PostgreSQL.

4.2 Data Preprocessing

→Handle missing values: To ensure that the model can work with complete data, missing values will be addressed using techniques like interpolation. Interpolation estimates missing values based on existing data points, ensuring a smooth transition and preventing gaps that could skew the analysis. Other methods such as imputation (replacing missing values with mean, median, or mode) may also be considered depending on the dataset's characteristics.

→Normalize data: Data normalization ensures that all features are on a similar scale, which is crucial for machine learning models to function effectively.

→Prepare data for Transformer model: The data will be formatted into suitable sequences for the Transformer model. This involves converting the time series into input-output pairs, where each sequence represents a window of data points. The data will be split into training, validation, and test sets,

4.3 Model Development

The anomaly detection model will be built using advanced Transformer architectures such as the Time Series Transformer. This architecture is specifically designed to handle sequential data and capture long-term dependencies, making it ideal for time series analysis. The model will be fine-tuned using transfer learning. The model's effectiveness will be evaluated using various metrics, including precision, recall, and F1-score, to ensure it accurately detects anomalies while minimizing false positives and negatives.

4.4 Visualization

The system will provide real-time anomaly detection, displaying anomalies with precise timestamps to help users understand when irregularities occur. An interactive dashboard will allow users to explore historical data, view detected anomalies over time, and gain insights into trends and patterns. Users will be able to filter data, zoom into specific time periods, and visualize anomalies in an easily digestible format.

4.5 Deployment

To make the model accessible for production use, the model will be deployed using Express.js, a lightweight web framework for Node.js. This will allow for easy integration with web applications, enabling users to interact with the anomaly detection model through RESTful APIs. Express.js will handle incoming requests for anomaly detection, process the data, and return the results in real time.

---

5. Non-Functional Requirements

5.1 Performance

The system will be optimized to process at least 1000 data points per second while maintaining a latency of under 200 ms. This ensures that real-time anomaly detection can be performed swiftly, enabling quick responses to irregularities without causing delays.

5.2 Security

→ SSL/TLS Encryption for Data in Transit: To ensure the security of data as it travels over the internet, the system will use SSL/TLS encryption. This encryption protocol protects the integrity and confidentiality of data by securing the connection between the client and the server, making it difficult for unauthorized parties to intercept or alter the data. This is crucial for safeguarding sensitive information, particularly in applications like financial systems or healthcare data.

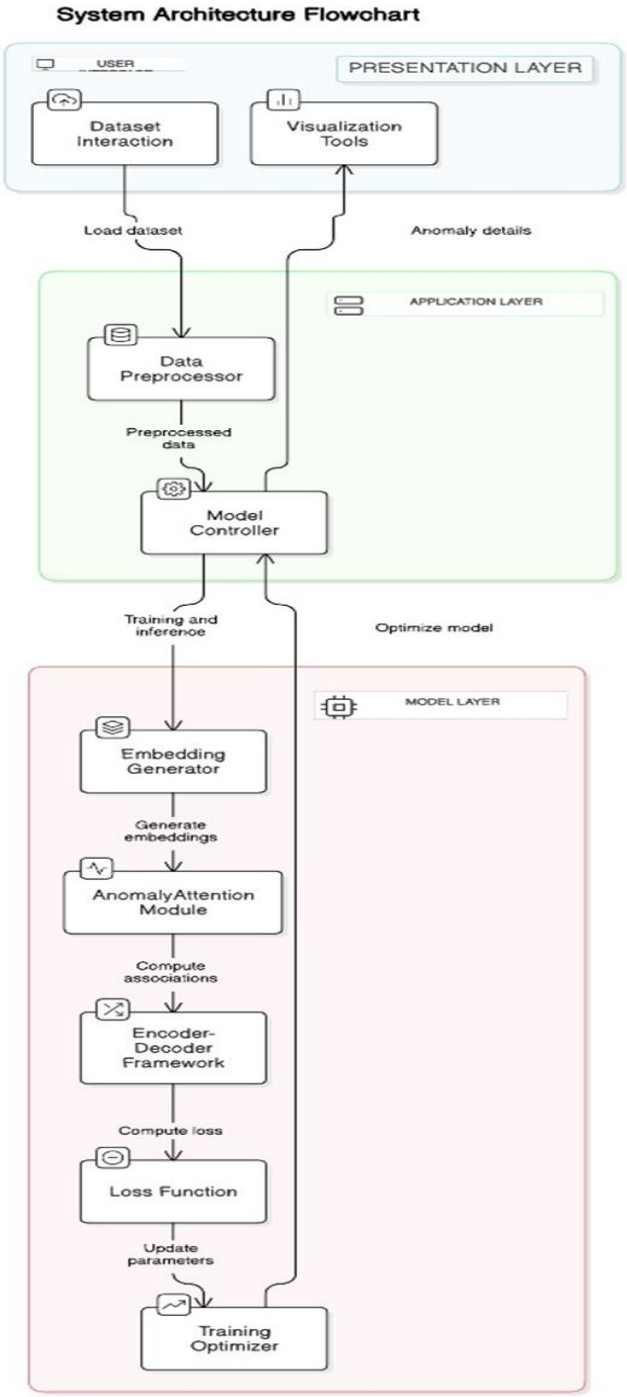
5.3 Scalability

The system will be designed to handle datasets with millions of data points efficiently. To achieve this, it will incorporate techniques such as distributed processing and optimized storage solutions, ensuring that even large volumes of time series data can be processed in a timely manner. The model will be able to scale horizontally, meaning it can handle increasing data sizes without compromising performance.

---

6. System Design

System Architecture



The system architecture for time series anomaly detection using transformers consists of three layers:

- Presentation Layer: Provides user interaction through dataset uploads and visualization tools to display anomaly details.
- Application Layer: Handles data preprocessing and manages the model controller for training and inference tasks.
- Model Layer: Includes embedding generation, anomaly attention mechanisms, encoder-frameworks, and optimization processes for anomaly detection.
  This design ensures seamless anomaly detection from raw data to detailed insights.