Machine Learning and Artificial Intelligence
A.J. Tallón-Ballesteros and C. Chen (Eds.)

doi:10.3233/FAIA200777

Explainable AI: Using Shapley Value to Explain Complex Anomaly Detection ML-Based Systems

Jinying ZOU [a], Ovanes PETROSIAN [b,1]
[a] *Faculty of Applied Mathematics and Control Processes, Saint-Petersburg State University, Saint-Petersburg, Russia*
[b] *Faculty of Applied Mathematics and Control Processes, Saint-Petersburg State University, Saint-Petersburg, Russia*

Abstract. Generally, Artificial Intelligence (AI) algorithms are unable to account for the logic of each decision they take during the course of arriving at a solution. This ”black box” problem limits the usefulness of AI in military, medical, and financial security applications, among others, where the price for a mistake is great and the decision-maker must be able to monitor and understand each step along the process. In our research, we focus on the application of Explainable AI for log anomaly detection systems of a different kind. In particular, we use the Shapley value approach from cooperative game theory to explain the outcome or solution of two anomaly-detection algorithms: Decision tree and DeepLog. Both algorithms come from the machine learning-based log analysis toolkit for the automated anomaly detection ”Loglizer”. The novelty of our research is that by using the Shapley value and special coding techniques we managed to evaluate or explain the contribution of both a single event and a grouped sequence of events of the Log for the purposes of anomaly detection. We explain how each event and sequence of events influences the solution, or the result, of an anomaly detection system.

Keywords. Anomaly detection, Log anomaly detection, Shapley value, DeepLog, Decision tree, Explainable AI

1. Introduction

Recently, deep learning has made great contributions toward the rapid development of AI. For machine learning, especially deep learning, explainable AI is a big challenge. Deep neural networks are a black box for us all. AI algorithms usually cannot explain the logic of each decision when providing a solution. Such opaque decisions are not adequately persuasive, especially in the fields of military, medical and financial security where stakes are high. Therefore explainable AI would be helpful:

- For users when AI technology is designed to offer solutions or help take decisions. System users should then understand why the system provides each specific solu-

[1]Corresponding author is associate professor at the Department of Mathematical Modelling of Energetic Systems of Saint Petersburg State University. Email: petrosian.ovanes@yandex.ru

tion. For example, a doctor who makes a diagnosis needs to be able to understand why the medical diagnostic system makes such a recommendation [1].
- For developers in order to understand the black box of deep learning. They can improve their methods and models for building better machine learning models and improving system capabilities [2].

Anomaly detection is an important problem that has been well-studied within diverse research areas and domains of application. A common need when analyzing real-world data-sets is determining which instances stand out as being dissimilar to all others. Such instances are known as anomalies, and the goal of anomaly detection (also known as outlier detection) is to determine all such instances in a data-driven fashion [3]. Anomalies can be caused by errors in data but sometimes are indicative of a new, previously unknown, underlying process. Hawkins in [4] defines an outlier as an observation that deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism. The most common causes of outliers or anomalies in a data set are *Data entry errors* (human errors), *Measurement errors* (instrument errors), *Experimental errors* (data extraction or experiment planning/executing errors), *Intentional* (dummy outliers made to test detection methods), *Data processing errors* (data manipulation or data set unintended mutations), *Sampling errors* (extracting or mixing data from faulty or disparate sources) and *Natural* (not an error but a novelty in the data). With respect to methods or algorithms, anomaly detection should be classified as *Supervised*, *Unsupervised*, *Hybrid Models* and *one-Class Neural Networks*. By application, anomaly detection can be classified by *Intrusion Detection*, *Fraud Detection*, *Malware Detection*, *Medical Anomaly Detection*, *Social Networks Anomaly Detection*, *Log Anomaly Detection*, *Internet of things (IoT) Big Data Anomaly Detection*, *Industrial Anomalies Detection*, *Anomaly Detection in Time Series*, *Video Surveillance*. More details can be found in a more comprehensive recent overview [5]. There are also some potential approaches, used to increase the performance and precision when obtaining the anomaly, such as successful geometric transformations model [6] combined with regression model [7] and Ito decomposition [8], to overcome the time limitations. In this paper we focus on *Log Anomaly Detection* and study two anomaly detection algorithms. One involves a very classic and widely-used decision tree algorithm and the second one a more modern and advanced deep learning DeepLog algorithm. However the aim of the paper is not to study at some point the best anomaly detection algorithms, but rather to advance Explainable AI techniques by applying them to anomaly detection systems of a different sort.

Anomaly detection algorithms are often thought to be limited because they cannot facilitate the process of validating results performed by domain experts. This is an urgent challenge for the industry. In 2019, Antwarg used the SHAP framework [9] to explain an anomaly detection system. They treat each feature as a player, and provide users with a more intuitive understanding by measuring each player's contribution to the solution. SHAP is based on the notion of optimal Shapley Value [10], which is a well-known solution concept from cooperative game theory [11]. Originally the Shapley value defines how to allocate profit, costs or, more generally, a utility among the players acting cooperatively. In the case of Explainable AI, the Shapley value can show the contribution each feature makes to the result of the anomaly detection system. It is important to note that the Shapley value does not only show the individual contribution of a feature to the result of the detection system, but also shows the contribution of a feature to all possible

combinations of the feature that constitutes the anomaly. The SHAP approach itself for XAI was proposed by Lundberg in [12]. Other related papers on SHAP are described below. In [13] authors present an improved SHAP using the Baseline Shapley (BShap) method which they further extend by using Integrated Gradients to the continuous domain. The paper [14] explores the dependence between SHAP values by extending the KernelSHAP method to handle dependent features. In the paper [15] authors described an extension of the SHAP method for trees under a framework called TreeExplainer to examine the global model structure using local explanations. Later the paper [16] describes a SHAP-based method to account for the predictions of time-series signals involving Long Short-Term Memory (LSTM) networks.

Besides SHAP, there are several other useful and applied algorithms for explaining black box (machine learning) algorithms, but in this paper our particular interest lies in the XAI approaches based on the use of the Shapley value:

- LIME is a method that interprets individual model predictions based on building a local approximation the model around a given prediction [17].
- DeepLIFT (Deep Learning Important Features) [18] is a method for decomposing the output prediction of a neural network on a specific input by back-propagating the contributions of all neurons in the network to every feature of the input.
- LRP (Layer-wise Relevance Propagation) [19] is a method that brings such explanative ability to potentially highly complex deep neural networks. It operates by propagating the prediction backward in the neural network using a set of purposely designed propagation rules.

For a more comprehensive and fundamental overview for Explainable AI approaches and models see [20].

In this paper we apply the Shapley approach to two anomaly detection systems with different structures. The first is the Decision tree [21], [22] in which we treat a single feature as a player (the feature-player approach). We differ from Antwarg's approach [9] in that we treat different events in the anomaly detection system as a player based on the data itself without considering the algorithm model. Nor do we use the SHAP framework [9], but develop our own framework based on the Shapley value. The second anomaly detection system we want to explain is DeepLog [23]. Current mainstream research for Shapley-related XAI treats a single feature as a player and then analyzes the contribution of each player to the result. But this approach has limitations for a class of anomaly detection systems where the anomaly can be the result not only of one feature, but by a sequence of features. Therefore, we consider the sequence of events as a player for modeling. The key challenge is a large number [24] of coalitions in the sequence-player approach in comparison to that of a feature-player approach. In order to avoid the problem of a big number of sequence players, we use the bi-level method for calculating the Shapley value. A related method for cooperative coalitional games can be found in [25]. On the first level we consider one event as a player, on the second level we consider a sequence of two events as a player.

In section 2, we briefly introduce algorithms for both Decision tree and DeepLog. In Section 3, we describe the Shapley value and how to apply it to explain the decision tree and DeepLog. In section 4, we attach our simulation result. In section 5 presents conclusions and discussions for future work.

All our research is based on the open source project "Loglizer", which is a machine learning-based log analysis toolkit for automated anomaly detection [26]. Our code for this research was uploaded to Github: $https://github.com/ZouJinying/XAI_loglizer$.

2. Anomaly Detection Algorithms

2.1. Decision tree

In 2004 Mike Chen and others proposed using Decision tree in order to classify failed and successful requests for anomaly detection in large system logs [21]. Decision tree is a tree structure composed of nodes and branches. The nodes are divided into leaf nodes (representing a certain category) and internal nodes (representing a certain attribute or feature). Branches represent a test output. The basic idea of Decision tree is to use entropy as a measure to select the attributes of its nodes. Each selects the attributes with the largest gains, that is, the attribute with the smallest entropy value. When the entropy equals zero, all the instances in node are considered to be the same cluster. Below see a brief version of an anomaly detection Decision tree algorithm applied to system logs analysis:

1. Choose the event as a root. The best root will be chosen according to the information gain. The equation for calculating the information gain entropy is shown in Eq. (1) as below:

$$G(D,a) = H(D) - H(D \mid a) = H(D) + \sum_{i=1}^{N} \frac{|D^i|}{|D|} H(D^i), \tag{1}$$

   where, $G(D,a)$ is information gain, $H(D)$ is a summary of information entropy for all features in the set D, $H(D \mid a)$ is the conditional information entropy under the condition of feature a.
2. Divide the samples into two sub-trees and find the maximum gain.
3. Continue the iterations (from step1 to step3) until there are no events or features left in input data set.
4. Each branch of tree is a prediction result that displays if the log is anomaly or normal.

2.2. DeepLog

DeepLog is a data-driven algorithm that uses large number of system logs for anomaly detection. The main intuition behind the design of DeepLog comes from natural language processing: treat log entries as sequence elements that follow certain patterns and grammatical rules [23]. As distinct from the log message counter-method, DeepLog is a deep neural network that uses long and short-term memory to model log sequences. Therefore, the importance of the sequence of events is greater than the message count of events. This study thus considers the sequence as a player to explain each specific solution. In this way we can analyze which sequences contribute more to the accuracy of the prediction and whether the sequences are meaningful.

Figure 1 shows the architecture of DeepLog which contains two parts: training and detection. The training data for DeepLog comes from the system's logs. The log is com-

bined by a log key and parameter value vector. At the training stage, the log needs to first be parsed. Then the obtained event sequence can be used as the training input for the detection model. After training is completed, the system can judge whether the log is normal according to the log key. If it is normal, DeepLog will further check the parameter value vector. If the parameter value vector is anomalous, it will be marked as an anomalous log.

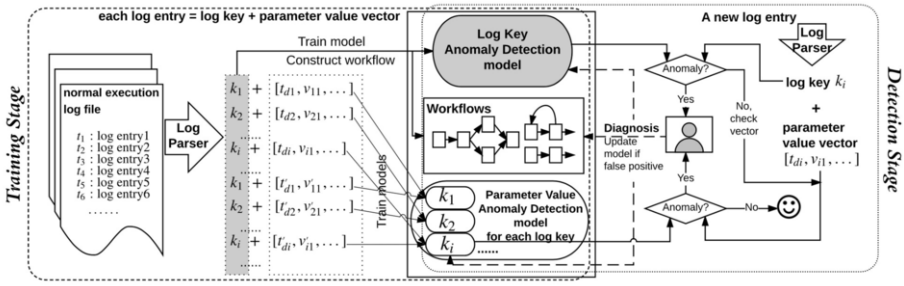

Figure 1. DeepLog architecture.

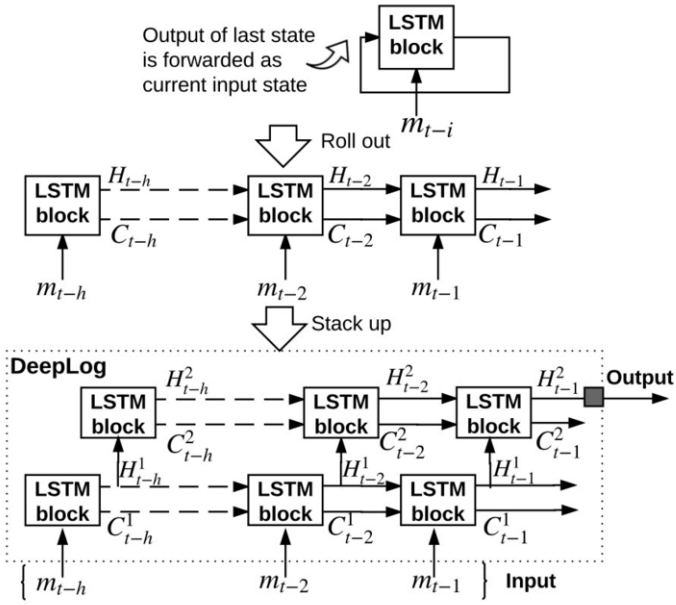

Figure 2. View of anomaly detection using stacked LSTM.

DeepLog uses Long Short-Term Memory (LSTM) in the recurrent neural network (RNN) as a framework to observe the long-term dependence of the sequence, figure 2. The top is single-nature LSTM block. Each LSTM block remembers the state for its input as a vector of fixed dimension. m_{t-i} is example of input. The center shows a series of LSTM blocks where a hidden vector H_{t-i} and state vector C_{t-i} are maintained in each cell. Both hidden vector and state vector will be passed to next as initial input to maintain

historical information. The bottom of figure 2 shows an example of DeepLog with two hidden layers of a deep neural network scheme. The input layer encodes n possible log keys from K as one-hot vectors. The output layer, which uses a standard multinomial logistic function, translates n layer of hidden state into a probability distribution function [23].

3. Shapley Value Approach for Explainable AI

3.1. Shapley Value in Game Theory

In the classical cooperative game theory it is assumed that players cooperate and together obtain a total reward and then distribute this reward between each other [27]. In order to share the total reward, the notion of imputation is introduced. In this paper as an imputation we use a classical cooperative solution Shapley value [10]. The explicit equation for the Shapley value is presented in Eq. (2) as below:

$$\varphi_i = \sum_{S|i\in S\subseteq N} \frac{(|S|-1)!(|N|-|S|)!}{|N|!}[v(S)-v(S\setminus\{i\})],\ i\in N, \tag{2}$$

where i is the number of players, N is the set of all players in the cooperative game, $S\subseteq N$ is a coalition of players, $|S|$ is the number of players in coalition S, $V(S)$ is a characteristic function of coalition $S\subseteq N$ that specifies the total payoff of coalition S.

In order to use the Shapley value in any domain it is necessary to calculate the values of characteristic function $V(S)$ for each coalition $S\subseteq N$. More details about how to calculate it for explaining anomaly detection in a log system will be presented in the following sections. In cooperative game theory, as can be seen from the Eq. (2), the Shapley value of player i shows the weighted sum of contributions of player i to the cooperation reward of each coalition S from N (term $[v(S)-v(S\setminus\{i\})]$). The left multiplier in the product (2) defines the probability that the coalition S itself will be formed, therefore the less the probability the less important is the individual contribution of player i to cooperation.

Using the Shapley value, the total reward is allocated among the players ($\sum_{i=1}^{n}\varphi_i = V(N)$). Generally speaking, if the contribution of the player to the cooperation is big, then his value of imputation is also big. In machine learning, the Shapley value approach can be applied to explain the contribution of each feature value to the overall solution.

3.2. Shapley Value for Decision Tree

The basic idea of Decision tree is to use a top-down recursive method to take information entropy as a measure to construct the fastest falling entropy value. The entropy value at the leaf node is zero. At this time, the instances in each leaf node belong to the same class. In other words, the essence of a Decision tree is a set of if-then rules. A Decision tree divides the feature space into disjoint units or regions.

In order to apply the approach based on Shapley value on the first step we need to calculate the values of the characteristic function for each coalition $S\subseteq N$, where N and S are the set and the subset of all features or unique events in the system log correspondingly. The meaning of characteristic function $V(S)$ of coalition S for an anomaly

detection system is the anomaly detection value or probability of anomaly based only on events from the set S. After calculating the characteristic function for each coalition S, it is possible to compute the Shapley value and, as a result, explain the result of Decision tree by explaining the contribution of each feature. The algorithm to calculate the Shapley value for the Decision tree anomaly detection is:

1. Define $n = |N|$ feature player from data set D.
2. Choose coalition $S \subseteq N$ of feature players. The total number of coalitions to consider is $\sum_{k=0}^{n} C_n^k$.
3. Run Decision tree algorithm for each coalition $S \subseteq N$ to obtain the value of characteristic function $V(S)$ or the accuracy of anomaly detection.
4. Repeat step $2-3$ until all values of characteristic function $V(S)$ are calculated.
5. Using the Eq. (2) calculate the Shapley value for all players.

3.3. Shapley Value for DeepLog

The DeepLog approach allows a user not only to find a feature or event that is the anomaly, but also to define a sequence of events that can lead to an anomaly in the system log. Therefore, Explainable AI should also address this issue by introducing the explaining not only for a set of individual events, but also for a sequence of events.

Here we also define the accuracy of anomaly prediction for the set of features S using the value characteristic function $V(S)$. But we consider 2 events and 1 target event as a feature or player, figure 3. All sequences will be divided and sorted by the target event. We will calculate the value of characteristic function $V(S)$ for sets S of sequence events in system log instead of the set of single events to verify the contribution of each sequence. The workflow of DeepLog is presented below:

1. Log analysis: The goal of log analysis is to convert unstructured log messages into structured event mapping, based on which complex machine learning models can be applied.
2. Feature extraction: usually structured logs can be cut into short log sequences through interval windows, sliding windows or conversation windows. In DeepLog, we use sliding windows for feature extraction, and vectorize each extracted sequence.
3. Anomaly detection: After the model is obtained through the training data, anomaly prediction is performed on the extracted data.

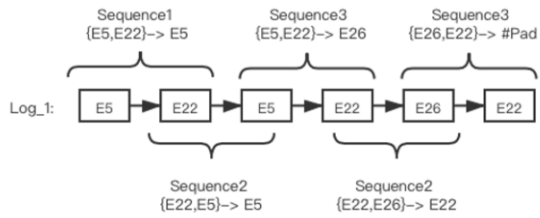

Figure 3. Example of sequence approach.

According to figure 3, short event sequences will be extracted from raw system long sequences to compose the set of features or players. This procedure is different from the

one in Decision tree algorithm. In DeeplLog algorithm, the sequence approach is considered to measure the contribution of each sequence to the anomaly detection. The sequence approach leads to the problem of a large number of coalitions or a curse of dimensionality. To avoid this problem we use a bi-level approach for calculating the Shapley value or sequence estimation for anomaly detection. On the first level, as a feature or player we consider only the target event, while on the second level for each fixed target event we consider the sequence of two events as a player. Therefore the number of players or features on the first level corresponds to the number of unique target events $E \in N$ in the system log and is equal to $|N|$. On the second level the number of features or players is various depending on the related target event E, and is equal to the number of unique sequences of two events $\{E_i, E_j\} \in N_E$ before the target event E ($|N_E|$), $E \in N$. Later for the first level the values of characteristic function $V(S)$, $S \subseteq N$ are calculated using the same approach as for the Decision tree algorithm. But for the second level the values of characteristic function $V_E(S)$, $S_E \subseteq N_E$ are calculated for each fixed target event $E \in N$. Later Shapley values are calculated using the Eq. (2) both for first level φ^t and for the second level with sequences of events φ_s. The resulting Shapley value for each sequence of events is obtained by the multiplication of Shapley values on the first and second levels. The algorithm to calculate the Shapley value for DeepLog anomaly detection is:

1. Using sliding window = 3 extract features (target events and corresponding two sequence events).
2. Define G_n as sequence player group, where n is number of unique target players.
3. chose i-th coalition, i from 0 to $\sum_{k=0}^{n} C_n^k$.
4. Run DeeplLog algorithm to obtain the accuracy of prediction.
5. Update the characteristic function for i-th coalition with accuracy prediction. i++, repeat step 3 to 5 till all characteristic function are obtained.
6. Use Eq. (2) to calculate the Shapley value for all players.

4. Simulation Results

In this section we present the simulation results of explaining the solution of pretrained Decision tree and DeepLog. For the Decision tree we use 2 data sets and for the DeepLog we use 1 data set of system logs. Simulations are run on the System: MacOS, CPU: 2.6 GHz, RAM: 8GB.

Link to data sets: https://github.com/logpai/loglizer/tree/master/data/HDFS

4.1. Simulation Results of Decision Tree

For the Decision tree we consider two input data sets. Both system logs consist of $n = 3900$ instances of the system log with 20 unique events. The initial system log is filtered to reduce the number of unique events to 14 because the other 6 appear rarely. Therefore, the total number of coalitions can be calculated in the following way: $\sum_{k=0}^{14} C_{14}^k = 16383$. The process of how to define players and coalitions using the initial system log is presented in figure 4. Using the Eq. (2), and values of characteristic function $V(S)$, we calculate the Shapley value. Below in figures 5 and 6 see the results for two test data sets. Here the contribution of each event of system log to the solution of Decision tree using the test data sets is defined.

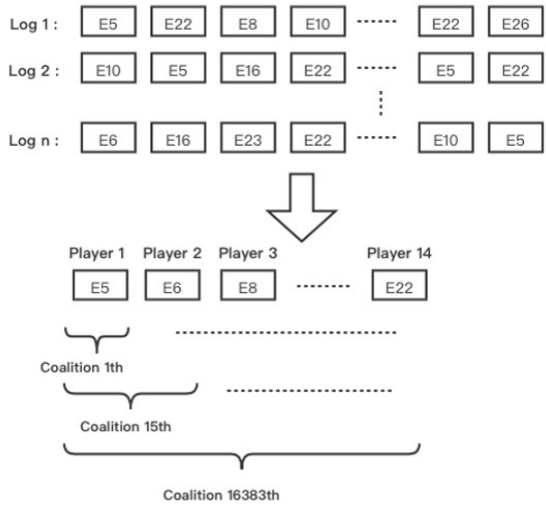

Figure 4. Connection between system log and components or cooperative game.

Figure 5 shows that the event $E11$ plays the most important role in the anomaly detection for data set 1 for Decision tree. The important conclusion for the input data set 1 is that, even if we remove all events apart from event $E11$ from the system log, then the anomaly detection would still be good because the Shapley value for event $E11$ is equal to 0.9636. In figure 6 note the simulation results for the input data set 2. Here events $E9$, $E11$ and $E26$ play an important role. This means that each of these events provoked an anomaly in the system log, Shapley values for events $E9$, $E11$ and $E26$ are equal to 0.322, 0.313, 0.317 correspondingly.

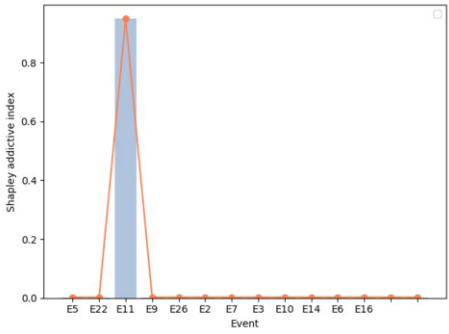

Figure 5. Shapley value with input data set 1.

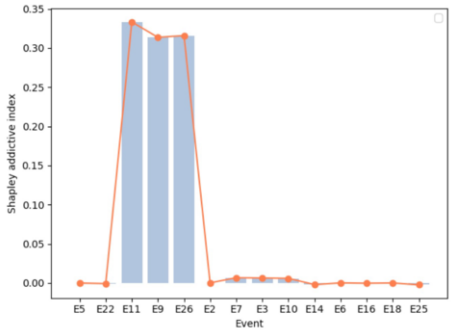

Figure 6. Shapley value with input data set 2.

4.2. Simulation Results of DeepLog

In figure 3 the process of bi-level method is shown. The window length for the second level of equal to 3.

After extracting features, we obtain a new data structure. The probability of an event after the sequence is the key to anomaly detection. Low probability means an anomaly

while a high probability means a normal sequence. Figure 7 also has element ”#Pad”, which means the end of the sequence. We can control whether players participate in the local level game by setting the event as Nan. On the second level all the sequences with the unique target event will be considered as players.

Test data set contains 1875 instances, that contain 15000 sequences of events. After cleanning the data, we obtain 7 unique events. In order to implement procedure described in secion 3.3 we consider two cooperative game models: first level game model for groups of events, second game model for sequences of events in each group.

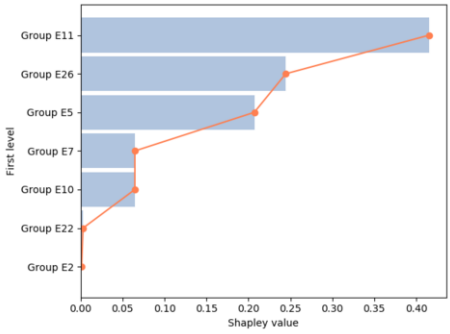

Figure 7. Shapley value for group players in the first level game.

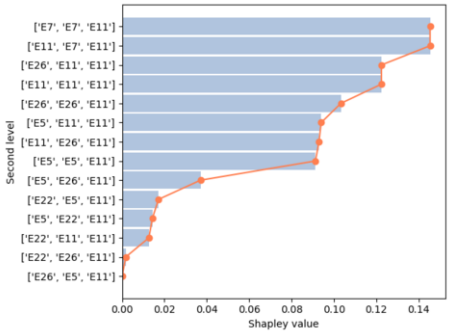

Figure 8. Shapley value for sequence players in the second level game for target event $E11$.

In the first level game model we define a player as a group of event sequences that correspond to a specific target event from the initial system log, see figure 3. Then we calculate the values of characteristic function and the Shapley value. It is important to notice that this is not yet the explaining result for the solution of DeepLog corresponding to the input data set. Figure 7 shows the contribution of each group of sequences corresponding to the target events: $E11$, $E26$, $E5$, $E7$, $E10$, $E22$, $E2$. It is easy to see that the contribution 0.415 to the result of DeepLog of the group $E11$ is the biggest, while the contribution of the group $E2$ is only 0.0011.

The second level game model is constructed separately for each group related to a target event. A player in the second level game is a sequence of events with the fixed target event. Using the Shapley value it is possible to define a contribution of each event sequence to the anomaly detection of event group. Figure 8 shows the contribution of each sequence of events to the anomaly detection for group of events related to the target event $E11$. Similar results are obtained for the target events $E2$, $E5$, $E7$, $E10$, $E22$, $E26$.

Finally when the Shapley values both for the first and the second level game models (figures 7 and 8) are obtained the Shapley value for each sequence of events is calculated using the apporach described in section 3.3. The meanning of this Shapley value is the contribution of sequence of events to the anomaly detection of DeepLog using the test data set. Figure 9 shows the contribution of each sequence of events to the anomaly detection of DeepLog using the test data set. It is easy to see sequences $\{E7, E7, E11\}$, $\ldots$, $\{E5, E5, E11\}$ have the biggest contribution to the anomaly detection. This means that the anomaly of the system log from test data set is concentrated in these sequences. The user of DeepLog system should check in detail what is the nature of these sequences to make a conclusion.

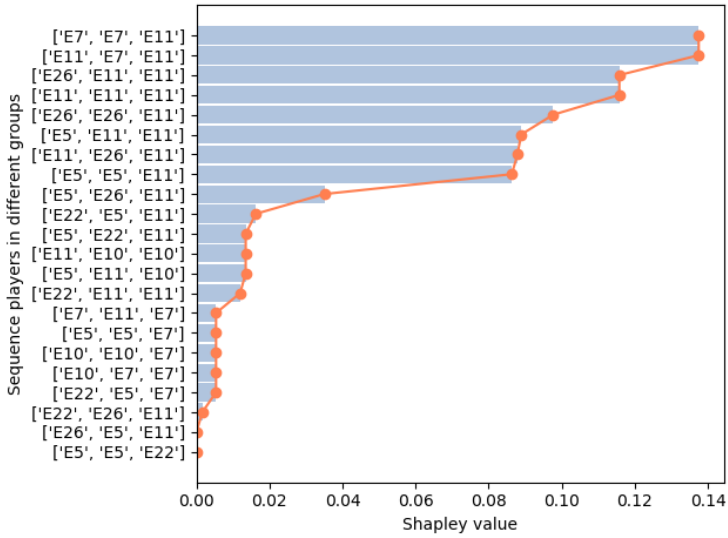

Figure 9. Shapley value for the result of anomaly detection using DeepLog with test data set.

5. Conclusion and Future work

At the current stage we are applying the Shapley value approach to static machine learning modeling, in particular to the Decision tree and DeepLog algorithms. As shown in the simulation section our approach can explain a specific solution using both single feature and sequence events. As it is indicated in figure 9, the user of anomaly detection system that has the explainable module could easily find the reason of the anomaly (specific event or sequence of events), not only the receive the anomaly alarm.

In the future, we will apply several sampling methods presented in the papers [28], [29], [30] to approximate the Shapley value. This would aid in evaluating the contribution for each unique sequence instead of using a bi-level method. And think about indicators to evaluate the precision and reliability of our approach. Besides, we plan to optimize the algorithm to decrease the time consumption and be comparable with well known framework SHAP [9].

6. Acknowledgement

The work of the second author is supported by Russian Foundation for Basic Research (RFBR) according to the research project No. 18-00-00727 (18-00-00725).

References

[1] Lundberg SM et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nature Biomed. Engin., 2.10 (2018): 749-760.

[2] Holzinger A et al. What do we need to build explainable AI systems for the medical domain? arXiv preprint, arXiv:1712.09923. – 2017.

[3] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. ACM Computing Surveys, 2009, 41(3).

[4] Hawkins D. Identification of outliers. Springer Netherlands, 1980, P. 188.

[5] Chalapathy R, Chawla S. Deep learning for anomaly detection: a survey, arXiv: Learning, 2019.

[6] Tkachenko R, Izonin I. Model and principles for the implementation of neural-like structures based on geometric data transformations. Adv Intell Syst Comput 754: 578–587.

[7] Izonin I, Tkachenko R, Kryvinska N, Tkachenko P. Multiple linear regression based on coefficients identification using non-iterative SGTM Neural-Like Structure. In International Work-Conference on Artificial Neural Networks, Springer, Cham, 2019 June, pp: 467-497.

[8] Tkachenko R, Izonin I, Vitynskyi P, Lotoshynska N, Pavlyuk O. Development of the noniterative supervised learning predictor based on the ITO decomposition and SGTM neural-like structure for managing medical insurance costs. Data, 2018, 3(4), 46.

[9] Antwarg L, Shapira B. Explaining anomalies detected by autoencoders using SHAP. arXiv preprint, arXiv:1903.02407. – 2019.

[10] Shapley LS. (August 21, 1951). Notes on the n-Person Game – II: The Value of an n-Person Game. Santa Monica, Calif.: RAND Corporation.

[11] Leon A P, Nikolay A Z. Game Theory (2nd Edition), World Scientific, 2016.

[12] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Neural Inform. Processing Syst., 2017, pp. 4765–4774.

[13] Sundararajan M, Najmi A. The many shapley values for model explanation. arXiv preprint, arXiv: 1908.08474, 2019.

[14] Aas K, Jullum M, Løland A. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. arXiv preprint, arXiv:1903.10464, 2019.

[15] Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee SI. From local explanations to global understanding with explainable AI for trees. Nature machine intelligence, 2020, 2(1): 2522–5839.

[16] Vega Garcıa M, Aznarte JL. Shapley additive explanations for NO2 forecasting. Ecol. Inform., Mar 2020, 56: 101039.

[17] Ribeiro MT, Singh S, Guestrin C. Why should I trust you? Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, – 2016. – . 1135-1144.

[18] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences //arXiv preprint arXiv:1704.02685. – 2017.

[19] Montavon G et al. Layer-wise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning. – Springer, Cham, 2019. – . 193-209.

[20] Arun D, Paul R. Opportunities and challenges in explainable artificial intelligence (XAI): a survey, arXiv, 2020.

[21] Chen M, Zheng AX, Lloyd J, Jordan MI, Brewer E. Failure diagnosis using decision trees. International Conference on Autonomic Computing, 2004. Proceedings., New York, NY, USA, 2004, pp. 36-43, doi: 10.1109/ICAC.2004.1301345.

[22] Liang YL, Zhang YY, Xiong H, Sahoo R. Failure Prediction in IBM BlueGene/L Event Logs.

[23] Anomaly detection and diagnosis from system logs through deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, 1285-1298.

[24] Xu W, Huang L, Fox A, Patterson D, Jordan MI. Large-Scale System Problems Detection by Mining Console Logs.

[25] Nikolay VK. Strategic stability of coalitions technological alliance parameters: a two-level cooperation. Contributions to Game Theory and Management, 2015, Volume 8, 111–136

[26] He SL, Zhu JM, He PJ, Michael RL. Experience report: system log analysis for anomaly detection. IEEE International Symposium on Software Reliability Engineering (ISSRE), 2016.

[27] Shapley L. A value for n-person games. Contributions to the Theory of Games. 1953. 2(1): 307–317.

[28] Mann I, Shapley LS. Values of large games 6: Evaluating the electoral college exactly. Tech. Rep., Rand Corp Santa Monica CA (1962).
[29] Castro J, Gomez D, Tejada J. Polynomial calculation of the shapley value based on sampling. Comput. Oper. Res. 2009, 36: 1726–1730.
[30] Maleki S, Tran-Thanh L, Hines G, Rahwan T, Rogers A. Bounding the estimation error of sampling based shapley value approximation. arXiv preprint, arXiv, 2013, 1306.4265.