Software System for Intelligent Monitoring of Storage Systems Through Log Analysis

Eugeny Mytarin(✉), Vadim Moshkin, and Ilya Andreev

Ulyanovsk State Technical University, Severny Venets Street, 32, 432027 Ulyanovsk, Russia
eugenytornado9@yandex.ru

Abstract. The article presents a description of the developed software platform that provides automation of data storage server state monitoring using a hybrid algorithm. The developed algorithm combines an ontological approach and machine learning elements. A comparative assessment of various approaches was carried out: cluster analysis, autoencoders, LSTM, fuzzy ontologies and hybrid models - when analyzing methods for searching for anomalies in time series. The hybrid evaluation model showed the best results for all criteria. It was chosen to develop a prototype of the software platform. Experiments were conducted on hardware storage system data. The software platform proved its effectiveness in solving the problem.

Keywords: monitoring · anomalies · storage system · ontology · time series · software system

1 Introduction

Modern technical systems are highly complex and consist of many interacting components, which makes their monitoring and diagnostics a very difficult task. Traditional approaches to assessing the state of such systems often face problems of uncertainty, data inaccuracy and the need to process large amounts of information. In this regard, there is a need to search for new effective methods for analyzing and predicting the state of complex technical systems [1].

One of the promising areas is the integration of machine learning methods and fuzzy ontologies. Machine learning automates the process of extracting knowledge from large amounts of data and creates models that can adapt to changing operating conditions [2].

Fuzzy ontologies provide an opportunity to formalize expert knowledge and work with incomplete and inaccurate data. This is especially important when analyzing the states of complex technical systems [3].

Within the framework of this project, a software platform was implemented that automates the process of predictive diagnostics of a complex technical system (using a data storage server as an example) by integrating machine learning methods and subject ontology [4].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15650, pp. 336–345, 2025.
https://doi.org/10.1007/978-3-031-96962-1_23

2 Comparative Analysis of Technical Time Series Anomalies Detection Methods and Software

a. **Software analogues**

Currently, separate software tools are used to solve the problem of data storage system diagnostics. The most common and effective software systems are:

Papertrail is software that collects diagnostic data from text files, Apache, MySQL DBMS, Heroku applications, Windows event log, routers and other objects [5].

Loggly is a service for monitoring and analyzing event logs. It centralizes and analyzes logs from various devices and applications in real time. Loggly supports data collection via various protocols, including Syslog, HTTP/HTTPS, TCP and UDP [6].

Sumo Logic is a cloud platform for log management, monitoring and security. Sumo Logic supports data collection via various protocols, including Syslog, HTTP/HTTPS, TCP and UDP [7].

Splunk is a software platform for monitoring, analyzing and visualizing data that allows you to collect data from multiple sources: application logs, network equipment, databases and other devices, to analyze it in real time and draw conclusions based on the data obtained [8].

Each of these tools has its own unique features and advantages, which makes them suitable for different use cases in the field of assessing complex technical systems.

However, the main drawback of the presented systems is the lack of modules that provide forecasting of the system's behavior in the future based on the current values of the characteristics. These systems work only in real time.

b. **Analysis methods**

Let us present a comparative characteristic of the most popular approaches to detecting anomalies in technical time series.

The main parameters by which we will conduct the comparison are:

- accuracy: an assessment of how well the method correctly identifies anomalies in the data;
- speed: indicates the time required to process the data and obtain the results;
- interpretability: describes the ease of understanding the results of the method and the mechanisms of its operation;
- noise resistance: characterizes the ability of the method to maintain its effectiveness in the presence of random variations or errors in the data;
- memory consumption: indicates the amount of RAM required to execute the method [9, 10].

The comparative table of characteristics of time series anomaly detection methods is an important tool for systematizing and analyzing different approaches.

Table 1 helps researchers quickly evaluate the advantages and disadvantages of each method. This facilitates a more informed choice of algorithm depending on the specifics of the problem, available resources, and requirements for accuracy, speed, interpretability, and noise resistance.

In addition, this table helps identify the area where each method may be most effective or limited. This is essential for the successful application of the methods in real-world conditions.

Table 1. Comparative table of characteristics of methods for searching anomalies in time series

Method	Accuracy	Speed	Interpretability	Noise Resilience	Memory Consumption
Cluster Analysis	low (70 – 85%)	very high (1 – 10 s)	high (easy to visualize and explain clusters)	low (robust to some types of noise, but sensitive to outliers)	low (100 – 500 MB) [11, 12]
Autoencoders	High (80 – 95%)	medium (30 – 600 s)	low (lack of explicit rules makes it difficult to determine which features of the data influence their inferences)	High (able to deal effectively with noise due to its filtering ability)	high (1 – 10 GB) [13, 14]
LSTM	high (85 – 95%)	low (60 – 600 s)	low (it is difficult to understand the influence of individual time dependencies on the model's output)	High (good tolerance to noise, especially in time series)	high (1 – 8 GB) [15, 16]
Fuzzy ontologies	medium (75 – 85%)	high (5 – 60 s)	medium (rules can be understandable and accessible for analysis, their complexity and number can make it difficult to fully understand the results)	Medium (the ability to take into account uncertainties and noise, but the effectiveness depends on the quality of the rules and the structure of the ontology)	medium (200 – 800 MB) [17]
Hybrid model	very high (90 – 98%)	low (60 – 600 s)	high (the ability to explain the results through fuzzy rules, while LSTM processes time dependencies)	high (combination of the strengths of both methods)	very high (3 - 15 GB) [18, 19]

In this regard, the task of implementing a predictive diagnostics approach to technical systems using intelligent data analysis methods in the form of a software platform is relevant.

3 Software Platform Architecture

The software platform developed within the project is intended for two groups of users - the system user and the subject area expert:

– the subject area expert creates a fuzzy ontological model, fills in SWRL rules and sets fuzzy annotations [20];
– the system user loads the prepared CSV source data, selects the values by which the time series will be created, searches for anomalies using the selected intelligent model (using the LSTM model as an example), and obtains a semantic output for each metric using the fuzzy ontology [21].

The main use cases are shown in Fig. 1.

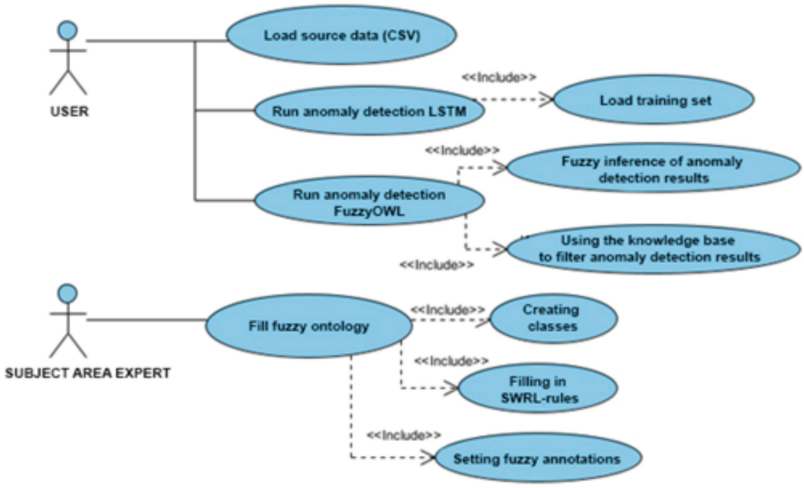


Fig. 1. Diagram of use cases for the developed software platform

The developed software platform consists of the following software modules:

- The preprocessing module loads the source data.
- The neural network module is the pytorch library, which allows training the LSTM model to work with the processed data.
- The module for working with fuzzy ontology is the owlready2 library, a file with fuzzy ontology in OWL format, SWRL rules in a text document, the FuzzyOwl plugin for working with fuzzy ontology and an API for working with SWRL rules [22, 23].

Figure 2 shows a diagram of the components of the developed software platform.

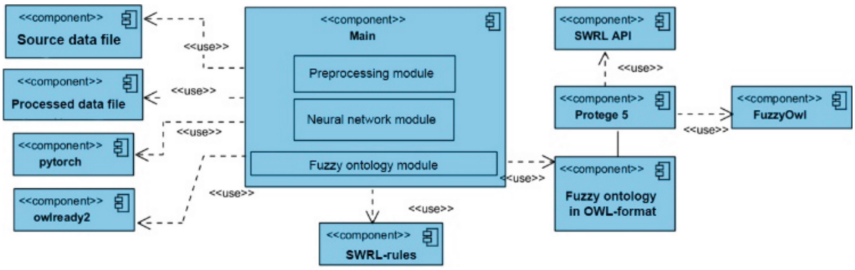


Fig. 2. Component diagram

4 Implementation of the Software Platform Prototype and Working Example

The software platform assumes the ability to load data in the form of time series, previously extracted from the logs of the technical system, pre-processed and presented as a CSV file. For operation, the source file must contain a "date" column, which stores the values of the collection date of all metrics (Table 2).

After loading the file, the user has the opportunity to select one of the columns from the dataset (Fig. 3).

The "Build a graph" button creates a time series, with the X axis showing the collection date of the metric selected by the column and the Y axis showing the values of this metric (Fig. 4).

The "LSTM anomaly search" button starts training the neural network. The architecture of the used LSTM network is shown in Fig. 5.

The autoencoder uses the backpropagation method.

The proposed neural network architecture includes the following layers:

- ConvLSTM1D is an LSTM layer. All input and output transformations within ConvLSTM1D are convolutional transformations.
- Conv1DTranspose is the transposed convolutional layer.
- Dropout is a layer that serves to prevent retraining of the neural network.

Once completed, the graph will highlight the expected anomalous metric values with red dots (Fig. 6).

When you click the "Load ontology" button, a dialog box appears for selecting a file in owl/rdf format to load the fuzzy ontology.

A fuzzy ontology file can contain various classes and properties. A file with SWRL rules can contain an unlimited number of rules. Each rule must start on a new line. The example of a SWRL rule for the data storage system condition assessment domain is shown below:

Metric(?m) ^ MetricHasName(?m, "totalNumberCreate") ^ hasValue(?m, ?value) ^ hasProbabilityOfAnomaly(?m, ?prob) ^ equal(?prob, 1) ^ greaterThanOrEqual(?value, 200) - > hasConclusion(?m, "Anomaly detected. Synchronization failure is highly probable.").

When you click the "FUZZY Anomaly Search" button, the fuzzy ontology processing begins. A new ontology object is created for each metric by the selected column, the values from the source data, the name are recorded, and the probability of the anomaly is set using the trapezoidal membership function.

After recording each metric, SWRL rules are applied. First, if the conditions are met, the has Anomaly properties are set, and then for all methods, the hasConclusion property is recorded with a fuzzy inference (Fig. 7).

The user can specify the upper bound of the trapezoidal membership function when the metric value with 100% probability will be anomalous (Fig. 8).

Table 2. Example of csv file for uploading to information system

lineNumber	date	time	countPers	countContracts	totalNumberCreate	totalNumberUpdate	createPers	createCitizenship	…	createInfo	createHistory
1	02.11.2024 0:00	126	5117	0	27	7	0	3	…	12	3
2	04.11.2024 0:00	173	5116	0	44	8	0	3	…	26	3
3	08.11.2024 0:00	138	5126	0	97	10	0	11	…	39	11
4	09.11.2024 0:00	139	5122	5079	24	2	2	2	…	10	2
5	11.11.2024 0:00	189	5138	5078	164	44	3	19	…	75	20
6	12.11.2024 0:00	217	5138	5080	458	8	24	24	…	106	24
7	14.11.2024 0:00	212	5139	5159	74	8	4	7	…	26	7
8	17.11.2024 0:00	174	5145	5166	90	12	7	9	…	35	9
…	…	…	…	…	…	…	…	…	…	…	…
n	29.11.2024 0:00	196	5135	5156	29	0	2	2	…	12	2

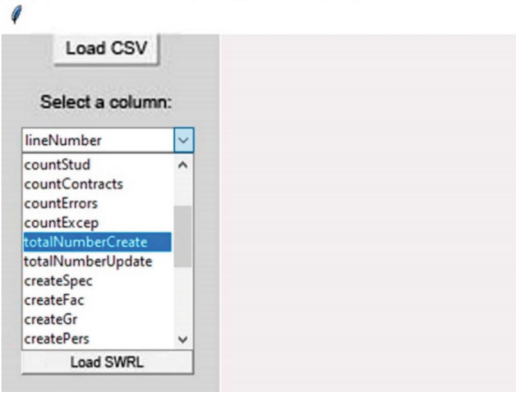


Fig. 3. Selecting a column

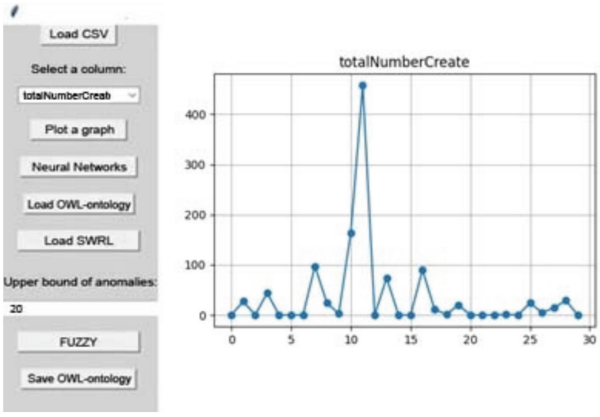


Fig. 4. Plotting a graph based on the original data

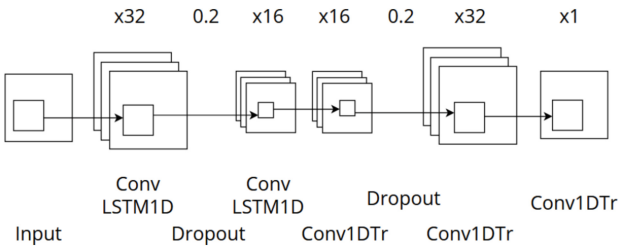


Fig. 5. Neural network structure

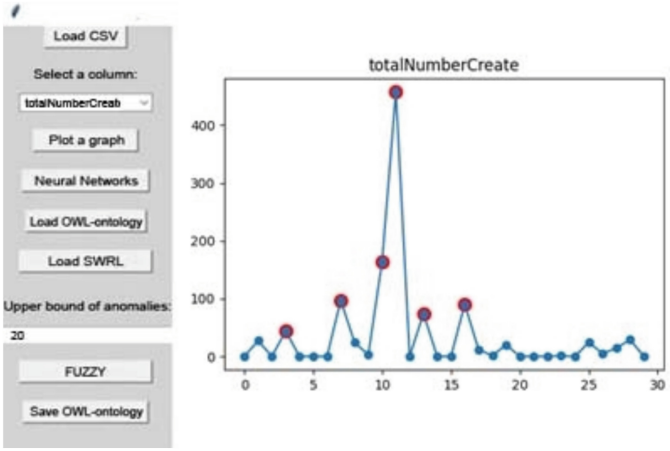


Fig. 6. Detected anomalies using LSTM

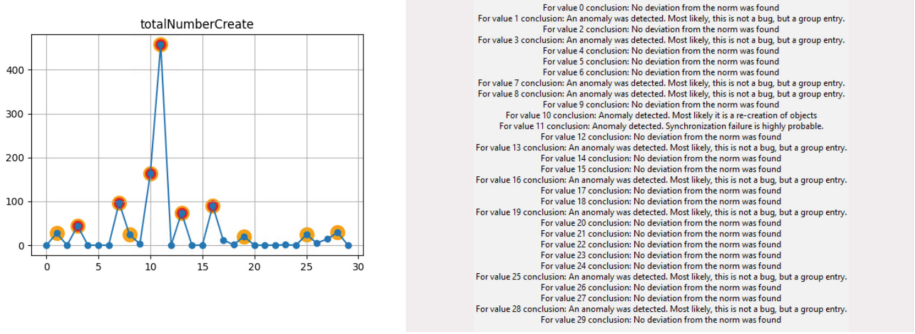


Fig. 7. Detected anomalies using fuzzy ontology and fuzzy inference

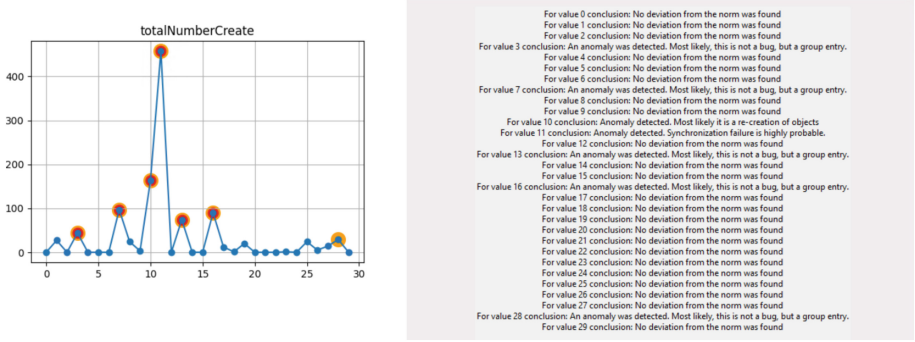


Fig. 8. Anomalies with a changed upper anomaly boundary

5 Conclusions

As a result of the project, the methods and services for assessing the state of complex technical systems were analyzed.

As part of the analysis of methods for searching for anomalies in time series, a comparative assessment of various approaches was carried out, including cluster analysis, autoencoders, LSTM, fuzzy ontologies and hybrid models. The best results for all criteria were demonstrated by the hybrid model for assessing the state of complex technical systems, which was selected for the development of a prototype.

To achieve the goal of the study, a software platform for searching for time series anomalies was designed and developed, using neural networks and a fuzzy ontological model.

The initial data were prepared, a fuzzy ontological model and SWRL rules were formed, and a prototype of the hybrid model was developed.

In general, the prototype allows you to:

- Load the initial data and select metrics for constructing a time series;
- Detect anomalies in a time series using LSTM and plot a graph;
- Load fuzzy ontology and SWRL rule files;
- Detect anomalies using fuzzy ontology and assign fuzzy logical inference to each metric;
- Save the new ontology with detected anomalies to a file.

Further development of the research may involve using the system in an enterprise with other data and changing the system architecture to work with real-time data.

Acknowledgments. This study was supported the Ministry of Science and Higher Education of Russia in framework of project No. 075-03-2023-143 “The study of intelligent predictive analytics based on the integration of methods for constructing features of heterogeneous dynamic data for machine learning and methods of predictive multimodal data analysis”.

References

1. Ilyin, E., Dryuchin, E.: Preparation of multidimensional data for training a neural network model in the problem of anomaly detection. Science, students, education: current issues of modern research: collection of articles from the III International scientific and practical conference: in 2 parts, Penza, November 30, 2022. Volume Part 1, Penza: Science and Education, pp. 62–68 (2022)
2. Bobillo, F., Straccia, U.: DeLorean: a reasoner for fuzzy OWL 2. Expert Syst. Appl. **39**(1), 258–272 (2012)
3. Bobillo, F., Straccia, U.: Fuzzy ontology representation using OWL 2. Int. J. Approximate Reasoning **52**(7), 1073–1094 (2011)
4. Fang, X., Yuan, Z.: Performance enhancing techniques for deep learning models in time series forecasting. Eng. Appl. Artif. Intell. **85**, 533–542 (2019). https://doi.org/10.1016/j.engappai.2019.07.011
5. Papertrail. https://www.papertrail.com. Accessed 22 Feb 2025
6. Loggly. https://www.loggly.com. Accessed 22 Feb 2025

7. Sumo Logic. https://www.sumologic.com. Accessed 22 Feb 2025
8. Splunk. https://www.splunk.com. Accessed 22 Feb 2025
9. Hewamalage, H., Ackermann, K., Bergmeir, C.: Forecast evaluation for data scientists: common pitfalls and best practices Data Min. Knowl. Discov. **37**(2), 788–832 (2023). https://doi.org/10.1007/s10618-022-00894-5
10. Mosin, V.G., Kozlovsky, V.N., Pantyukhin, O.V.: Detection of information channel anomalies based on predictive models in solving content quality analysis problems. Bullet. Tula State Univ. Tech. Sci. **3**, 421–425 (2024)
11. Polezhaev, M.N., Finogeev, A.A.: Predictive analysis of critical event indicators using a recurrent neural network with a transformer. Modern Sci. Intensive Technol. **4**, 63–68 (2024)
12. Himeur, Y., et al.: Artificial intelligence based anomaly detection of energy consumption in buildings: a review, current trends and new perspectives Appl. Energy, **287**, 116601 (2021). https://doi.org/10.1016/j.apenergy.2021.116601
13. Dietterich, T.: Machine learning for sequential data: a review Struct. Syntactic Stat. Pattern Recogn. 15–30 (2002). https://doi.org/10.1007/3-540-70659-3_2
14. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**, 504–507 (2006)
15. Bohara, B., et al.: Short-term aggregated residential load forecasting using BiLSTM and CNN-BiLSTM. In: Proceedings of the International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), pp. 37–43. IEEE
16. Elchenkov, R.A., Dunaev, M.E., Zaitsev, K.S.: Forecasting time series in real-time streaming data processing. Int. J. Open Inf. Technol. **10**(6), 62–69 (2022)
17. Fuzzy Ontology Representation using OWL 2. - http://www.umbertostraccia.it/cs/software/FuzzyOWL/index.html. Accessed 22 Feb 2025
18. Moshkin, V., Kurilo, D., Yarushkina, N.: Integration of fuzzy ontologies and neural networks in the detection of time series anomalies. Mathematics **11**(5), 1204 (2023)
19. Moshkin, V., Egov, E., Gavrilova, Y.: Development of a time series forecasting method based on entropy measures of diagnostic information. In: 2024 International Ural Conference on Electrical Power Engineering (UralCon), Magnitogorsk, Russian Federation, pp. 725–729 (2024). https://doi.org/10.1109/UralCon62137.2024.10718924
20. Protégé: ontology editor. - https://protege.stanford.edu. Accessed 22 Feb 2025
21. Trofimov, I.V.: Evolution of the expressive capabilities of the OWL language. Softw. Syst. Theory Appl. **2**, 4(8), 85–94 (2011)
22. Arkadov, G.V., Kotsoev, K.I., Trykova, I.V.: Detection of anomalous events of the detection system of free and loosely fixed objects using a convolutional autoencoder. questions of atomic science and technology. Series Nuclear Reactor Constants, 3, 245–253 (2023)
23. Moshkin, V.S., Kurilo, D.S., Andreev, I.A.: Hybridization of ontologies and neural networks in the problems of detecting anomalies of time series. Pattern Recognit Image Anal. Image Anal. **33**, 425–431 (2023). https://doi.org/10.1134/S105466182303032X