1. Proposed Solution

The proposed solution is a modular, ensemble-based framework for multivariate time series anomaly detection and feature attribution. The system is designed to process time-series data, train on a designated "normal" period, and subsequently score all data points for anomalous behaviour.

The core of the framework is an intelligent ensemble of three distinct models, each selected for its specialization in detecting a specific class of anomaly:

1. **Local Outlier Factor (LOF)**: Targets density-based outliers, effective for Threshold Violations.
2. **Minimum Covariance Determinant (MCD)**: Targets deviations in the data's correlational structure, effective for Relationship Changes.
3. **Variational Autoencoder (VAE)**: Targets deviations from learned non-linear patterns, effective for Pattern Deviations.

The final Abnormality Score for each timestamp is determined by the maximum score produced by any of the three models. This ensures maximum sensitivity to any single type of anomaly without dilution. Feature attribution is then derived from the specific model that produced the winning score for that timestamp.

2. Technical Framework

The technical framework of each model is given below

2.1. Local Outlier Factor (LOF) for Density-Based Detection

LOF quantifies the anomalous nature of a data point by comparing its local density to the local densities of its neighbours. A point is considered an outlier if it resides in a region of significantly lower density than its neighbours.

Mechanism: The LOF score is a ratio based on the Local Reachability Density (LRD). The score for a point A is calculated as the average LRD of its neighbours divided by its own LRD.

$$LOF_k(A) = \frac{\sum_{B \in N_k(A)} \frac{LRD_k(B)}{LRD_k(A)}}{|N_k(A)|}$$

Where LRD_k is local reachability density, it's the score indicating how tightly packed a data point is with its closest neighbours.

A score $LOF_k(A) >> 1$ indicates that the point A is in a sparser region than its neighbours, making it a clear density-based outlier. This is highly effective for detecting threshold violations that push a data point far from any cluster of normal operations. **Hence it detects localized deviations that global models (e.g., Isolation Forest) may miss.**

2.2. Minimum Covariance Determinant (MCD) for Relational Anomaly Detection

MCD provides a robust estimate of the data's covariance, making it resilient to outliers during training. It identifies the "center" of the normal data and measures how far each point is from this center using the Mahalanobis distance.

Mechanism: The anomaly score is the Mahalanobis Distance (D_M), which measures the distance from a point x to a distribution's center, accounting for inter-variable correlations.

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}$$

Here, μ and S are the robust mean and covariance matrix estimated by the MCD algorithm.

Unlike Euclidean distance, D_M, is sensitive to the correlational structure of the data (captured in S^{-1}). A high D_M, score indicates that a data point violates the learned relationships between variables, directly addressing the detection of Relationship Changes. **Therefore, MCD is resistant to outliers during training, unlike PCA, and highlights breakdowns in inter-variable correlations.**

2.3. Variational Autoencoder (VAE) for Pattern Anomaly Detection

The VAE is a generative deep learning model that learns a probabilistic latent representation of the normal training data. Anomalies are identified as data points that the model fails to reconstruct accurately from this learned representation.

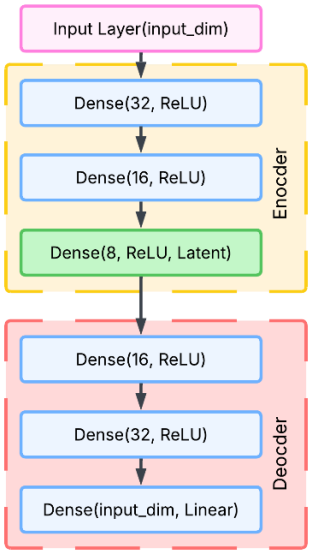

Figure 1: VAE Architecture Diagram

Mechanism: The VAE is trained to minimize a loss function composed of two terms: the Reconstruction Error and the Kullback-Leibler (KL) Divergence.

$$Loss = E[\log P(x|\hat{x})] - D_{KL}\big(Q(z|x) \parallel P(z)\big)$$

The anomaly score is derived from the reconstruction error term, typically the Mean Squared Error between the input x and the reconstructed output $\hat{x}$.

The model learns the complex, non-linear "rules" of normal system behaviour. Data points that deviate from these temporal or feature-based patterns cannot be successfully passed through the VAE's encoding-decoding process, resulting in a high reconstruction error. This makes it exceptionally powerful for detecting subtle Pattern Deviations. **Hence VAE establish a structured and continuous latent space that supports generative capabilities, allowing the synthesis of new data samples.**

3. Implementation & Technologies

The solution is implemented as an automated pipeline in Python 3.9+.

1. **Core Libraries**: Pandas, NumPy, Scikit-learn
2. **Machine Learning Frameworks**: Scikit-learn for the LocalOutlierFactor and EllipticEnvelope (MCD) implementations, TensorFlow/Keras to construct and train the VAE model.
3. **Workflow**: The pipeline ingests the raw CSV, separates the training period, fits the scaler and all three models on this "normal" data, and then applies them to the full dataset to generate the final scores and feature attributions before exporting the augmented CSV.

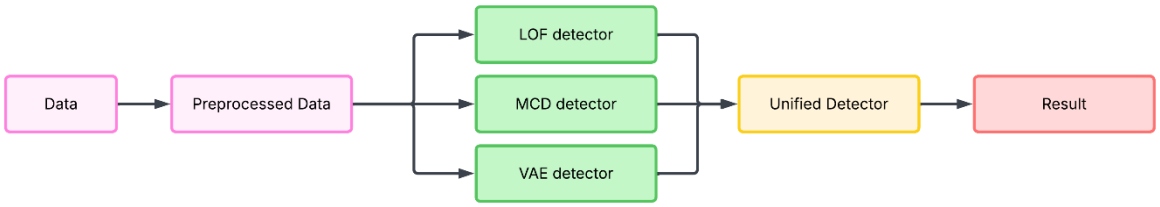

Figure 2: Data Pipeline

Methodology: To run this solution, click on this link https://github.com/haiderameez/TSD-Anomaly-Detection**. The repository contains a readme file that explains how to run the solution.**

4. Feasibility and Viability

The solution is highly feasible. It's built entirely on standard Python libraries (Pandas, Scikit-learn, TensorFlow) and proven ML models. The modular design and modest computational needs for the target dataset size confirm its immediate viability.

4.1. Potential Challenges & Risks

Scalability: In-memory processing limits performance on datasets that exceed system RAM.

Hyperparameter Optimization: Static model hyperparameters risk suboptimal detection accuracy across diverse datasets. This is because each dataset might have its own set of **unique hyperparameters**.

Concept Drift: Model accuracy will degrade over time if the statistical properties of "normal" operational data evolve.

4.2. Strategies for Overcoming Challenges

1. For **Scalability**, **Integrate Dask** to enable out-of-core computation, allowing the pipeline to process datasets larger than RAM by operating on parallelized data chunks.

2. To fix the issue of **hyperparameter optimization**, we can implement an **automated tuning module** using Bayesian Optimization to efficiently find optimal model parameters and maximize detection accuracy.

3. Reducing **context drift** can be achieved by establishing a monitoring pipeline that uses the **Kolmogorov-Smirnov (K-S) test** to detect data drift. A significant drift will automatically trigger model retraining on new data to ensure long-term accuracy.

5. Research and References:

1. LOF: Understanding Local Outlier Factor (LOF) for Anomaly Detection: A Comprehensive Guide - Medium (May 2024)

2. MCD:

Minimum covariance determinant and extensions - Wiley Computational Statistics (2018)

Anomaly detection using Minimum Covariance Determinant (MCD) by Dmitrii Stepanov - Medium (July 2022)

Outliers detection with the minimum covariance determinant estimator in practice - ScienceDirect (2009)

3. VAE:

Variational Autoencoder for Anomaly Detection: A Comparative Study - arXiv (August 2024)

Multivariate time series anomaly detection with variational autoencoder and spatial–temporal graph network - ScienceDirect (April 2024)

VELC: A New Variational AutoEncoder Based Model for Time Series Anomaly Detection - arXiv (2020)

4. Ensemble Methods: Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach - PMLR (2018)