A Comparative Study of Anomaly Detection Techniques on Multivariate Server Metrics

Sreevarshini Srinivasan
University of Maryland, College Park

July 2025

Abstract

Anomaly detection is the process of identifying unexpected observations that deviate from normal patterns. This study compares the performance of three unsupervised learning algorithms—Isolation Forest, One-Class SVM, and Autoencoder—on the Pooled Server Metrics (PSM) dataset, which contains multivariate time-series data from production servers. Through exploratory analysis, modeling, and evaluation using AUC and F1 scores, we demonstrate the trade-offs between detection precision and recall in real-world anomaly detection scenarios.

1 Introduction

Anomalies, or outliers, are observations that deviate from expected behavior. In server performance data, anomalies may signal hardware failure, software crashes, or unexpected loads. Accurate detection is critical for minimizing downtime and optimizing operations.

This report explores three commonly used unsupervised algorithms for detecting anomalies in high-dimensional, real-world datasets.

2 Dataset Description

The dataset used is the Pooled Server Metrics (PSM) dataset from Kaggle, which contains over 25 numerical metrics collected at 1-minute intervals from production eBay servers.

- **psm_train.csv**: Training set with normal server behavior
- **psm_test.csv**: Test set with potential anomalies
- **psm_test_label.csv**: Ground truth labels (1 = anomaly, 0 = normal)

3 Preprocessing and Exploratory Data Analysis

The preprocessing steps included:

- Timestamp conversion to datetime format
- Numeric conversion and scaling using `StandardScaler`
- Duplicate removal

Exploratory analysis involved:

- Summary statistics (mean, median, standard deviation)
- Correlation matrix to identify multicollinearity
- Time-series plots to visualize feature behavior
- PCA to inspect variance structure and clusters

4 Models Evaluated

This study evaluates three unsupervised anomaly detection algorithms, each representing a distinct modeling paradigm: tree-based methods, kernel methods, and deep learning. All models were trained using only the normal data available in the training set and were then applied to the test data. The goal was to detect instances that deviate from typical patterns without relying on labeled anomalies during training.

4.1 4.1 Isolation Forest

Isolation Forest is a tree-based ensemble method specifically designed for anomaly detection. It operates on the principle that anomalies are few and different, and therefore more susceptible to isolation in random partitions. Each tree is constructed by randomly selecting a feature and a split value. Anomalous points are likely to be isolated earlier in the tree, resulting in shorter average path lengths across the ensemble.

This model is well-suited for high-dimensional data and is computationally efficient, making it a strong baseline for many real-world anomaly detection tasks. In our implementation, we used 100 estimators with a contamination rate of 10%, indicating an expected anomaly proportion in the test data.

Key Parameters:

- `n_estimators` = 100
- `contamination` = 0.1

4.2 4.2 One-Class SVM

One-Class Support Vector Machine (SVM) is a kernel-based method that attempts to learn the boundary of a high-density region that encloses the normal data. Points that lie outside this boundary are considered anomalies. It relies on a transformation of the input space using kernels, enabling the detection of nonlinear patterns.

This model is sensitive to its hyperparameters: `nu` (an upper bound on the fraction of anomalies) and `gamma` (kernel coefficient). Although computationally more expensive than Isolation Forest, it can capture more complex boundaries in the feature space.

Key Parameters:

- `kernel` = "rbf"
- `nu` = 0.1
- `gamma` = 0.001

4.3 4.3 Autoencoder

An Autoencoder is a type of feedforward neural network that is trained to reconstruct its input. It compresses the data into a lower-dimensional representation (encoder) and then attempts to reconstruct the original input (decoder). The reconstruction error (typically mean squared error) is used as the anomaly score: higher errors indicate potential anomalies.

In this study, we trained the autoencoder only on normal data. During inference on test data, instances with reconstruction error above a fixed threshold were labeled as anomalies. The model architecture included two hidden layers in both the encoder and decoder, with ReLU activation and sigmoid output.

Architecture:

- Input layer: 25 units (one per feature)
- Encoder: $16 \rightarrow 8$ units
- Decoder: $16 \rightarrow 25$ units
- Activation functions: ReLU (hidden), Sigmoid (output)

Training Details:

- Optimizer: Adam
- Loss: Mean Squared Error (MSE)
- Epochs: 10
- Batch size: 64
- Validation split: 0.1

Anomaly Threshold: We set the anomaly threshold at the 95th percentile of the reconstruction error on the training data, assuming anomalies are rare and should lie in the tail of the error distribution.

Each model was evaluated based on how well it could detect anomalies in the test data using the available ground truth labels. Performance trade-offs are discussed in Section 6.

5 Evaluation Metrics

We used the following metrics to evaluate model performance:

- **Precision**: The proportion of predicted anomalies that were correct
- **Recall**: The proportion of actual anomalies that were detected
- **F1-Score**: Harmonic mean of precision and recall
- **AUC-ROC**: Area under the ROC curve

6 Results

Table 1: Performance Summary of Models

Model	**AUC**	**Precision (1)**	**Recall (1)**	**F1 (1)**
Isolation Forest	0.2892	0.58	0.24	0.34
One-Class SVM	0.3843	0.39	0.43	**0.41**
Autoencoder	**0.6471**	**0.70**	0.13	0.21

The Autoencoder achieved the best AUC score, indicating strong ability to rank anomalies. However, One-Class SVM yielded the most balanced performance in detecting anomalies.

7 Conclusion

While the Autoencoder showed the highest AUC, its recall was low, making it suitable for high-precision use cases. One-Class SVM offered the best balance between precision and recall. Isolation Forest was conservative in anomaly detection.

8 Future Work

Future directions include:

- Threshold optimization for the Autoencoder

- Exploring temporal models like LSTM Autoencoders
- Deployment in real-time anomaly alerting systems using Streamlit or FastAPI

References

- PSM Dataset: `https://www.kaggle.com/datasets/ljolm08/pooled-server-metrics-psm`
- Scikit-learn Documentation: `https://scikit-learn.org/`
- TensorFlow Autoencoder Guide: `https://www.tensorflow.org/tutorials/generative/autoencoder`