World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(REVIEW ARTICLE)

Check for updates

Uncovering time series anomaly using deep learning technique

P. Chiranjeevi, Yadavalli Ramya, Chinthala Balaji *, Bathini Shashank and Abbdi Sainath Reddy

Department of CSE (Data Science), ACE Engineering College, Hyderabad, Telangana, India.

World Journal of Advanced Research and Reviews, 2024, 22(01), 879–887

Publication history: Received on 03 March 2024; revised on 11 April 2024; accepted on 13 April 2024

Article DOI: https://doi.org/10.30574/wjarr.2024.22.1.1129

Abstract

Time series data, characterized by its sequential and temporal nature, plays a crucial role in various domains such as finance, healthcare, and industrial processes. Identifying anomalies within time series data is a critical task with applications ranging from fault detection to fraud prevention. Traditional anomaly detection techniques often struggle to capture complex temporal patterns and dependencies in time series data. This study presents a novel time series anomaly detection method using long short-term memory (LSTM) neural networks. LSTMs are a type of recurrent neural networks (RNNs) designed to model long-term dependencies, making them suitable for capturing the complex temporal relationships present in time series data. Here we use a financial dataset featuring opening and closing time, we predict the volume of money and the current price of that money at a specific time. The results demonstrate the efficacy of the deep learning-based approach in detecting anomalies within time series data, outperforming traditional methods in terms of sensitivity and adaptability to complex temporal patterns. The proposed methodology presents a promising solution for real-world applications where early detection of anomalies is crucial for proactive decision-making and system integrity.

Keywords: Classification; Machine Learning; Neural networks; LSTM; Decision-making; System integrity

1. Introduction

Time series data, characterized by its sequential and temporal nature, is ubiquitous across diverse domains, including finance, healthcare, industrial processes, and beyond. Extracting valuable insights from such data is paramount for decision-making, yet the presence of anomalies or irregularities poses a significant challenge. Anomaly detection in time series data is crucial for identifying deviations from normal patterns and enabling timely responses to potential issues, such as system failures, fraud, or health abnormalities.

Traditional methods of time series anomaly detection often rely on statistical metrics and rule-based approaches. However, these methods may struggle to capture complex temporal dependencies and patterns inherent in real-world data. In recent years, deep learning techniques, and specifically Long Short-Term Memory networks (LSTMs), have emerged as powerful tools for modelling and analyzing sequential data, offering a promising avenue for address the complexity of time series anomaly detection.

LSTMs, a type of recurrent neural network (RNN), are well-suited for tasks that involve learning and predicting sequences. Unlike traditional feedforward neural networks, LSTMs are designed to capture long-range dependencies in sequential data, making them particularly effective in handling time series with intricate temporal structures. The ability of LSTMs to retain information over extended time intervals positions them as a robust choice for modelling the dynamic patterns present in time series datasets.

* Corresponding author: Chinthala Balaji

The primary objective of this research is to explore the application of LSTM-based deep learning techniques for time series anomaly detection. By leveraging the inherent capabilities of LSTMs to recognize and learn temporal dependencies, we aim to develop a robust anomaly detection system capable of accurately identifying deviations from normal behaviour in time series data.

This study encompasses the design, implementation, and evaluation of LSTM-based models for anomaly detection, considering diverse datasets from various domains. The focus is not only on the technical aspects of model architecture but also on addressing practical challenges such as the scarcity of labelled anomaly data, interpretability of results, and scalability to real-world applications.

Through a comprehensive exploration of LSTM-based time series anomaly detection, we aim to contribute to the growing body of research in the intersection of deep learning and temporal data analysis. The outcomes of this study hold the potential to enhance our ability to proactively detect and mitigate anomalies in time series data, thereby improving the reliability and robustness of systems across different domains.

2. Related Work

A substantial amount of research on outlier detection has already been conducted in statistics in the previous century (Rousseeuw and Leroy, 1987; Barnett and Lewis, 1994; Hawkins, 1980; Beckman and Cook, 1983). An extensive review of novelty detection techniques using deep learning framework and advanced statistical methodology has been presented in Markou and Singh (2003a, 2003b). A plethora of published research literature has been conducted by applying anomaly detection techniques in various applications in different domains. Anomaly detection has emerged as the topic of focus for many surveys and review papers in recent years. An extensive survey of anomaly detection techniques developed in machine learning and statistics has been provided by (Hodge and Austin, 2004; Nguyen and Armitage,

2008,These surveys provide extensive background on outliers or anomalies and the challenges associated with detecting them, thereby marked a significant impact on further research in various fields. In 2003, Noble and Cook detected unusual patterns within graph-based data using the concept of 'conditional entropy' (Noble and Cook, 2003). In 2009, Chandola et al. also surveyed different aspects of anomaly detection techniques that have been proposed in research but not covered in the literature of Hodge and Austin, providing more meaningful insight into the real-world applications they are employed in (Chandola et al., 2009).

Other exhaustive survey papers exist that focus more specifically on the techniques and applications of anomaly detection in several domains includes (Patcha and Park, 2007; Garcia-Teodoro, 2009; Callado et al., 2009; Zhang et al., 2009; Sperotto et al., 2010). Patcha and Park (2007) presented surveys of anomaly detection techniques used specifically for cyber intrusion detection. Callado et al. (2009) reported major techniques and problems identified in IP traffic analysis, with an emphasis on application detection. Zhang et al. (2009) presented a survey on anomaly detection methods in network intrusion detection.

A review of flow-based intrusion detection was presented in Sperotto et al. (2010), who elaborates on the concepts of flow and become integral to sentiment analysis. By representing words as vectors, these embeddings capture semantic relationships and contextual nA thorough comparative analysis is crucial for evaluating the strengths and weaknesses of existing sentiment analysis models. Comparative studies often consider factors such as accuracy, efficiency, and adaptability across domains. Deep learning models are pitted against each other in various scenarios to determine their performance metrics. Comparative analyses assess the accuracy and precision of models in sentiment classification. Deep learning models, with their ability to

capture intricate patterns, often showcase superior accuracy, especially in contexts with nuanced sentiment expressions. Classified attacks and provided a detailed discussion of detection techniques for DoS attacks. Other literature surveys have been reported in the context of wireless networks (Sun et al., 2006; Sun et al., 2007a, Sun et al., 2007b; Sun et al., 2007c). Sun et al. (2007a) presented a survey of intrusion detection techniques for mobile ad-hoc networks (MANET) and wireless sensor networks (WSN). Their analysis also highlighted the several important research issues and challenges in the context of building IDSs by integrating aspects of mobility. A systematic literature review of different graph-based anomaly detection techniques that have been studied in published literature (Pourhabibi et al., 2020).

3. Existing System

In the existing system, Anomaly Detection Using a Sliding Window Technique and Data Imputation with Machine Learning for Hydrological Time Series. Here he used the sliding window technique which is also a powerful technique in detecting real time anomalies, but it has a limited range and that is its drawback. Mostly based on complexity of data in choosing LSTM and Sliding Window technique. Deep learning for anomaly detection in multivariate time series. See here he used Multivariate time series data for anomaly detection but here are its drawbacks High Dimensionality, Complex Patterns, Lack of Labelled Data.

Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series Mainly they consists of generator and discriminator. The major drawbacks are Training Instability, Mode Collapse, Lack of Interpretability

4. Proposed System

Basically, the first primal Anomaly detection methods was just based on careful observations, records, and Statistics Now moving out in the Modern civilization, technology has advanced , computer is very well known , many algorithms have been developed, there are literally so many algorithms for anomaly detection But here

we are using the LSTM [LONG-SHORT-TERM-MEMORY],as we've already discussed about LSTM.

The Motive here is to minimize reconstruction error based on a loss function

5. Architecture of the System

A system architecture diagram is a visual representation of the highlevel structure of a system, illustrating how various components and modules interact with each other to achieve the system's goals

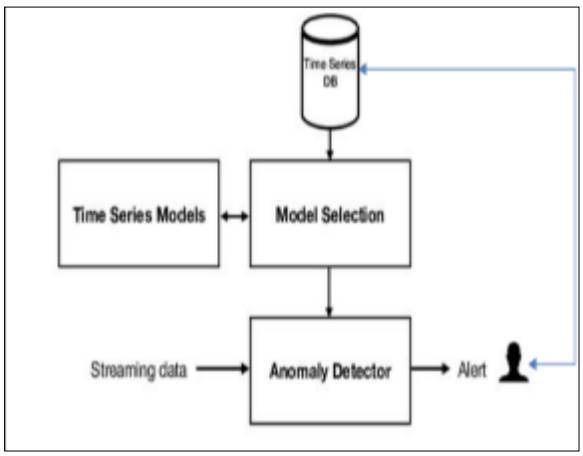


Figure 5 System Architecture

User: In time series anomaly detection, users play a crucial role in various stages of the process from the data preparation to model evaluation.

Time Series Database: A time series database is a type of database that is optimized for handling and storing time-stamped data, making it particularly well-suited for managing and querying time series data

Model Selection: Model selection in time series anomaly detection refers to the process of choosing an appropriate algorithm or model to detect anomalies in time series data.

Time Series Models: Time series models in time series anomaly detection are mathematical or statistical representations designed to capture the patterns and underlying structures in time-ordered data.

Anomaly detector: An anomaly detector using LSTM (Long Short-Term Memory) is a model designed to identify unusual patterns or outliers in time series data using thecapabilities of LSTM neural networks. LSTMs are a type of

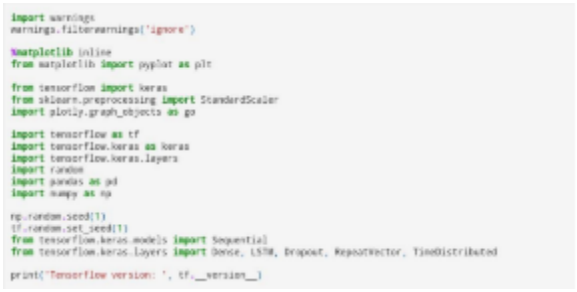
recurrent neural network (RNN) known for their ability to capture long-term dependencies and patterns in sequential data.

Output: The results of the data analysis process are then outputted.

Visualization: The output is visualized for better understanding and interpretation

6. Methodology

Description of the Machine Learning Algorithms and Techniques Chosen: Libraries imported are:



```
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
from matplotlib import pyplot as plt

from tensorflow import keras
from sklearn.preprocessing import StandardScaler
import plotly.graph_objects as go

import tensorflow as tf
import tensorflow.keras as keras
import tensorflow.keras.layers
import random
import pandas as pd
import numpy as np

np.random.seed(1)
tf.random.set_seed(1)
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout, RepeatVector, TimeDistributed

print('Tensorflow version: ', tf.__version__)
```

Figure 6 Libraries imported

6.1. Data Pre-processing

Data pre-processing is a fundamental step in preparing raw data for analysis or machine learning. It involves tasks like handling missing data, dealing with outliers, normalizing and scaling features, encoding categorical variables, and performing feature engineering. By addressing these issues, pre-processing ensures that the data is clean, consistent, and suitable for modeling. It enhances the performance of machine learning algorithms by improving their ability to identify patterns and relationships within the data, ultimately leading to more accurate and reliable results.

Furthermore, data pre-processing is essential for mitigating challenges like imbalanced datasets, inconsistent formatting, and the curse of dimensionality. Techniques such as oversampling or under sampling address imbalances, while standardization of data formats and reducing dimensionality contribute to better Uncovering Time Series Anomaly Using Deep Learning Technique 9 CSE(DATA SCIENCE) computational efficiency and model generalization. In summary, data pre-processing is a critical phase that transforms raw data into a form conducive to effective analysis and modeling, laying the foundation for robust and meaningful insights from the data.

6.2. Exploratory Data Analysis (EDA)

The initial phase involves setting time stamps in the datasets and performing exploratory data analysis (EDA) using visualizations and statistical analyses. Key patterns, trends, and anomalies are identified.

Autoencoders are an unsupervised learning technique, although they are trained using supervised learning methods. The goal is to minimize reconstruction error based on a loss function, such as the mean squared error. The steps we will follow to detect anomalies in dataset is using an LSTM autoencoder:

Train an LSTM autoencoder on the stock price data from 1985–09–04 to 2013–09–03

We assume that there were no anomalies and they were normal. Using the LSTM autoencoder to reconstruct the error on the test data from 2013–09–04 to 2020–09–03. • If the reconstruction error for the test data is above the threshold, we label the data point as an anomaly

6.3. Visualize The Time Series

Time series visualization and analytics let you visualize time series data and spot trends to track change over time. Time series data can be queried and graphed in line graphs, gauges, tables and more. Using time series visualization and analytics, you can generate forecasts and make sense of your data.

Time series visualization and analytics let you visualize time series data and spot trends to track change over time. Time series data can be queried and graphed in line graphs, gauges, tables and more.

Using time series visualization and analytics, you can generate forecasts and make sense of your data. Time series data provides significant value to organizations because it enables them to analyze important real-time and historical metrics.

Figure 7 Visualization

6.4. Preprocessing

Preprocessing is a critical step in preparing time series data for anomaly detection using Long Short-Term Memory (LSTM) networks. The goal is to transform the raw time series into a format that can be effectively used for training an LSTM autoencoder. Here are the key preprocessing steps

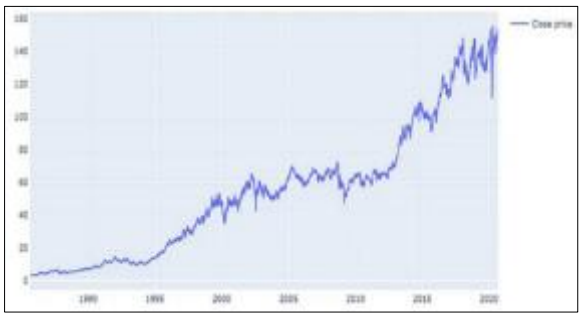


Figure 8 Visualization of Times series

6.4.1. Sequence Creation

Organize the time series data into sequences of fixed length. Each sequence represents a window of past observations. The choice of sequence length depends on the characteristics of the data and the desired trade-off between capturing short-term and long-term dependencies.

6.4.2. Normalization

Normalize the data to ensure consistent scales across features. LSTM networks are sensitive to the scale of input data. Common normalization techniques include Min-Max scaling or z-score normalization.

6.4.3. Train-Test Split

Divide the data into training and testing sets. The training set is used to train the LSTM autoencoder, while the testing set is reserved for evaluating the model's performance. Ensure that the testing set includes periods with and without anomalies to effectively evaluate the model's ability to detect anomalies.

6.4.4. Feature Selection

If the time series data includes multiple variables, decide which variables to include in the model. Choose variables that are relevant to the anomaly detection task. It may be beneficial to perform feature engineering, creating lag features, moving averages, or other transformations that capture temporal patterns.

6.4.5. Verify Data Integrity

Double-check that the preprocessing steps do not introduce errors or distort the underling structure of the time series data

6.5. Build The Model

We define the reconstruction LSTM Auto encoder architecture that expects input sequences with 30 time steps and one feature and outputs a sequence with 30 time steps and one feature.

- Repeat Vector() repeats the inputs 30 times.
- Set return_sequences=True, so the output will still be a sequence.
- TimeDistributed(Dense(X_train.shape[2])) is added at the end to get the output, where X_train.shape[2] is the number of features in the input data

6.6. Training The Model

Training an LSTM (Long Short-Term Memory) model for time series anomaly detection involves optimizing the model parameters to minimize the difference between the actual input sequences and their reconstructions. The goal is to train the model to capture the normal patterns in the time series data and identify anomalies based on deviations from these patterns.

Training a Long Short-Term Memory (LSTM) model involves optimizing the model's parameters to minimize the difference between its predictions and the actual target values

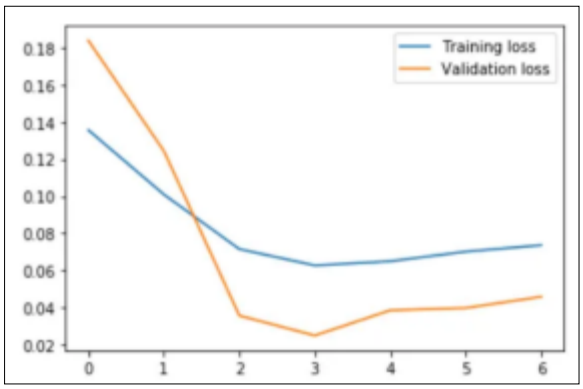


Figure 9 Training the Model

6.7. Determine Anomalies

- Find MAE loss on the training data.
- Make the max MAE loss value in the training data as the [reconstruction error threshold].
- If the reconstruction loss for a data point in the test set is greater than this [reconstruction error threshold] value then we will label this data point as an anomaly.

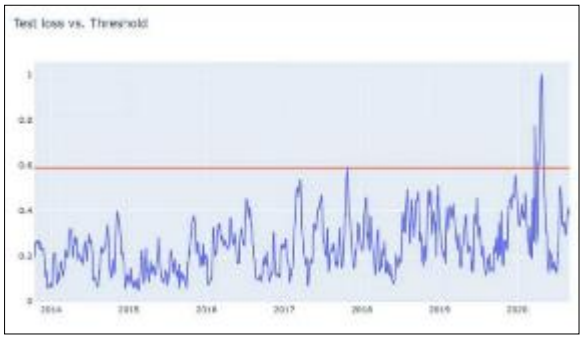


Figure 10 How Threshold is shown

6.8. Displaying Detected Anomalies

After training an LSTM model for time series anomaly detection, you can use the trained model to detect anomalies in new or unseen data. The basic idea is to compare the model's predictions (reconstructions) with the actual input data and identify instances where the reconstruction error is unusually high.

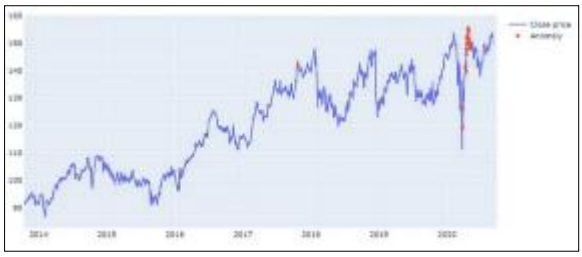


Figure 11 Anomalies Detected

7. Results and discussion

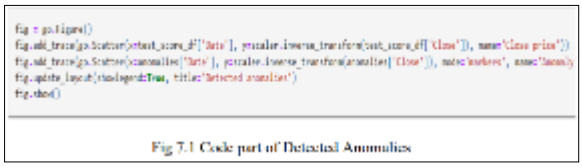


Fig 7.1 Code part of Detected Anomalies

Figure 12 Code part of Detected Anomalies

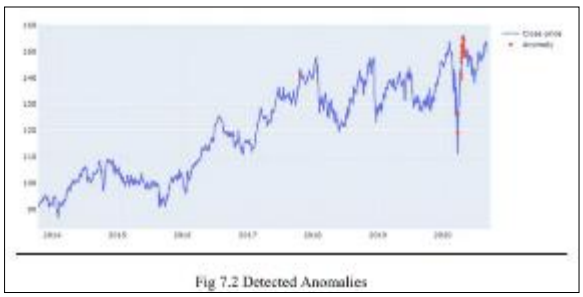


Fig 7.2 Detected Anomalies

Figure 13 Detected Anomalies

The code starts by creating a figure object.

The figure object is used to create the scatter plot, which is then added to the figure. A Scatter object is created and passed in as an argument for the add_trace function of the Figure class. This function takes two arguments: x and y, which are coordinates on the graph where data will be plotted. The mode parameter specifies how data should be plotted on this graph; it can either be markers or points depending on what you want to see.

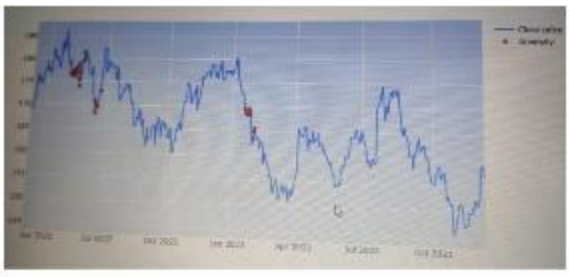


Fig 7.3 Anomaly Visualization

Figure 14 Anomaly Visualization

Finally, update_layout sets whether or not there should be a legend at the bottom of your graph (showlegend=True) and title that appears above it (title='Detected anomalies').The code then creates another Scatter object with different parameters than before: name='Close price' instead of 'Anomaly', mode='markers', showlegend=False, and no title ('Detected anomalies').The code will generate a scatter plot of the Close price versus Date. • The figure will also display an anomaly in the form of a marker.

8. Conclusion

In conclusion, time series anomaly detection using LSTM involves a multi-step process, from data preprocessing to model training and anomaly identification. It requires a balance of domain knowledge, experimentation, and careful consideration of model parameters. The adaptability of LSTM networks makes them powerful tools for capturing intricate temporal patterns in time series data. As technology advances, further research and innovations in deep learning approaches to anomaly detection will continue to enhance the effectiveness of these models in real-world applications.

LSTMs are well-suited for sequential data due to their ability to capture long-term dependencies. They maintain a memory of previous inputs, which is crucial for understanding temporal patterns in time series data. The architecture of the LSTM model plays a crucial role. It typically involves an encoder-decoder structure or a simple LSTM layer followed by a dense layer for reconstruction. LSTMs can automatically learn meaningful representations of the input time series data during the training process. The hidden states of the LSTM encode important features and patterns.

Anomaly detection with LSTMs often involves training the model on normal (nonanomalous) time series data. The model learns to reconstruct normal patterns during training. Anomaly detection is often based on the reconstruction loss, which measures the difference between the input time series and its reconstructed version by the LSTM. Higher reconstruction loss indicates potential anomalies. Establishing an appropriate threshold for the reconstruction loss is critical. Techniques such as setting a fixed threshold, using statistical measures (e.g., mean and standard deviation), or employing more advanced methods like the Isolation Forest can be applied.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Onat, I., Miri, A.: An intrusion detection system for wireless sensor networks. In:International Conference on Telecommunications (2017)

[2] Pozzolo, A.D., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card frauddetection: A realistic modeling and a novel learning strategy. IEEE Transactionson Neural Networks & Learning Systems 29(8), 3784–3797 (2018)

[3] Schlegl, T., Seebck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsu-pervised anomaly detection with generative adversarial networks to guide markerdiscovery (2017)

[4] Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXivpreprint arXiv:1901.03407 (2019)

[5] Bayer, J., Osendorfer, C.: Learning stochastic recurrent networks. Eprint Arxiv(2015)

[6] Pham, N., Pagh, R.: A near-linear time approximation algorithm for angle-basedoutlier detection in high-dimensional data pp. 877–885 (2012)

[7] Pham, N.: L1-depth revisited: A robust angle-based outlier factor in high-dimensional space. In: Joint European Conference on Machine Learning andKnowledge Discovery in Databases. pp. 105–121. Springer (2018)

[8] Sathe, S., Aggarwal, C.C.: Subspace histograms for outlier detection in linear time.Knowledge & Information Systems pp. 1–25 (2018)

[9] Kim, C., Lee, J., Kim, R., Park, Y., Kang, J.: Deepnap: Deep neural anomaly pre-detection in a semiconductor fab. Information Sciences 457, S002002551830375X(2018)

[10] Malhotra, P., Vig, L., Shroff, G., Agarwal, P.: Long Short Term Memory Networksfor Anomaly Detection in Time Series p. 6 (2015)

[11] Sabokrou, M., Fathy, M., Hoseini, M., Klette, R.: Real-time anomaly detectionand localization in crowded scenes. In: Proceedings of the IEEE conference oncomputer vision and pattern recognition workshops. pp. 56–62 (2015)

[12] Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In:Proceedings of the 23rd ACM SIGKDD International Conference on KnowledgeDiscovery and Data Mining. pp. 665–674. ACM (2017)

[13] Slch, M., Bayer, J., Ludersdorfer, M., Smagt, P.V.D.: Variational inference foron-line anomaly detection in high-dimensional time series (2016)

[14] An, J., Cho, S.: Variational autoencoder based anomaly detection using recon-struction probability. Special Lecture on IE 2(1) (2015)

[15] 15. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deepautoencoding gaussian mixture model for unsupervised anomaly detection (2018)