UDC 004.032.26

MODELS AND METHODS FOR FINDING ANOMALIES IN TIME SERIES USING NEURAL NETWORKS

Ali Zhuban
Master's student, Computational and Data Science Department,
Astana IT University, Astana, Kazakhstan

Scientific Supervisor: Svitlana Biloshchytska

In today's world, where different devices collect big data on a daily basis, it is important to understand which indicators are the norm and which indicators are unusual. It is the detection of anomalies in time series that is one of the important tasks in many fields, such as manufacturing, finance and cybersecurity. This study will be conducted using SKAB datasets, which is specially designed by developers to test models. Datasets contain sensor readings (pressure, temperature, flow rate, etc.). In this work, models such as LSTM Autoencoder, Isolation Forest, and Hotelling's T2 method are used. As a result of the comparative analysis, it was found that the LSTM Autoencoder performed better than the other models with optimal results.

Keywords: Time series, Anomalies, LSTM, Isolation Forest, Data Analysis.

Introduction

Anomaly detection in time series plays a key role in areas such as finance, healthcare, cybersecurity, and industrial monitoring. A time series is a sequence of observations recorded over time that can exhibit various patterns, including trends, seasonality, and cyclical fluctuations. Anomalies in such data represent deviations from expected behavior and may indicate important events, system failures, fraudulent activities, or security threats [1,2]. Their identification is crucial to ensure the reliability of systems, improve process efficiency, and prevent financial or operational losses. Anomalies in time series can be classified into three main types: point-based, contextual, and collective. Point anomalies are individual observations that differ significantly from the rest of the sample, for example, a sharp spike in network traffic that may indicate a cyberattack. Contextual anomalies look unusual only in certain conditions, for example, a sharp increase in temperature in winter. Collective anomalies are characterized [3,4] by abnormal behavior of a group of data points, for example, an unexpected change in customer preferences. The process of detecting anomalies in time series faces a number of difficulties. First, the high variability of the data makes it difficult to distinguish between normal fluctuations and true anomalies. Secondly, trends and seasonal effects can hide deviations,

requiring careful data preprocessing. Third, anomalies are rare, which leads to unbalanced datasets and complicates model training and validation. Various approaches are used to identify anomalies, from statistical methods to machine learning and deep learning algorithms. Traditional statistical methods such as moving averages, autoregressive models, and control charts are highly interpretable, but they may not be able to handle complex data dependencies. Machine learning methods, including clustering, isolation forests, and support vectors, provide greater flexibility. Deep learning, represented by recurrent neural networks (RNNs), long-term short-term memory (LSTM) networks, and autoencoders, demonstrates high efficiency in detecting time dependencies and complex anomalies.

This study aims to explore and apply advanced methods for detecting anomalies in time series, with a particular focus on the SKAB dataset. In particular, the use of T2-Hotelling statistics will improve the accuracy of anomaly identification and improve monitoring of industrial systems.

Literature Review

This study [4] highlight that the rapid expansion of the Internet of Things (IoT) and the increasing use of sensors in industrial settings have led to the generation of vast amounts of complex data over time, known as multivariate time series data. This type of data provides a more comprehensive view by integrating information from multiple sensors. However, managing and preparing this data for analysis is challenging. Each sensor measures different attributes, operates at varying frequencies, and may have dependencies with other sensors, making the preprocessing phase time-consuming and requiring specialized domain knowledge. Time series anomaly detection plays a vital role in identifying unusual patterns in sequential data, making it valuable in various fields. In finance, for instance, it can detect fraudulent transactions, while in healthcare, it helps identify irregularities in vital signs. Traditional methods often struggle with the dynamic nature of time series data, but anomaly detection techniques can adapt more effectively. Additionally, these methods are efficient since they do not necessarily require labeled data for every anomaly.

Currently, extensive research is being conducted on time series anomaly detection [5], with different approaches tailored to specific domains. In this research [6], abnormal changes in GDP components over time were analyzed, while unusual weather patterns were identified based on wave heights across the four seas [7]. Despite the significant research in this field, there is still no universally accepted solution for detecting anomalies in time series data. By identifying anomalies, this technology enhances decision-making, helps prevent potential issues, and reduces costs. Ultimately, the ability to detect anomalies in time series data is crucial for extracting valuable insights from the vast and continuously growing datasets generated today.

This paper [8] examines the application of Temporal Convolutional Networks (TCNs) for detecting anomalies in multivariate time-series data. TCNs, known for their ability to capture long-range dependencies, are trained to predict future values, with anomalies identified based on prediction errors modeled using a multivariate Gaussian distribution. The model employs causal and dilated convolutions to ensure predictions rely only on past data while effectively capturing temporal dependencies. Residual connections enhance training stability, and multi-scale feature maps improve pattern recognition. The framework is tested on three real-world datasets: Electrocardiograms (ECG), space shuttle telemetry, and 2-D gesture data. Results indicate that TCNs with multi-scale features outperform standard TCNs in precision and F-score, demonstrating their effectiveness in identifying anomalies across diverse time-series patterns. This approach enhances anomaly detection in complex data, making it applicable to various real-world scenarios.

This article [9] presents TadGAN, an unsupervised anomaly detection method designed for time-series data using Generative Adversarial Networks (GANs). TadGAN addresses key challenges in anomaly detection, including the lack of labeled data, vague anomaly definitions, and complex temporal dependencies. The model utilizes LSTM-based Generators and Critics, incorporating cycle consistency loss to enhance time-series reconstruction. Training involves adversarial learning with Wasserstein loss and gradient penalty for stability. Anomaly scores are computed by combining reconstruction errors with Critic outputs, exploring techniques such as dynamic time warping. TadGAN is evaluated against eight baseline methods across 11 datasets from NASA, Yahoo, Numenta, Amazon, and Twitter, consistently achieving the highest average F1 score. The model is particularly effective in detecting collective anomalies and handling diverse anomaly types, demonstrating superior performance and generalizability in real-world time-series anomaly detection.

This research [10] present WANEH (Wavelets, Neural Networks, and Hilbert Transform), a deep learning-based anomaly detection algorithm for time-series data. This method is highly versatile, with applications in transportation, structural health monitoring, and earthquake prediction. WANEH learns normal system behavior without requiring anomalous training data, enhancing its adaptability across domains. It employs wavelet analysis for multi-resolution signal denoising and reconstruction while leveraging deep neural networks to capture both short- and long-term dependencies. Anomalies are identified through hierarchical analysis of residual signals using probabilistic ROC methods. Successfully applied to seismic electric signals for earthquake forecasting and smartphone data for road defect detection, WANEH demonstrates strong transferability with minimal adjustments. The study concludes that WANEH is a robust, efficient tool for real-time anomaly detection, significantly advancing expert systems in various fields.

This study is aimed at studying existing models for detecting anomalies in time series and their comparative analysis. The paper discusses various methods, including statistical approaches, machine learning, and deep neural networks. To evaluate the effectiveness and accuracy of the models, the SKAB dataset is used, which contains data on industrial equipment. The analysis is based on quality metrics such as F1 score and other metrics, which allows you to determine the most effective algorithms. The results of the study will help identify the strengths and weaknesses of various approaches.

Methods

Dataset

The SKAB dataset (Skoltech Anomaly Benchmark) is designed for detecting anomalies in data collected from industrial equipment. It consists of time-series data from sensors monitoring a hydraulic system, measuring parameters such as pressure, temperature, and fluid flow. The dataset includes both normal operating conditions and various anomalies caused by malfunctions or deviations in system performance. SKAB serves as a benchmark for testing and comparing anomaly detection methods, enabling the evaluation of their accuracy, adaptability, and generalization capabilities. With its structured format and labeled anomalies, this dataset is a valuable resource for developing and training machine learning and deep learning models aimed at industrial monitoring and fault detection.

datetime	Accelerometer1RMS	Accelerometer2RMS	Current	Pressure	Temperature	Thermocouple	Voltage	Volume Flow RateRMS	anomaly	changepoint
09.03.2020 10:34	0.0270797	0.039615	0.871339	0.054711	75.4955	25.8338	244.09099999999998	32.0	0.0	0.0
09.03.2020 10:34	0.026994999999999998	0.0387587	1.30128	0.054711	75.5445	25.8408	224.17	32.0	0.0	0.0
09.03.2020 10:34	0.026807099999999997	0.0395207	янв.45	0.38263800000000003	75.6607	25.8227	234.157	32.9986	0.0	0.0
09.03.2020 10:34	0.0268166	0.038630300000000006	1.36495	0.054711	75.575	25.8262	229.90200000000002	32.9986	0.0	0.0
09.03.2020 10:34	0.026392900000000004	0.038769800000000001	0.7918390000000001	0.054711	75.4351	25.8382	251.697	32.0015	0.0	0.0
09.03.2020 10:34	0.026646100000000002	0.039029400000000006	0.750212	0.38263800000000003	75.5475	25.8164	246.40400000000002	32.0	0.0	0.0
09.03.2020 10:34	0.026355200000000002	0.039175699999999994	1.26386	0.054711	75.562	25.8271	252.84400000000002	32.9986	0.0	0.0
09.03.2020 10:34	0.0264336	0.038891300000000004	0.670358	0.38263800000000003	75.5594	25.8319	232.287	32.0015	0.0	0.0
09.03.2020 10:34	0.026160500000000003	0.0383421	1.1567399999999999	0.054711	75.6128	25.8254	228.051	32.0	0.0	0.0
09.03.2020 10:34	0.026616700000000004	0.039701	1.14015	0.054711	75.5508	25.8136	238.87099999999998	32.9986	0.0	0.0
09.03.2020 10:34	0.0265192	0.0393903	0.717334	0.054711	75.4214	25.8173	225.102	32.0015	0.0	0.0
09.03.2020 10:34	0.026465499999999996	0.038479	0.706354	0.054711	75.5427	25.8181	222.423	32.0	0.0	0.0
09.03.2020 10:34	0.026818	0.0392095	0.722432	0.054711	75.5546	25.8236	248.792	32.0	0.0	0.0
09.03.2020 10:34	0.026886200000000002	0.0395024	1.02624	-0.27321599999999996	75.6409	25.8192	231.104	32.9986	0.0	0.0
09.03.2020 10:34	0.026950799999999997	0.03841	0.430271	0.38263800000000003	75.5952	25.8138	214.55599999999998	32.0015	0.0	0.0
09.03.2020 10:34	0.0266082	0.039156199999999995	0.8172020000000001	-0.27321599999999996	75.5374	25.8139	240.985	32.0	0.0	0.0
09.03.2020 10:34	0.027126	0.039795300000000006	0.5864779999999999	0.38263800000000003	75.5812	25.8121	220.19799999999998	32.0	0.0	0.0
09.03.2020 10:34	0.0267264	0.038764599999999996	0.9648610000000001	0.38263800000000003	75.4874	25.8176	229.05900000000003	32.0	0.0	0.0
09.03.2020 10:34	0.026802299999999998	0.0392944	0.993223	0.38263800000000003	75.5356	25.8072	231.609	32.0	0.0	0.0
09.03.2020 10:34	0.0266923	0.0395978	0.483839	0.054711	75.7014	25.822	213.68099999999998	32.9986	0.0	0.0
09.03.2020 10:34	0.026862900000000002	0.040122000000000005	0.8768799999999999	0.054711	75.6438	25.814	223.567	32.0015	0.0	0.0
09.03.2020 10:34	0.026370099999999997	0.0395298	0.613635	0.054711	75.7002	25.8196	222.257	32.9986	0.0	0.0
09.03.2020 10:34	0.0271039	0.0406657	0.634912	0.054711	75.6795	25.8162	231.982	32.0015	0.0	0.0
09.03.2020 10:34	0.026762599999999998	0.040183800000000006	0.514584	0.054711	75.7262	25.8182	228.66	32.0	0.0	0.0
09.03.2020 10:34	0.0271855	0.041388	0.7662899999999999	0.054711	75.6298	25.8182	224.274	32.0	0.0	0.0

Fig.1. SKAB Dataset

SKAB consists of 35 files, each dataset includes the following columns: datetime, Accelerometer1RMS, Accelerometer2RMS, Current, Pressure, Temperature, Thermocouple, Voltage, RateRMS, anomaly, and changepoint

Model Training and Evaluation

In this study, three anomaly detection methods were evaluated: Hotelling's T^2, LSTM Autoencoder, and Isolation Forest. Each of these models operates based on different principles and was assessed using key performance metrics, including F1 Score, False Alarm Rate (FAR), Missing Alarm Rate (MAR), and the NAB Score under different settings (Standard, Low False Positives, Low False Negatives). Hotelling's T^2 is a multivariate statistical approach that identifies anomalies by

measuring the Mahalanobis distance of observations from the dataset's mean vector. The method calculates a T^2 statistic for each instance, and anomalies are flagged based on a predefined p-value threshold (e.g., 0.99, 0.999). Additionally, Principal Component Analysis (PCA) can be applied to reduce dimensionality and focus on the most informative features. The model's performance was measured by its ability to balance anomaly detection accuracy while minimizing false alarms. The LSTM Autoencoder is a deep learning model designed for sequential anomaly detection. It consists of an encoder-decoder structure, where LSTM layers learn to compress the input sequence into a latent representation and then reconstruct it. The difference between the input and the reconstructed output, known as the reconstruction error, serves as the anomaly score. The model was trained using the Adam optimizer and a Mean Absolute Error (MAE) loss function, with Early Stopping implemented to prevent overfitting. Hyperparameters such as batch size, number of time steps (N_STEPS), and validation split were varied to analyze their impact on detection performance.

Finally, the Isolation Forest algorithm was examined as a tree-based ensemble method for anomaly detection. It isolates data points through recursive random partitioning, with anomalies typically being isolated faster than normal instances. The contamination parameter was adjusted to control the proportion of expected anomalies in the dataset, and different numbers of estimators were tested. The model's effectiveness was assessed based on its ability to achieve a high F1 Score while maintaining an optimal balance between false and missing alarms.

Each model’s performance was evaluated using F1 Score as the primary metric, complemented by False Alarm Rate (FAR) and Missing Alarm Rate (MAR) to quantify misclassifications. Additionally, the NAB Score was used to provide a standardized comparison of model performance under different sensitivity settings. The results obtained from each approach highlight the trade-offs between detecting anomalies accurately and minimizing false detections.

Result

This study investigates anomaly detection in time series data using three models: Hotelling’s T^2, LSTM Autoencoder, and Isolation Forest. Performance is assessed using F1 Score, False Alarm Rate, and Missing Alarm Rate.

Isolation Forest

The Isolation Forest model with different contamination values showed varying results. With contamination=0.02, the model achieved the lowest False Alarm Rate (20.81%), but at the cost of a high Missing Alarm Rate (61.47%), indicating it missed many anomalies. Increasing contamination to 0.05 balanced detection rates, leading to an F1 Score of 0.63. Figure 2 provides a detailed performance breakdown.

Hotelling’s T^2

The Hotelling’s T^2 method was tested with and without PCA-based dimensionality reduction. The PCA variant (explained variance = 0.9, p_value =

0.99) achieved a slightly better performance than the standard approach, reducing the False Alarm Rate from 26.42% to 23.3% while maintaining a similar Missing Alarm Rate. However, overall F1 scores remained lower than the best-performing models. Figure 3 illustrates the comparative performance of these configurations.

Model	Parameters	F1	Other metrics
Isolation Forest	n_jobs = -1, contamination=0.1	0.7	False Alarm Rate 45.21 % Missing Alarm Rate 25.1 % NAB: Standard - 3.92 LowFP - -0.49 LowFN - 6.26
Isolation Forest	n_jobs = -1, contamination=0.02	0.49	False Alarm Rate 20.81 % Missing Alarm Rate 61.47 % Standard - 20.89 LowFP - 13.17 LowFN - 26.17
Isolation Forest	n_jobs = -1, contamination=0.05 n_estimators=100	0.63	False Alarm Rate 32.28 % Missing Alarm Rate 40.91 % Standard - 12.45 LowFP - 7.22 LowFN - 15.85

Fig.2. Performance metrics of Isolation Forest models with different contamination values

Hotelling's T2	scaling=True, using_pca=False explained_variance=0.85, p_value=0.999	0.56	False Alarm Rate 26.42 % Missing Alarm Rate 52.15 % Standard - 12.81 LowFP - 4.52 LowFN - 17.13
Hotelling's T2	scaling=True, using_pca=True, explained_variance=0.9, p_value=0.99	0.58	False Alarm Rate 23.3 % Missing Alarm Rate 50.77 % Standard - 11.94 LowFP - -3.09 LowFN - 18.12

Fig.3 Performance comparison of Hotelling's T^2 with and without PCA.

LSTM Autoencoder

The LSTM Autoencoder with BATCH_SIZE=64 achieved the lowest Missing Alarm Rate (30.84%), reducing undetected anomalies. However, this came with a higher False Alarm Rate (39.42%). The best balance among the LSTM configurations was achieved with BATCH_SIZE=32 and N_STEPS=10, yielding an F1 Score of 0.67. Figure 4 compares the performance of different LSTM configurations.

LSTM Autoencoder	EPOCHS = 100 BATCH_SIZE = 32 VAL_SPLIT = 0.1 N_STEPS = 10	0.67	False Alarm Rate 33.33 % Missing Alarm Rate 34.86 % Standard - 20.62 LowFP - 15.77 LowFN - 23.64
LSTM Autoencoder	EPOCHS = 100 BATCH_SIZE = 16 VAL_SPLIT = 0.1 N_STEPS = 5	0.62	False Alarm Rate 31.79 % Missing Alarm Rate 43.51 % Standard - 19.07 LowFP - 10.59 LowFN - 24.43
LSTM Autoencoder	EPOCHS = 100 BATCH_SIZE = 64 VAL_SPLIT = 0.2 N_STEPS = 10	0.68	False Alarm Rate 39.42 % Missing Alarm Rate 30.84 % Standard - 16.54 LowFP - 12.14 LowFN - 19.36

Fig.4. Performance metrics of LSTM Autoencoder models with different batch sizes and sequence lengths.

The anomaly detection methodology has some limitations. The dataset may contain imbalances in certain features, affecting the overall detection performance. Additionally, parameter sensitivity in models like Isolation Forest and LSTM Autoencoder affects anomaly detection results, requiring fine-tuning for different datasets. The choice of window size (N_STEPS) in LSTM models significantly impacts model accuracy, as shorter sequences may miss long-term dependencies. Furthermore, real-world time series data often contains missing values and noise, which can impact anomaly detection effectiveness.

Conclusion

This study explores anomaly detection in time series data using three distinct models: Hotelling's T^2, LSTM Autoencoder, and Isolation Forest. Various configurations of these models were tested to evaluate their effectiveness in identifying anomalies while minimizing false and missing alarms. The results demonstrated that the Isolation Forest model with a contamination level of 0.1 achieved the highest F1 Score (0.70) but at the cost of a high False Alarm Rate (45.21%). In contrast, the same model with a contamination level of 0.02 significantly reduced false alarms (20.81%) but suffered from a high Missing Alarm Rate (61.47%), leading to missed detections. The LSTM Autoencoder models provided a more balanced performance, with the batch size of 64 achieving the lowest Missing Alarm Rate (30.84%), though with a slightly higher False Alarm Rate (39.42%). Among different configurations, the LSTM Autoencoder with a batch size of 32 and N_STEPS of 10 offered the best balance between detection accuracy and alarm rates (F1 Score: 0.67, FAR: 33.33%, MAR: 34.86%). Hotelling's T^2 performed better with PCA, reducing its False Alarm Rate from 26.42% to 23.3%, although the improvement was relatively small.

These findings highlight the trade-offs in anomaly detection methods, with deep learning models offering better adaptability to time series data, while statistical and tree-based methods provide interpretable and computationally efficient alternatives. Future research could explore hybrid approaches that combine these techniques to enhance anomaly detection performance.

REFERENCES

1. Liu, S., Zhou, B., Ding, Q., Hooi, B., Zhang, Z., Shen, H., & Cheng, X. (2022). Time series anomaly detection with adversarial reconstruction networks. IEEE Transactions on Knowledge and Data Engineering, 35(4), 4293-4306.
2. Ji, Z., Gong, J., & Feng, J. (2021). A novel deep learning approach for anomaly detection of time series data. Scientific Programming, 2021(1), 6636270.
3. Cheng, H., Tan, P. N., Potter, C., & Klooster, S. (2009, April). Detection and characterization of anomalies in multivariate time series. In Proceedings of the 2009 SIAM international conference on data mining (pp. 413-424). Society for Industrial and Applied Mathematics.
4. Dix, M., Chouhan, A., Ganguly, S., Pradhan, S., Saraswat, D., Agrawal, S., & Prabhune, A. (2021, August). Anomaly detection in the time-series data of industrial plants using neural network architectures. In 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService) (pp. 222-228). IEEE
5. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019, July). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2828-2837).
6. Zhao, P., Chang, X., & Wang, M. (2021). A novel multivariate time-series anomaly detection approach using an unsupervised deep neural network. IEEE Access, 9, 109025-109041.
7. Gupta, M., Gao, J., Sun, Y., & Han, J. (2012). Community trend outlier detection using soft temporal pattern mining. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23 (pp. 692-708). Springer Berlin Heidelberg.
8. Kut, A., & Birant, D. (2006). Spatio-temporal outlier detection in large databases. Journal of computing and information technology, 14(4), 291-297.
9. He, Y., & Zhao, J. (2019, June). Temporal convolutional networks for anomaly detection in time series. In Journal of Physics: Conference Series (Vol. 1213, No. 4, p. 042050). IOP Publishing.

10. Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., & Veeramachaneni, K. (2020, December). Tadgan: Time series anomaly detection using generative adversarial networks. In 2020 ieee international conference on big data (big data) (pp. 33-43). IEEE.

11. Kanarachos, S., Christopoulos, S. R. G., Chroneos, A., & Fitzpatrick, M. E. (2017). Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and Hilbert transform. Expert Systems with Applications, 85, 292-304.

МОДЕЛИ И МЕТОДЫ ОБНАРУЖЕНИЯ АНОМАЛИЙ ВРЕМЕННЫХ РЯДОВ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

Али Жубан

Научный руководитель: Белощицкая С.

В современном мире, где различные устройства ежедневно собирают большие объемы данных, важно понимать, какие показатели являются нормальными, а какие - отклоняющимися от нормы. Обнаружение аномалий во временных рядах - одна из важных задач во многих сферах, таких как промышленность, финансы и кибербезопасность. Данное исследование проводится с использованием наборов данных SKAB, специально разработанных разработчиками для тестирования моделей. Датасеты содержат показания различных датчиков (давление, температура, скорость потока и др.). В работе используются такие модели, как LSTM Autoencoder, Isolation Forest и метод T² Хотеллинга. В результате сравнительного анализа было установлено, что модель LSTM Autoencoder показала наилучшие результаты по сравнению с другими методами.

Ключевые слова: временные ряды, аномалии, LSTM, Isolation Forest, анализ данных.

УАҚЫТТЫҚ ҚАТАРЛАРДАҒЫ АНОМАЛИЯЛАРДЫ НЕЙРОЖЕЛІЛЕР АРҚЫЛЫ АНЫҚТАУ ҮШІН МОДЕЛЬДЕР МЕН ӘДІСТЕР

Али Жубан

Ғылыми жетекші: Белощицкая С.

Қазіргі таңда әртүрлі құрылғылар күн сайын орасан зор көлемде деректер жинайтын заманда, қандай көрсеткіштердің қалыпты, ал қайсысының қалыптан

ауытқитынын түсіну өте маңызды. Уақыттық қатарлардағы аномалияларды анықтау — өнеркәсіп, қаржы, киберқауіпсіздік сияқты көптеген салаларда өзекті міндеттердің бірі болып табылады. Бұл зерттеу SKAB деп аталатын, модельдерді тестілеуге арнайы әзірленген деректер жиынтығының көмегімен жүргізіледі. Бұл деректер жиынтығы түрлі сенсорлардың көрсеткіштерін (қысым, температура, ағын жылдамдығы және т.б.) қамтиды. Жұмыста LSTM Autoencoder, Isolation Forest және Хотеллингтің T^2 әдісі сияқты модельдер қолданылды. Салыстырмалы талдау нәтижесінде LSTM Autoencoder моделі басқа әдістермен салыстырғанда ең жоғары нәтижелер көрсеткені анықталды.

Кілт сөздері: уақыттық қатарлар, аномалиялар, LSTM, Isolation Forest, деректерді талдау.