mathematics

MDPI

Article

Integration of Fuzzy Ontologies and Neural Networks in the Detection of Time Series Anomalies

Vadim Moshkin *, Dmitry Kurilo and Nadezhda Yarushkina

Department of Information Systems, Ulyanovsk State Technical University, Severny Venets Str., 32, Ulyanovsk 432027, Russia
* Correspondence: v.moshkin@ulstu.ru

Abstract: This paper explores an approach to solving the problem of detecting time series anomalies, taking into account the specifics of the subject area. We propose a method based on the integration of a neural network with long short-term memory (LSTM) and Fuzzy OWL (Fuzzy Web Ontology Language) ontology. A LSTM network is used for the mathematical search for anomalies in the first stage. The fuzzy ontology filters the detection results and draws an inference for decision making in the second stage. The ontology contains a formalized representation of objects in the subject area and inference rules that select only those anomaly values that correspond to this subject area. In the article, we propose the architecture of a software system that implements this approach. Computational experiments were carried out on free data of technical characteristics of drilling rigs. The experiments showed high efficiency, but not the maximum efficiency of the proposed approach. In the future, we plan to select a more efficient neural network architecture for mathematical anomaly detection. We also plan to develop an algorithm for automatically filling the rules of inference into the ontology when analyzing text sources.

Keywords: time series; fuzzy ontology; Fuzzy OWL; anomaly; LSTM; SWRL; inference

MSC: 8T37

Citation: Moshkin, V.; Kurilo, D.; Yarushkina, N. Integration of Fuzzy Ontologies and Neural Networks in the Detection of Time Series Anomalies. *Mathematics* **2023**, *11*, 1204. https://doi.org/10.3390/math11051204

Academic Editor: Andrzej Sokołowski

Received: 17 January 2023
Revised: 22 February 2023
Accepted: 27 February 2023
Published: 1 March 2023

1. Introduction

Anomaly detection is an area of data mining that allows one to find values that stand out from the total mass.

An anomaly is a section of the time series in which the behavior of an object does not correspond to expected forecasts or deviates significantly from typical behavior [1].

There are many methods for detecting anomalies. Most methods examine individual objects for differences from normal objects. However, they do not take into account the aspect of the temporal significance of the data. These anomalies are also referred to as point anomalies.

The anomaly detection problem for time series (TS) is a traditional anomaly detection problem. Numerous studies including Chandola et al. [2], Agyemang et al. [3], and Hodge et al. [4] have studied the problem of anomaly detection. Several methods have been proposed for character sequence anomaly detection. They are presented in a review by Chadola et al. [5]. At the same time, there are a limited number of methods for detecting anomalies for one-dimensional and multidimensional TS.

It is important to pay attention not only to the detection of anomalies by the value of the parameter, but also to the detection of anomalies by the time intervals of the series [6–8].

The time series anomaly can manifest itself in different ways [9,10]. It may appear:

- In the long-term preservation of a certain trend;
- In frequent changes of tendencies;
- In changing the frequency of the series;

- In the value of the indicator below or above, a certain threshold value for a long period of time.

The following anomaly search tasks are distinguished:

- Recognition of anomalies by the context of the studied series [11];
- Identification of discrepancies when compared with the ideal (predicted) series;
- Recognition of anomalies in noisy series (separation of noise and anomalous values) [12].

Anomalies can be classified according to the problem of detecting anomalies in time series data:

- Detection of contextual anomalies in TS [13–15];
- Detection of an anomalous subsequence within a given time series [16,17];
- Detection of anomalous time series from database time series [18].

In addition to the classification of anomalies, the methods for detecting these anomalies are also important. All methods can be conditionally divided into two groups:

- Methods for searching for anomalies based on the analysis of only the original time series;
- Anomaly search methods based on comparison of the original time series with reference time series.

The anomaly search methods related to the first group were considered above. Let's consider what methods exist for searching for anomalies based on comparing the obtained time series with the reference ones. These methods include:

- Methods based on the basic kernel [19–30];
- Methods based on windows [31–33];
- Markov methods [28,30];
- Hidden Markov methods [34–40].

Currently, approaches to managing complex technical systems using hybrid algorithms are actively being developed.

For example, the author of [41] proposes the Virtual Reference Feedback Tuning of a combination of two control algorithms, Active Disturbance Rejection Control as a model-free control algorithm and fuzzy control, in order to exploit the advantages of data-driven control and fuzzy control.

In the paper by [42], an indirect adaptive iterative learning control scheme is proposed for both linear and nonlinear systems to enhance the P-type controller by learning from set points.

In [43], the authors counted about 158 different methods of anomaly search algorithms, which belong to different groups. The papers [44–48] also present a set of approaches for detecting anomalous points and anomalous sequences. Most of the considered approaches are based on machine learning algorithms.

As can be seen from the above examples, most of the existing methods for searching for time series anomalies (including in control problems) depend on the training sample, and past values and cannot take into account the specifics of a particular subject area. In one subject area, a particular value is anomalous, but in another, it is not. In this regard, an urgent task is to develop a hybrid approach that would take into account the semantics of time series and at the same time not lose efficiency in the search for anomalies.

In addition, the problem of semantic interpretation of the results of the analysis and taking into account the features of the subject area is relevant [49]. Within the framework of this study, an iterative approach was developed to identify anomalies in time series, taking into account the characteristics of the subject area. The subject area is represented as a Fuzzy OWL (Fuzzy Web Ontology Language) ontology.

The article is organized as follows: Section 2 introduces the general structure of the developed time series anomaly search algorithm. The block diagram of the algorithm and the architecture of the LSTM network are presented. The LSTM network is used at the first stage of the anomaly search algorithm. It also describes the process of fuzzification of the output values of a neural network for integration with a fuzzy ontology.

Section 3 presents the rig domain ontology model, lists the classes and axioms. The axioms validate the numerical values obtained at the output of the LSTM network. Section 3 provides an example of an SWRL rule. The inference of the anomaly analysis results is carried out using a set of SWRL rules.

Section 4 describes in detail the software architecture that implements the proposed algorithm.

Finally, experiments on the application of the developed anomaly search method using the rig domain ontology are presented and discussed in Section 5.

Concluding remarks and directions for future research are presented in Section 6.

2. Hybrid Time Series Anomaly Search Algorithm

A feature of the proposed approach is the consistent application of the anomaly detection technique using LSTM networks (long short-term memory) to time series, the fuzzification of values, and the logical inference of search results using an ontology containing a set of SWRL (Semantic Web Rule Language) rules including fuzzy modifiers [50]. The general algorithm is shown in Figure 1.

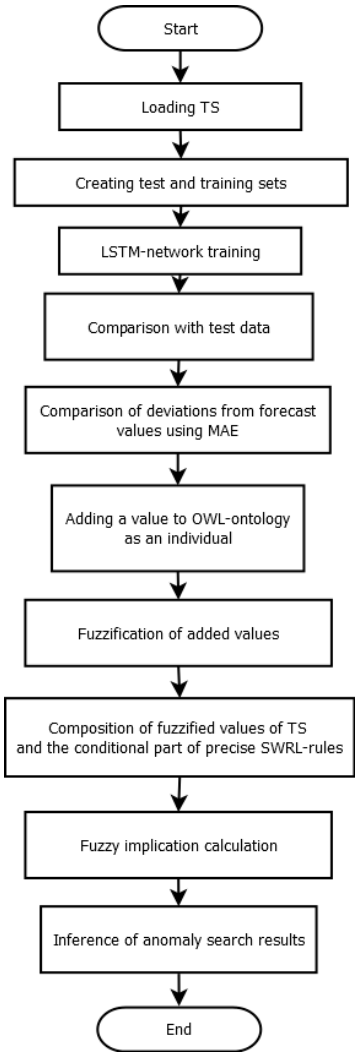

Figure 1. Proposed anomaly detection algorithm.

The first step is to search for an anomaly by means of a neural network of the structure shown in Figure 2 [51].

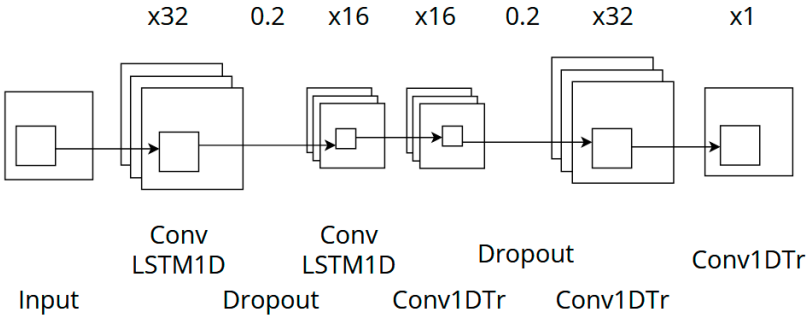

Figure 2. Neural network structure.

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 [52–55].

This type of neural network architecture was chosen based on previous studies [56]. However, the goal of our study was not to choose the most efficient neural network architecture in determining the anomaly of the time series. Our goal is to create an approach that takes into account the semantics of the subject area. Therefore, instead of the LSTM network, any efficient neural network architecture can be used that satisfies the conditions of the problem [56].

After receiving a set of anomalous values, the algorithm assumes the following operations:

- Fuzzification of anomalous TS data. Fuzzification involves the interpretation of clear anomalous x_0 values as a fuzzy point.
- Composition of the input variable and the conditional part of the SWRL rules contained in the ontology: $x_0 \circ A_i$, $y_0 \circ B_i$. The ontology contains a set of SWRL rules that contain fuzzy modifiers. If the fact is given by a fuzzy point, then at this stage the corresponding degree of membership is calculated.
- Calculation of fuzzy implication

$$(x_0 \circ A_i) \wedge (y_0 \circ B_i) \rightarrow C_i \forall R.$$

The result of these steps for all rules is n fuzzy values for output Z.

3. Applying a Domain Ontology

OWL-ontology (Web Ontology Language) was applied for semantic interpretation and filtering of anomaly search results in time series.

The OWL-ontology contains classes and objects of the subject area on which the experiments were carried out. The developed model of the ontology includes a set of SWRL rules in addition to classes, objects, and relations. Rules contain the following elements:

- Classes of the subject area;
- Class of anomalous values;
- Relationships between classes of the subject area;
- Auxiliary relation, which confirms the truth of the judgment about the anomaly of the time series;
- Variables [57].

The subject area of drilling rigs was chosen for experiments. All data are publicly available on the Internet.

The data used contains information about 5 drilling rigs. More than 3000 values of time series of several characteristics of drilling rigs were used to train the neural network and experiments.

The architecture of the developed ontology is shown in Figure 3.

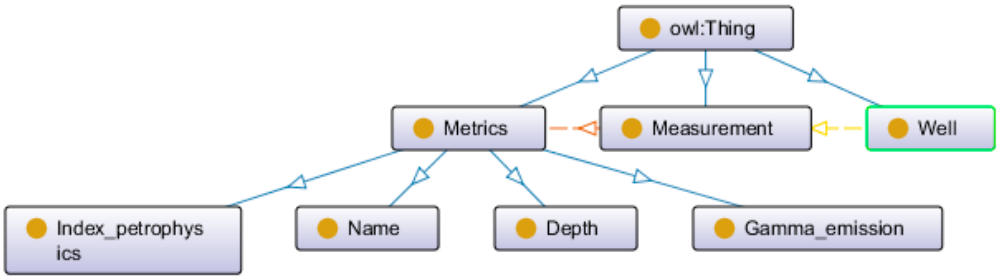

Figure 3. Fragment of the developed OWL ontology.

The ontology includes the following classes:

- "Well" includes the corresponding real domain objects.
- "Measurement" includes time series values that correspond to objects of the "Metrics" class.
- "Metrics" includes objects that establish a relationship between a "Measurement" and some domain metric.

The SWRL rules included in the ontology also include the classes described above. An example of the developed SWRL rule is shown in Figure 4.

```
Well(?w) ^ hasMeasurement(?w, ?m) ^ Measurement(?m) ^
hasMetrics(?m, ?index) ^ Index_petrophysics(?index) ^ hasIndexP(?index, ?indexValue) ^ hasMetrics(?m, ?dep) ^
Depth(?dep) ^ hasDepth(?dep, ?depValue) ^ swrlb:lessThan(?indexValue, 0.5) -> hasAnomaly(?m, "rule1")
```

Figure 4. An example of the developed SWRL rule.

The presented SWRL rule provides the solution for the following tasks:

- Checking the completeness of rig metrics data.
- Checking the value of the petrophysical index for reaching the threshold value of 0.5.

This rule uses the obtained value of the time series and draws a logical conclusion using a parameter that is not used in the neural network.

The atom consists of checking the achievement of a constant parameter. If the value is not reached, then the measurement can be considered abnormal, otherwise the value is correct.

The ontological analysis of anomaly search results using LSTM networks begins with the removal of all existing objects in the ontology. The algorithm writes to the ontology all the anomalies found by the neural network at each iteration. The algorithm unloads all the values available in the ontology after the operation of the SWRL rules.

Objects are written into the ontology in accordance with the described ontology model (Figure 3). The rules are run after recording the anomalous values found by the neural network [58].

4. Time Series Anomaly Search Software System

The software system that implements the described algorithm consists of several modules. The architecture of the developed software system is shown in Figure 5.

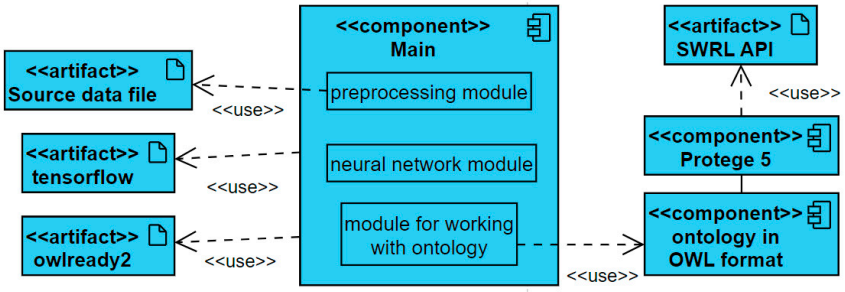

Figure 5. Component diagram.

The initial data for analysis are loaded from an external file. The ontology is stored in a file with the owl extension.

The developed software system uses the following libraries:

- OWLReady2 provides ontology processing in Python. The owlready2 library is used to unload and load ontology classes, objects, and relations from external sources.
- The sqrlb library functions are used for validation. The individual is put true in the identity property if all predicates are true.
- SWRL API provides an interface for creating and editing SWRL rules in the Protege 5 ontology editor [57,58].

The data preprocessing software module loads data from files, extracts table column headers and transfers them to the DataFrame structure. The timestamps for the data are the indexes of the rows in the resulting table.

The software system as a sequence of business processes with input data is shown as an IDEF0 diagram in Figure 6. The controllers are data about the subject area entered into the ontological model by an expert. The output of each block is sequentially passed to the next block.

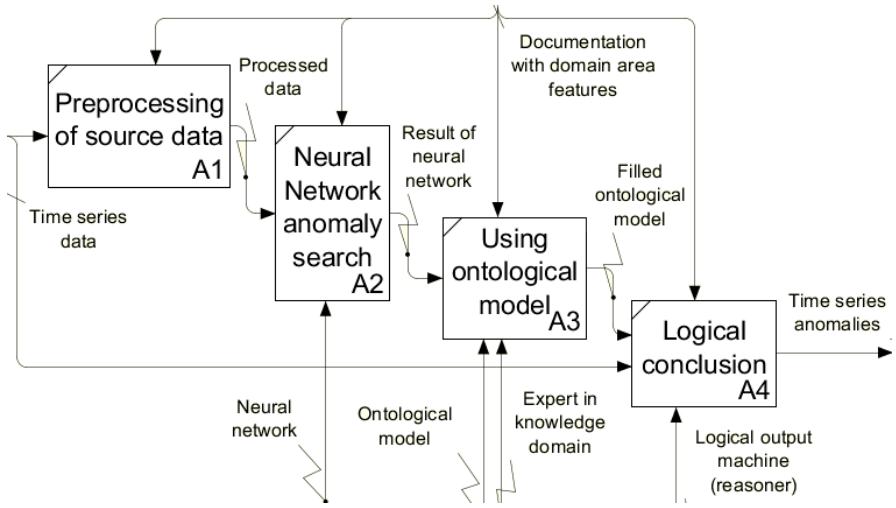

Figure 6. IDEF0 system view.

More about the neural system module and the ontology module in the IDEF3 diagrams is shown in Figures 7 and 8.

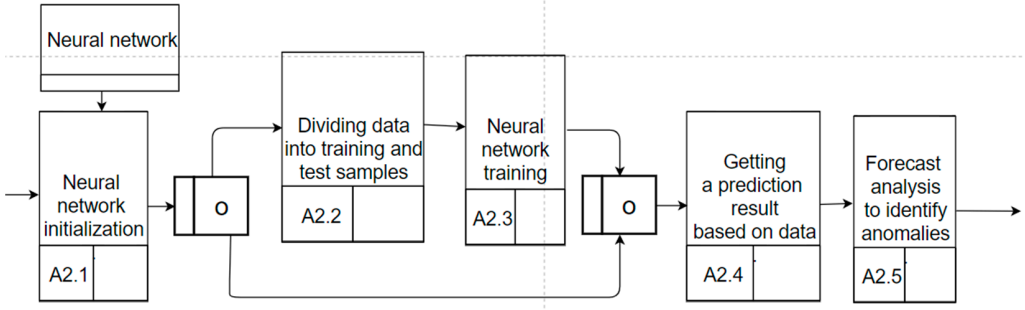

Figure 7. IDEF3 neural network module.

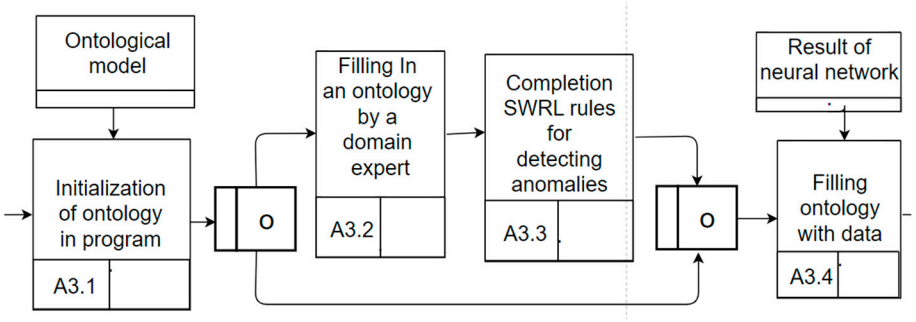

Figure 8. IDEF0 ontology module.

At the user's choice, it is possible to use an existing ontology for analysis or train a new one using the selected data.

At the user's choice, it is possible to use an already completed ontology for loading anomalies or to form a new model by a domain expert.

5. Experiments

The data used contain information about five drilling rigs [59]. The data are the facies logs from nine wells from the Council Grove gas reservoir located in Kansas. Facies are studied from core samples in every half foot and matched with logging data in well locations. Feature variables include five from wireline log measurements and two geologic constraining variables that are derived from geologic knowledge.

More than 3000 values of time series of several characteristics of drilling rigs were used to train the neural network and experiments.

Characteristics of the drilling rigs selected for the experiments are the following:

- GR—measure gamma emission;
- Delta PHI—porosity index in petrophysics;
- ILD_log10—resistivity measurement.

Graphs of the initial data are presented in Figure 9.

The measurements were carried out at regular intervals. Timestamps are placed from the current day to the end of measurements in one day increments. Storing the date in indexes allows the data to be processed faster and build graphs.

Experiments were carried out with different sizes of the training set to assess the accuracy of the neural network and reduce the amount of initial data for network training. The graph of the result is shown in Figure 10. The proximity of the indicator to zero indicates a greater similarity of the predicted values to the present.

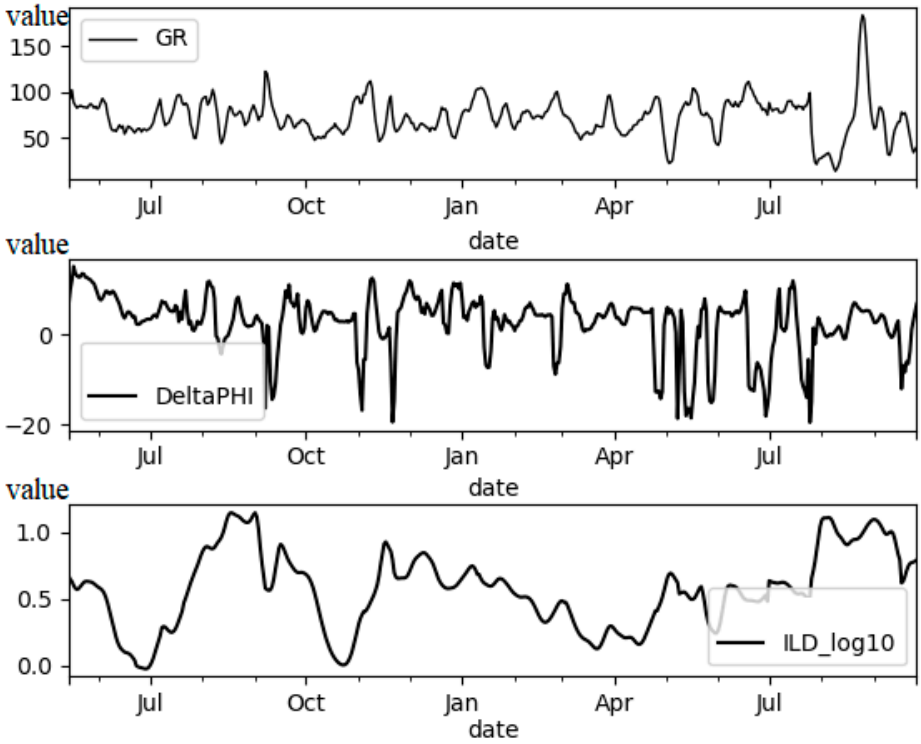

Figure 9. Fragment of source data.

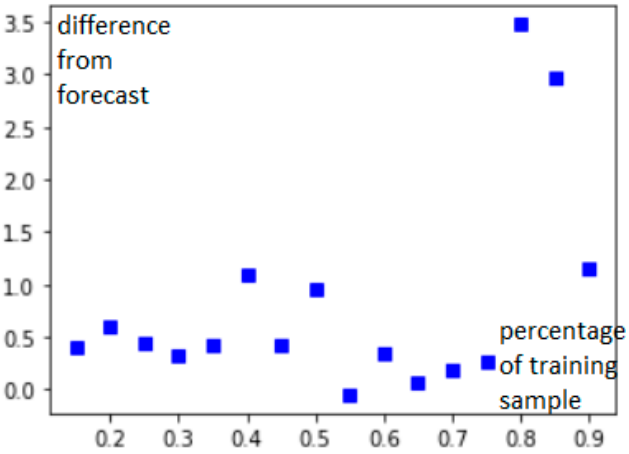

Figure 10. Dependence of accuracy on the size of the training set.

The data show overfitting of the model with a sample of more than 80%, the recommended values are about 55–65%, and a slight deviation from the forecast with a small training sample of 25–35%. This indicates the possibility of using a small amount of data for training. This will lead to an increase in the number of false positives that will be processed later at the stage of the ontological model.

The initial data are an array of independent samples of different oil rigs. Data can be displayed on graphs. The preprocessed data are fed into the neural network described in Section 2. Gamma radiation measurements were used for the experiments for presentation on graphs. Each graph refers to a separate well; 50% of the original data were used to train the neural network, 50% were used as a test set.

About 400 measurements are in the examples presented. The results of applying the LSTM–neural network are shown in Figures 11–13. Anomalous values found by the neural network are highlighted in red.

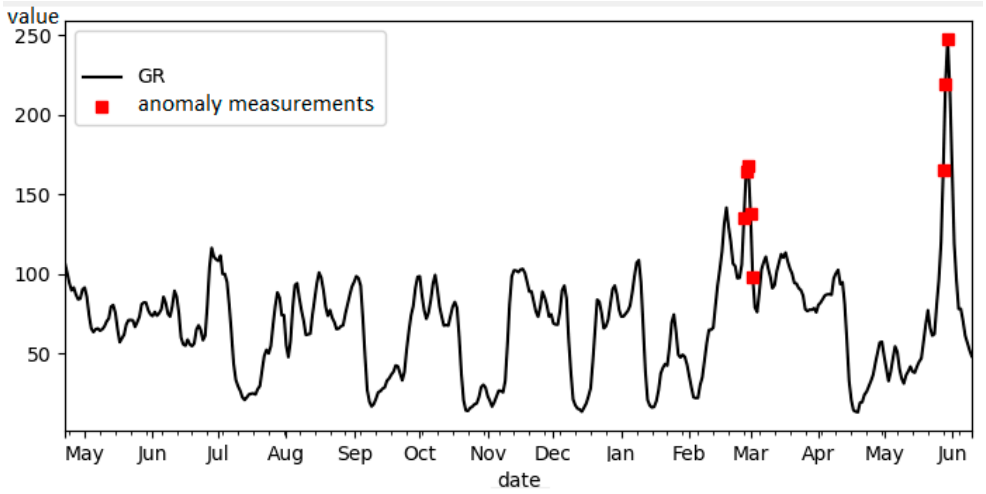

Figure 11. Data sampling 1.

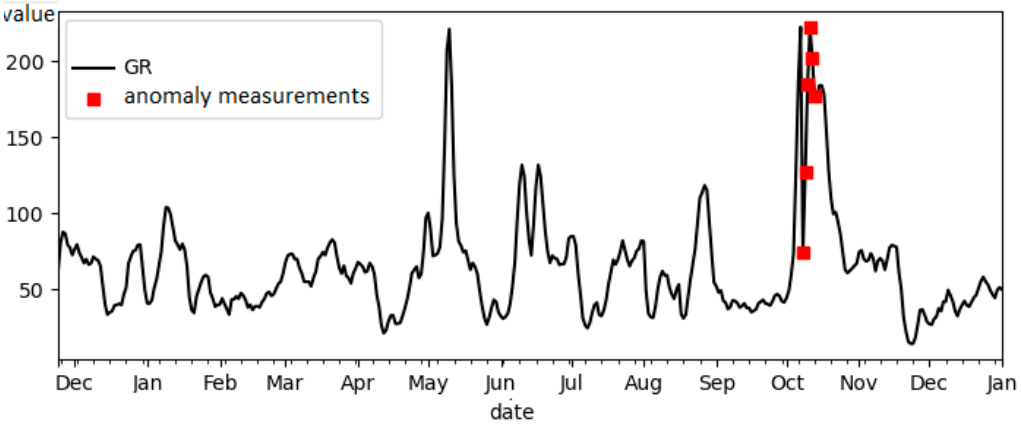

Figure 12. Data sampling 2.

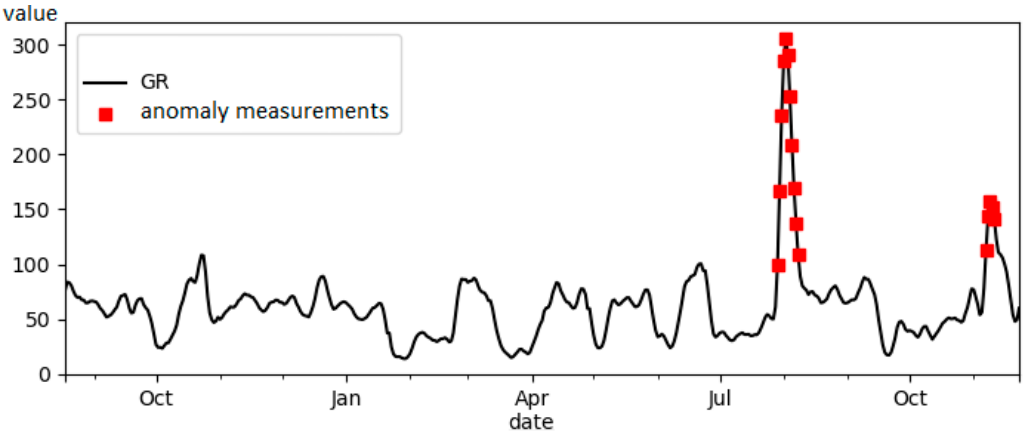

Figure 13. Data sampling 3.

Most of the anomalous values are found by the neural network. However, the neural network does not allow for taking into account the complex behavior of data and the features of the subject area. Therefore, the ontology is used to filter the found anomalous values at the next stage.

All objects are written into the ontology in accordance with its model. In the loop, the anomaly flag is checked and all values accepted by the ontology as truly abnormal are selected (Figure 14).

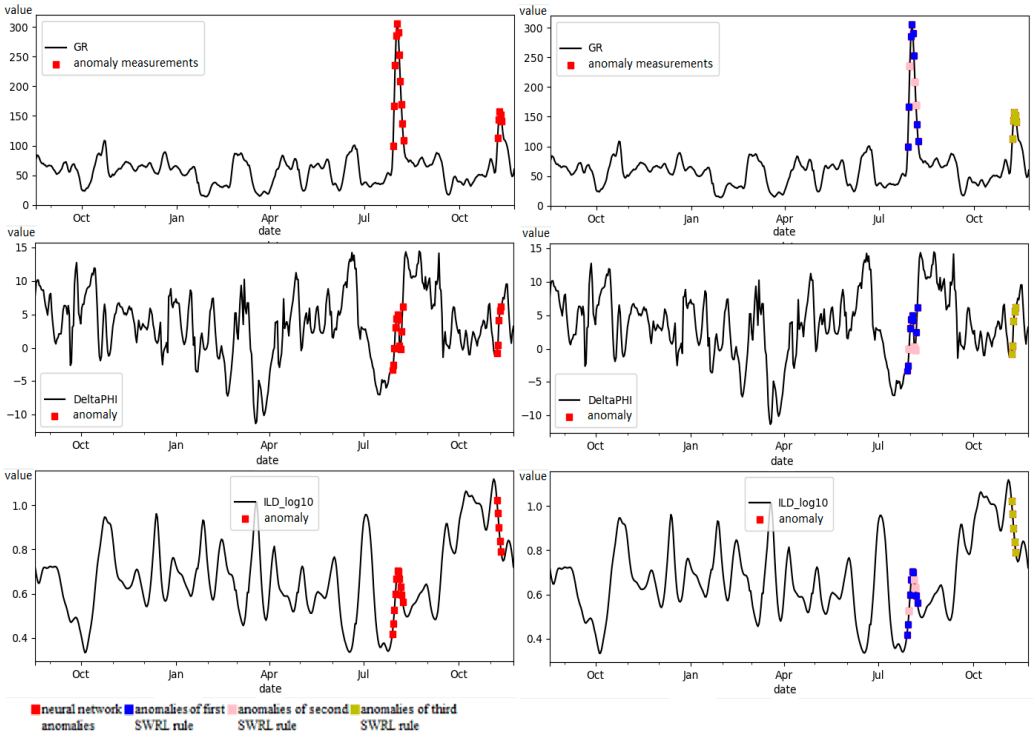

Figure 14. Results of the system operation.

The left side of the graphs shows the anomalous values found only by the neural network. On the right side of the graphs, the result of applying the ontology is presented. Different colors of points correspond to different applied SWRL rules from the ontology. Each of the rules uses different metrics from GR, Delta PHI, and ILD. Separation by several rules can also classify anomalies into categories.

The Mean Absolute Error (MAE) metric was used to assess the quality of the developed algorithm. MAE reaches 80% in the framework of the experiments. This value is not the maximum. Neural networks of this type achieve an accuracy well over 90 percent.

Advantages of the developed approach are as follows:

- The algorithm takes into account the specifics of the subject area when searching for anomalies in time series. A particular value is anomalous in one subject area, but not in another. The use of neural networks does not solve this problem, and since it is based on training data, training data may not be enough. An ontology stores the knowledge of experts in a given subject area.
- The algorithm showed high efficiency (but not maximum) in accordance with the MAE metric.

However, this approach has a number of disadvantages:

- Experts develop an ontology of the subject area, in particular, experts develop a set of SWRL rules that filter anomaly search results using a neural network. This approach is labor intensive. Therefore, we plan to develop an approach for automated extraction of rules from text resources.
- The efficiency of anomaly detection at the first stage of the algorithm (using neural networks) depends on the quality of the training sample. Therefore, the approach works more efficiently on large data samples. This problem is partially solved by taking into account the regularities of the subject area, embedded in the ontology.

6. Conclusions

This article presents the developed method for detecting anomalies in time series, taking into account the specifics of the subject area, presented in the form of a fuzzy ontology.

The approach involves the use of LSTM networks for the mathematical search for anomalies. A fuzzy ontology filters anomaly detection results and draws inferences for decision making. The paper also presents a number of computational experiments to search for anomalous values in the technical characteristics of drilling rigs. For this, open data were used. Based on the results of the experiments, the following conclusions can be drawn:

- Using an ontology together with neural networks allows you to semantically filter anomaly search results. Using only machine learning algorithms does not solve the problem in which a particular value is anomalous in one subject area, but not in another. Using only machine learning does not fully solve the problem, since it is based on training data, training data may not be enough. An ontology stores the knowledge of experts in a given subject area. Thanks to this, the developed algorithm can be transferred from one subject area to another: to find deviations in the traffic of computer networks, in the oil pressure of helicopter units, in weather temperature, etc. It is just the ontology that is needed to be changed.
- The algorithm showed high efficiency (but not maximum) in accordance with the MAE metric. It is possible to modify the architecture of the neural network to improve efficiency.

During the development of the project, we plan to conduct a series of experiments to change the architecture of the neural network. This is necessary to obtain higher anomaly detection results at the first stage of the algorithm.

At the second stage of the algorithm, we plan to add metric classes. This will allow you to operate with variables that are not tied to the names of specific parameters, but have links with metric classes.

We plan to expand the set of SWRL rules to more accurately take into account the features of the subject area under consideration. We also plan to develop an approach for automated extraction of rules from text resources using NLP methods (Natural Language Processing).

Author Contributions: Conceptualization, N.Y.; Software, D.K.; Formal analysis, V.M. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported Ministry of Science and Higher Education of Russia in framework of project No. 075-00233-20-05 from 3 November 2020 «Research of intelligent predictive multimodal analysis of big data, and the extraction of knowledge from different sources».

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://www.kaggle.com/datasets/imeintanis/well-log-facies-dataset (accessed on 10 January 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, X.; Zhang, L.; Cao, Y.; Jin, K.; Hou, Y. Anomaly Detection Approach in Industrial Control Systems Based on Measurement Data. *Information* **2022**, *13*, 450. [CrossRef]
2. Aggarwal, C. On abnormality Detection in Spuriously Populated Data Streams. In Proceedings of the 5th SIAM Data Mining Conference, Newport Beach, CA, USA, 21–23 April 2005; pp. 80–91.
3. Du, W.; Fang, L.; Peng, N. Lad: Localization anomaly detection for wireless sensor networks. *J. Paral. Distrib. Comput.* **2006**, *66*, 874–886. [CrossRef]
4. Hickinbotham, S.; Austin, J. Novelty Detection in Airframe Strain Data. In Proceedings of the International Conference on Pattern Recognition, Barcelona, Spain, 15 December 2000; pp. 536–539.
5. Budalakoti, S.; Srivastava, A.; Otey, M. Anomaly detection and Diagnosis Algorithms for Discrete Symbol Sequences with Applications to Airline Safety. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Montreal, QC, Canada, 7–10 October 2007; Volume 37.
6. Krupski, J.; Graniszewski, W.; Iwanowski, M. Data Transformation Schemes for CNN-Based Network Traffic Analysis: A Survey. *Electronics* **2021**, *10*, 2042. [CrossRef]

7. Wen, T.; Keyes, R. Time series anomaly detection using convolutional neural networks and transfer learning. *arXiv* **2019**, arXiv:1905.13628.
8. Hwang, R.-H.; Peng, M.-C.; Huang, C.-W.; Lin, P.-C.; Nguyen, V.-L. An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection. *IEEE Access* **2020**, *8*, 30387–30399. [CrossRef]
9. Scholkopf, B.; Williamson, R.; Smola, A.; Shawe-Taylor, J.; Platt, J. Support vector method for novelty detection. *Adv. Neural Inf. Process. Syst.* **2000**, *12*, 582–588.
10. Hurdle, E.E.; Bartlett, L.M.; Andrews, J.D. System Fault Diagnostics Using Fault Tree Analysis. In Proceedings of the 16th Conference on Advances in Reliability Technology Symposium (ARTS), Leics, UK, 3 April 2005; pp. 203–222.
11. Korbicz, J.; Kowal, M. *Intelligent Systems in Technical and Medical Diagnostics*; Springer: Dordrecht, The Netherlands, 2013; p. 530.
12. Keogh, E.; Lonardi, S.; Chiu, B.Y.C. Finding Surprising Patterns in a Time Series Database in Linear Time and Space. In Proceedings of the 8th ACM SIGKDD International Conference on Kowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 550–556.
13. Campbell, C.; Bennett, K. A linear programming approach to novelty detection. *Adv. Neural Inf. Process. Syst.* **2000**, *13*.
14. Zhu, W.; Beroza, G.C. PhaseNet: A Deep-Neural-Network-Based Seismic Arrival Time Picking Method. *Geophys. J. Int.* **2018**, *216*, 261–273. [CrossRef]
15. Wu, W.; He, L.; Lin, W.; Su, Y.; Cui, Y.; Maple, C.; Jarvis, S.A. Developing an Unsupervised Real-Time Anomaly Detection Scheme for Time Series with Multi-Seasonality. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 4147–4160. [CrossRef]
16. Thill, M.; Konen, W.; Bäck, T. Time Series Encodings with Temporal Convolutional Networks. In *International Conference on Bioinspired Methods and Their Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 161–173.
17. Marín, G.; Casas, P.; Capdehourat, G. Rawpower: Deep learning based anomaly detection from raw network traffic measurements. In Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos, Budapest, Hungary, 20–25 August 2018; pp. 75–77.
18. Park, D.; Hoshi, Y.; Kemp, C.C. A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1544–1551. [CrossRef]
19. Dridi, A.; Boucetta, C.; Hammami, S.E.; Afifi, H.; Moungla, H. STAD: Spatio-Temporal Anomaly Detection Mechanism for Mobile Network Management. *IEEE Trans. Netw. Serv. Manag.* **2020**, *18*, 894–906. [CrossRef]
20. Tajer, A.; Veeravalli, V.V.; Poor, H.V. Outlying sequence detection in large data sets: A data-driven approach. *IEEE Signal Process. Mag.* **2014**, *31*, 44–56. [CrossRef]
21. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection for discrete sequences: A survey. *IEEE Trans. Knowl. Data Eng.* **2010**, *24*, 823–839. [CrossRef]
22. Dean, D.A.; Goldberger, A.L.; Mueller, R.; Kim, M.; Rueschman, M.; Mobley, D.; Sahoo, S.; Jayapandian, C.P.; Cui, L.; Morrical, M.G.; et al. Scaling Up Scientific Discovery in Sleep Medicine: The National Sleep Research Resource. *Sleep* **2016**, *39*, 1151–1164. [CrossRef] [PubMed]
23. Chiu, B.; Keogh, E.; Lonardi, S. Probabilistic discovery of time series motifs. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 493–498.
24. Bicego, M.; Murino, V.; Figueiredo, M.A. A sequential pruning strategy for the selection of the number of states in hidden Markov models. *Pattern Recognit. Lett.* **2003**, *24*, 1395–1407. [CrossRef]
25. Hadjieleftheriou, M.; Kollios, G.; Tsotras, V.J.; Gunopulos, D. Efficient Indexing of Spatiotemporal Objects. In *Advances in Database Technology—EDBT 2002: 8th International Conference on Extending Database Technology Prague, Czech Republic, 25–27 March 2002*; Springer: Berlin/Heidelberg, Germany, 2002.
26. Salvador, S.; Chan, P. Learning States and Rules for Detecting Anomalies in Time Series. *Appl. Intell.* **2005**, *23*, 241–255. [CrossRef]
27. Shmueli, G.; Fienberg, S.E. *"Current and Potential Statistical Methods for Monitoring Multiple Data Streams for Bio-Surveillance", Statistical Methods in Counter-Terrorism*; Wilson, A., Olwell, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2006.
28. Rebbapragada, U.; Protopapas, P.; Brodley, C.E.; Alcock, C. Finding anomalous periodic time series: An application to catalogs of periodic variable stars. *arXiv* **2009**, arXiv:0905.3428. [CrossRef]
29. Ozkan, H.; Ozkan, F.; Kozat, S.S. Online Anomaly Detection Under Markov Statistics with Controllable Type-I Error. *IEEE Trans. Signal Process.* **2015**, *64*, 1435–1445. [CrossRef]
30. Que, J.; Tsui, F.-C. Rank-based spatial clustering: An algorithm for rapid outbreak detection. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 218–224. [CrossRef]
31. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
32. Que, J.; Tsui, F.-C. Spatial and Temporal Algorithm Evaluation for Detecting Over-The-Counter Thermometer Sale Increases during 2009 H1N1 Pandemic. *Online J. Public Health Inform.* **2012**, *4*, ojphi.v4i1.3915. [CrossRef]
33. Cheng, H.; Tan, P.; Potter, C.; Klooster, S. Detection and characterization of anomalies in multivariate time series. In Proceedings of the Ninth SIAM International Conference on Data Mining, Sparks, NV, USA, 30 April—2 May 2009.
34. Brumley, D.; Poosankam, P.; Song, D.; Zheng, J. Automatic patch-based exploit generation is possible: Techniques and implications. In Proceedings of the 2008 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 18–21 May 2008; pp. 143–157.
35. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer Science Business Media LLC: New York, NY, USA, 2006.
36. Bolton, R.J.; Hand, D.J. Statistical Fraud Detection: A Review. *Stat. Sci.* **2002**, *17*, 235–255. [CrossRef]
37. He, H.; Luo, X. A novel HMM-based approach to anomaly detection. *J. Inf. Comput. Sci.* **2004**, *1*, 91–94.

38. Chawla, N.V.; Japkowicz, N.; Kotcz, A. Special issue on learning from imbalanced data sets. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 1–6. [CrossRef]
39. Joshi, S.S.; Phoha, V.V. Investigating hidden Markov models capabilities in anomaly detection. In Proceedings of the 43rd Annual Southeast Regional Conference—Volume 2; ACM: New York, NY, USA, 2005; pp. 98–103.
40. Ghosh, S.; Reilly, D.L. Credit card fraud detection with a neural-network. In System Sciences, 1994. In Proceedings of the Twenty-Seventh Hawaii International Conference on System Science, Vol 3, Information Systems: DSS/Knowledge-Based Systems, Los Alamitos, CA, USA, 4–7 January 1994.
41. Roman, R.C.; Precup, R.E.; Petriu, E.M. Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems. *Eur. J. Control.* **2021**, *58*, 373–387. [CrossRef]
42. Chi, R.; Li, H.; Shen, D.; Hou, Z.; Huang, B. Enhanced P-type control: Indirect adaptive learning from set-point updates. *IEEE Trans. Autom. Control* **2022**, *68*, 1600–1613. [CrossRef]
43. Schmidl, S.; Wenig, P.; Papenbrock, T. Anomaly detection in time series: A comprehensive evaluation. *Proc. VLDB Endow.* **2022**, *15*, 1779–1797. [CrossRef]
44. Wenig, P.; Schmidl, S.; Papenbrock, T. TimeEval: A benchmarking toolkit for time series anomaly detection algorithms. *Proc. VLDB Endow.* **2022**, *15*, 3678–3681. [CrossRef]
45. Boniol, P.; Linardi, M.; Roncallo, F.; Palpanas, T.; Meftah, M.; Remy, E. Unsupervised and scalable subsequence anomaly detection in large data series. *VLDB J.* **2021**, *30*, 909–931. [CrossRef]
46. Chen, R.-Q.; Shi, G.-H.; Zhao, W.-L.; Liang, C.-H. A joint model for IT operation series prediction and anomaly detection. *Neurocomputing* **2021**, *448*, 130–139. [CrossRef]
47. Paparrizos, J.; Kang, Y.; Boniol, P.; Tsay, R.S.; Palpanas, T.; Franklin, M.J. TSB-UAD: An end-to-end benchmark suite for univariate time-series anomaly detection. *Proc. VLDB Endow.* **2022**, *15*, 1697–1711. [CrossRef]
48. Ryzhikov, A.; Borisyak, M.; Ustyuzhanin, A.; Derkach, D. NFAD: Fixing anomaly detection using normalizing flows. *PeerJ Comput. Sci.* **2021**, *7*, e757. [CrossRef] [PubMed]
49. Yarushkina, N.; Andreev, I.; Moshkin, V.; Moshkina, I. Integration of Fuzzy OWL Ontologies and Fuzzy Time Series in the Determination of Faulty Technical Units. In *Computational Science and Its Applications—ICCSA 2019. Lecture Notes in Computer Science*; Misra, S., Ed.; Springer: Cham, Switzerland, 2019; Volume 11619, pp. 545–555. [CrossRef]
50. Moshkin, V.; Yarushkina, N.; Moshkina, I. The Intelligent System for Interactive Analysis and Forecasting of Graph Data. In *Intelligent and Fuzzy Systems. INFUS 2022. Lecture Notes in Networks and Systems*; Kahraman, C., Tolga, A.C., Cevik Onar, S., Cebi, S., Oztaysi, B., Sari, I.U., Eds.; Springer: Cham, Switzerland, 2022; Volume 504, pp. 870–878. [CrossRef]
51. Malhotra, P.; Vig, L.; Shroff, G.M.; Agarwal, P. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN 2015: European Symposium on Artificial Neural Networks; Computational Intelligence and Machine Learning Bruge*; ESANN: Bruges, Belgium, 22–24 April 2015.
52. Krizhevsky, A. *Convolutional Deep Belief Networks on CIFAR-10*; University of Toronto: Toronto, ON, Canada, 2010.
53. Nikolov, T. *Recurrent Neural Network Based Language Model*; Brno University of Technology: Brno, Czech Republic, 2010.
54. Sak, H.; Senior, A.; Beaufays, F. *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling*; Google Inc.: Mountain View, CA, USA, 2014.
55. Moshkin, V.; Moshkina, I.; Yarushkina, N. The Software System for Calculating the Aggregated Forecast of Time Series. In *Open Semantic Technologies for Intelligent Systems. OSTIS 2021. Communications in Computer and Information Science*; Golenkov, V., Krasnoproshin, V., Golovko, V., Shunkevich, D., Eds.; Springer: Cham, Switzerland, 2022; Volume 1625, pp. 152–164. [CrossRef]
56. Kurilo, D.S.; Moshkin, V.S. Development of a system for hybrid detection of time series anomalies. In Proceedings of the 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russia, 3–27 May 2022; pp. 1–4. [CrossRef]
57. Jean-Baptiste, L. *Ontologies with Python*; Apress: Berkeley, CA, USA, 2020.
58. Yarushkina, N.; Moshkin, V.; Andreev, I.; Klein, V.; Beksaeva, E. Hybridization of fuzzy inference and self-learning fuzzy ontology-based semantic data analysis. In Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry"(IITI'16), Sochi, Russia, 16–21 May 2016. [CrossRef]
59. Well Log Facies Dataset. Available online: https://www.kaggle.com/datasets/imeintanis/well-log-facies-dataset (accessed on 10 January 2023).