ANALYSIS OF THE PROBLEM OF MULTIVALUED OF CLASS LABELS ON THE SECURITY OF COMPUTER NETWORKS

D. I. Rakovskiy,
Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russia
Prophet_alpha@mail.ru

DOI: 10.36724/2664-066X-2022-8-6-10-17

ABSTRACT

Modern computer networks have a complex infrastructure that requires constant monitoring to detect anomalous conditions that can cause malfunctions, which is unacceptable for large-scale distributed networks. An important problem in the intelligent processing of syslog data is the existence of multi-label datasets. Among the Russian-language scientific publications, the problem under consideration in the context of information security of computer networks is not presented. The purpose of the research work is to increase the security of computer networks through the use of multi-label learning methods in solving the problem of classifying system log class labels. In this paper, a comparative analysis of single-label and multi-label classifiers in a computational experiment on the Mean accuracy metric was carried out. According to the results of the analysis, 80% of single-label classifiers were inferior in classification accuracy according to the Mean accuracy multi-label metric to their counterparts, which may indicate a strong influence of multi-label class labels on the models under consideration. The considered structure of experimental data in a tabular form is influenced by the multi-label problem much more strongly than it can be estimated by a standard frequency check, which actualizes further research in this direction.

KEYWORDS: *supervised learning, multi-label classification, multiclass classification, information security, multi-label learning*

I. Introduction

Modern computer networks (CNs) have a complex infrastructure that requires constant monitoring to detect anomalous conditions that can cause malfunctions, which is unacceptable for large-scale distributed CNs [1].

CN security can be achieved through the use of classical measures to prevent traffic interception – the installation of software and hardware information protection [2]; intrusion detection and prevention systems [3, 4]; antivirus software [5] and other solutions [6].

Works devoted to the development of software solutions for detecting and leveling cyber threats in CN are relevant [7]. Known works on the estimation and prediction of the state of complex objects: application for information security [8, 9].

An important problem in the intelligent processing of syslog data is the existence of datasets containing entries with multiple class label associations. That is, the class associated with an object is characterized by a set of labels.

A dataset suitable for classification typically contains a set of features and an associated set of class labels. The goal of classification is a trained model capable of assigning an appropriate class to an unknown object (records in "historical data").

Works, one way or another, exploring the problems of multi-label, are united by the term: Multi-Label Learning, MLL [10]. MLL generalizes the notion of data analysis to the realm of tasks, in which multiple labels can be associated with each object. Among these articles, a cluster of works on the analysis of text corpora [11] and the tone of messages in social networks [12] stands out.

Domestic works devoted to the analysis of data sets generated by CN with multi-label and class labels are currently not presented. Existing works, for example, [13, 14], are devoted to aspects of fuzzy classification. Fuzzy classification belongs to the field of fuzzy logic, which is part of the multi-class learning methods.

MLL is indirectly related to the concept of Misclassification. The term is currently used to label works devoted to solving problems of data mislabeling [15] and improving classification accuracy [16].

Information security characterizes the preservation of the properties of confidentiality, integrity and availability of information [17]. It should be noted that the analysis of the impact of multi-label class labels on CN security must be carried out in a certain terminological context. GOST R ISO/IEC 27000, from which the above definition of information security derives, as well as GOST R ISO/IEC 12207, was chosen as such a context. According to the referenced document, paragraph 3.25, "Security: The ability of a computer system to protect information and data so as to prevent their unauthorized reading or modification by other systems and individuals, and so that systems and individuals admitted to them do not receive failures."

It is necessary to clarify the security of information circulating in the CN: “The security of information is the maintenance at a given level of those parameters of the information located in the automated system that characterize the established status of its storage, processing and use” [18]. It follows from the two definitions that the security of information circulating in the CN is related to the security of the supporting infrastructure [19].

Within the framework of this work, we will analyze the influence of multi-label on the accuracy of the classification of CN states that are directly related to the profile of the normal functioning of CN [20].

The aim of the work is to increase the security of computer networks through the use of multi-label learning methods in solving the problem of classifying system log class labels.

II. Generation of CN class labels

The CN can be represented as a set of M sets of values of discretely changing attributes of "historical data" of the CN:

$$\boldsymbol{A} \subseteq A_{first} \cup A_{second} = \{A_{first\,1} \times A_{first\,2} \times ... \times A_{first\,len_1}\} \cup \cup \{A_{second\,1} \times A_{second\,2} \times ... \times A_{second\,len_2}\}; \quad (1)$$

In this equations, $A_m = \{a_{mn}; m = \overline{1,M}, n = \overline{1,N}\}$, $A_m \subset \boldsymbol{A}$, $M = len_1 + len_2$.

Attributes in (1) can be divided into two types: primary $\{A_{first\,k_1}; k_1 = \overline{1, len_1}\}$ and secondary $\{A_{second\,k_2}; k_2 = \overline{1, len_2}\}$.

The primary attributes are obtained directly from the system sensors installed inside the CN. Secondary attributes are obtained as a result of processing primary attributes. Examples of secondary attributes can be, for example, the average signal delay time in the CN, the number of lost packets in the CN for a particular host, and so on.

To describe CN, we introduce a set of class labels of categorical type - S, which we will call “CN states”. The CN states can also be entered as a set:

$$\boldsymbol{S} = \{S_1, S_2, ..., S_M\} \cup \{s_{normal}\}; S_m = \{s_i; i = \overline{1, I}\} \quad (2)$$

where S_m – m-th subset of the CN states associated with the corresponding A_m attribute of CN. Power of the subset S_m has an upper bound equal to I. In practice, the subsets included in S may have different powers. An example of elements with different power is the inequality $|S_1| \neq |S_2|$.

In the case of $\forall S_m = \varnothing$ status s_{normal} is entered, characterizing the normal functioning of the CN.

To automate the process of determining the states of the CN we introduce a set of rules

$$\begin{aligned} &\boldsymbol{METARULES} = \{RULE_1, RULE_2, ..., RULE_M\}, \\ &RULE_m = \{r_{mj}; j = \overline{1, |S_m|}\} \end{aligned} \quad (3)$$

Each subset - $RULE_m$ is associated with the corresponding subset of CN states by the m-th attribute S_m. The power of a subset $RULE_m$ depends on the power of the corresponding subset S_m. The iteration variable j is introduced to account for the difference in power of different subsets S_m. If all subsets are identical S_m, $j = \overline{1, |S_m|} \equiv i = \overline{1, I}$, the upper boundary will be identical to I.

Decisive rules are proposed to be selected based on the individually entered Service Level Objectives, SLO, based on the technical and operational characteristics of the CN.

Consider the process of labeling a set of attributes corresponding to n observations of historical data (n rows in the table of historical data) – $\{a_{1n}, a_{2n}, ..., a_{Mn}\}$. The specified string is an argument of the labeling function $mark(\{a_{1n}, a_{2n}, ..., a_{Mn}\})$, and forms a set of labels set_n, corresponding to the n-th line.

The set_n is formed by checking each attribute of the n string $\{a_{1n}, a_{2n}, ..., a_{Mn}\}$ – for compliance with the rules of the corresponding set $RULE_m$ (3) – r_{mj}. If the rule r_{mj} is fulfilled, then in the set of labels set_n element s_{mj} is added, where $j = \overline{1, |S_m|}$.

The labeling process can be formalized as:

$$mark : \{a_{1n}, a_{2n}, ..., a_{Mn}\} \to set_n;\ set_n \subseteq \boldsymbol{S}, \text{ where}$$

$$mark(\{a_{1n}, a_{2n}, ..., a_{Mn}\}) = \begin{cases} set_n, \text{ if } set_n \neq \varnothing \\ s_{normal}, \text{ otherwise} \end{cases},$$

$$\text{where } set_n = \left\{ \begin{matrix} s_{mj} \in S_m \mid r(a_{mn}, j) = 1, \\ j = \overline{1, |S_m|},\ m = \overline{1, M} \end{matrix} \right\}, \quad (4)$$

$$\text{where } r(a_{mn}, j) = \begin{cases} 1, \text{ if rule } r_{mj} \in RULE_m \text{ is followed} \\ 0, otherwise \end{cases}$$

If none of the rules are satisfied, then $set_n = \left\{ s_{mj} \in S_m \mid r(a_{mn}, j) = 1,\ j = \overline{1, |S_m|}, m = \overline{1, M} \right\} = \varnothing$.

This means that the result of marking will be a predetermined state of the CN - s_{normal}. Each item r_{mj}, is a freely defined verbal-logical rule introduced for a particular CN.

The rules can be paired with the security policy relevant for the CN: with the threat model; with the SLO service level indicators; with other methods of security and service quality assessment.

When marking rules affect the data of the CN, each record (string – $\{a_{1n}, a_{2n}, ..., a_{Mn}\}$) is assigned either a set of states set_n, according to relation (4), or the state s_{normal}.

The labeling of "historical data" about the behavior of the CN can be represented as a table of size M columns by N rows:

$$D_N = \{(\{a_{1n}, a_{2n}, ..., a_{Mn}\}, set_n); m = \overline{1, M}, n = \overline{1, N}\},$$

where the n-th row of record attribute values $\{a_{1n}, a_{2n}, ..., a_{Mn}\}$ the state of the CN and the set of labels set_n.

Although not the only way of marking experimental data but is the most convenient in terms of organizing the processing and analysis of data by specialized software tools.

III. Structure and description of the studied network infrastructure

Research to evaluate network performance was carried out on a CN consisting of 6 hosts forming a cluster managed by Rancher (Fig. 1) [21]. The host interaction architecture of the studied CN is based on the principle of virtualization and interaction between Docker containers; services managed by an Apache Spark cluster; databases (PostgreSQL; Apache Ignite; Apache Cassandra; Redis); Apache Ignite cluster; software based on microservice architecture and other auxiliary modules.

Technical characteristics of distributed CN host machines are given in Table. 1. Machines #1 - #3 form the physical topology of the distributed CN; machines No. 4-6 operate through virtualization by the VMware ESXI operating system based on machines No. 1-3.

To collect data on 6 CN machines, special software for obtaining information from system sensors was used: *packetbeat* (aggregates HTTP and DNS request protocol traffic); *metricbeat* (aggregates data on CPU usage, disk usage, memory usage, network usage, system processes); *filebeat* (aggregates message log data); *execbeat* (aggregates the execution of specialized scripts and sending the result of their execution).

To collect indicators related to SLO, the CN under consideration implements a system for synchronous monitoring of all hosts. The scheme for collecting indicators is shown in Figure 2. The received data is aggregated in a centralized storage managed by Apache Cassandra.

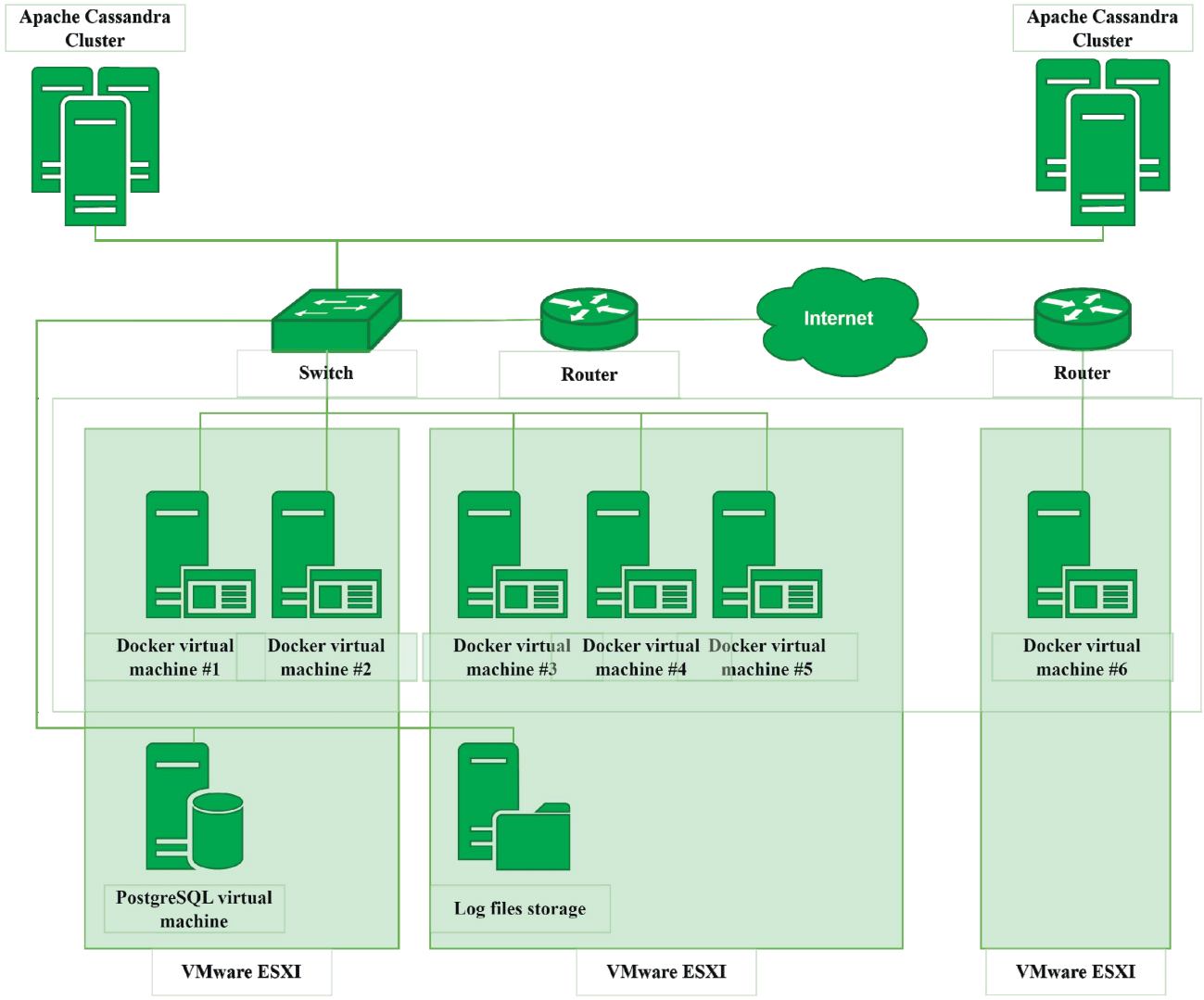

Fig. 1. Scheme of the studied network infrastructure

Table 1

CN configuration

No	Host mapping to a virtual machine	Operating system	Number of Cores	RAM, GB	Hard disk capacity (total), GB	Processor Model
1	server3-20 (physical machine from the cluster Apache Cassandra)	CentOS Linux 7	4	64	1524	Intel(R) Xeon(R) CPU E3-1220 v6 @ 3.00GHz
2	server3-21 (physical machine from the cluster Apache Cassandra)	CentOS Linux 7	4	64	1524	Intel(R) Xeon(R) CPU E3-1220 v6 @ 3.00GHz
3	server3-22 (physical machine from the cluster Apache Cassandra)	CentOS Linux 7	4	64	1524	Intel(R) Xeon(R) CPU E3-1220 v6 @ 3.00GHz
4	server24- 384-1 (Docker virtual machine №1; Docker virtual machine №2)	Ubuntu 18.04.1 LTS	5	50,05	68	Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz
5	server24- 384-2 (Docker virtual machine №3; Docker virtual machine №4; Docker virtual machine №5)	Ubuntu 18.04.1 LTS	6	48,61	265	Intel(R) Xeon(R) CPU E5-2420 v2 @ 2.20GHz
6	server24- 384-3 (Docker virtual machine №6)	Ubuntu 18.04.1 LTS	8	60	285	Intel(R) Xeon(R) CPU E5-2420 v2 @ 2.20GHz

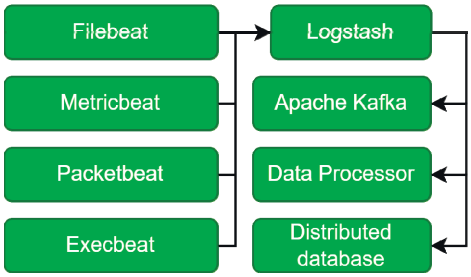

Fig. 2. Diagram of using the SLO-related metrics aggregation system

On the hosts given in Table 2, in accordance with the diagram in Figure 2, aggregators of software and hardware metrics are installed: *Packetbeat* (host network activity aggregator – traffic monitoring, HTTP protocol and DNS queries) [22]; *Metricbeat* – an aggregator of indicators associated with the operating system and host hardware devices – the use of CPU, memory, disks, running processes; *Filebeat* is a syslog aggregator; *Execbeat* – software for CN tests by generating and sending arbitrary scripts for execution. *Execbeat* was used to send ICMP requests (ping requests) to determine network latency and send GET requests using it to determine the server's response time to a sent request.

Each of the four types of aggregators sends data to the central point CN – the *Logstash* log aggregator, which converts all incoming information into JSON files. The choice of format is due to the generally accepted notation of the structure of the structure of JSON files. The described stack of aggregators is widely used in the construction of information processing systems in the field of information security [23-26].

After conversion, the JSON file is sent to the Apache Kafka message handler, which performs a buffering function between the large input stream and the distributed database. Hyperlinks to said software are provided in footnote [1].

IV. The problem of primary (first) and secondary (second) attributes

Let us consider an illustrative example of the results of single-valued classifiers - binary and multi-class - with multi-label experimental data on the basis of the results given in [15]. A fragment of this data is given in Table I.

An actual applied problem is the determination of CN states without knowledge of secondary attributes. In this case, the labels of the SLO classes are determined only on the basis of the primary data of the system sensors under conditions of partial uncertainty of the remaining parameters.

Consider two cases:

1. Complete a priori certainty of both primary and secondary attributes of CN at each moment of time;

2. Partial uncertainty of CN secondary attributes that are either unknown or computed with a long delay.

With full attribute information (A_{first} and A_{second},), due to total dependency set_n on A_{second}, the task of classifying the CN state is performed by a multi-label classifier with an accuracy close to ideal, i.e. without mistakes. An obstacle to such an ideal classification is the identification of direct transformation rules A_{second} to set_n ($A_{second} \rightarrow set_n$).

If the rules are represented by trivial logical conditions "if … then …", then the classification accuracy of many rule-based classifiers (for example, decision trees or neural networks) will be close to ideal. If the secondary attributes are unknown, but the primary attributes and the corresponding CN states are known, the secondary attributes will be a latent variable. In the absence of information about secondary attributes, the single-label mapping of primary attributes to CN states is not guaranteed, since secondary attributes become latent variables. However, the fundamental possibility of displaying primary attributes in CN states is still possible.

V. Computational experiment

To compare two classification methods - the "classic" single-label and multi-label - let's conduct a computational experiment in Python with the following input data. It is proposed to consider the single-label approach to classification using the example of multiclass algorithms selected according to two criteria:

– openness of the source code (the library that implements this algorithm is in the public domain);

– availability of multi-label implementation of this algorithm. According to the established criteria, the following algorithms were selected from the open scikit-learn library of the Python programming language [27]:

Tree.DecisionTreeClassifier – classifier generated on the basis of the "Decision Tree" algorithm (non-parametric supervised learning method);

Tree.ExtraTreeClassifier – A classifier generated on the basis of the "Extra Decision Tree" algorithm (non-parametric supervised learning method). When searching for the best split to split the node samples into two groups, for each of the randomly selected attributes, the best split is selected according to the specified criterion;

Ensemble.ExtraTreesClassifier – classifier generated on the basis of the "Extra Decision Tree" algorithm (ensemble implementation);

Neighbors.KNeighborsClassifier – Classifier generated based on the voting algorithm "K-Neighbors";

[1] - PACKETBEAT. Lightweight shipper for network data // Elastic URL: https://www.elastic.co/beats/packetbeat
- METRICBEAT. Lightweight shipper for metrics. // Elastic URL: https://www.elastic.co/beats/metricbeat
- FILEBEAT. Lightweight shipper for logs. // ElasticURL: https://www.elastic.co/beats/filebeat
- Elastic beat to call commands in a regular interval and send the result to Logstash // Elasticsearch URL: https://github.com/christiangalsterer/execbeat
- Apache Kafka. A distributed streaming platform. // Apache Kafka URL: https://kafka.apache.org/

Ensemble.RandomForestClassifier - A classifier generated on the basis of the "Random Forest" algorithm (ensemble implementation).

The indicators of the SLO service level (decision rules) and the corresponding CN states associated with secondary attributes are obtained, formed in the form of thresholds that determine the categorical markers CN states:

– If none of the service level objectives has been violated, then the CN state is equal to the normal marker.
– If the signal delay time to the test server (*ping_avg*) > 5 ms, then CN state is equal to the *signal_delay* marker.

– If the response time of the test server (*server_response_timetotal*) > 1.5 s, then the CN state is equal to the *server_response_delay* marker.

- If the number of packets lost during transmission to the test server (*network_outdropped*) > 0, then the CN state is equal to the *packets_dropped* marker.

– If the time to process a request by the disk of the host machine (*disk_ioreadmergespersec*) > 2 s, then the CN state is equal to the *disk_iowriteawait* marker.

The number of decision rules and the CN attributes affected by them can be increased depending on the task, but 5 class labels are sufficient for illustration.

Consider the distribution of experimental data over the number of simultaneously violated service level indicators. The initial distribution is shown in Figure 3.

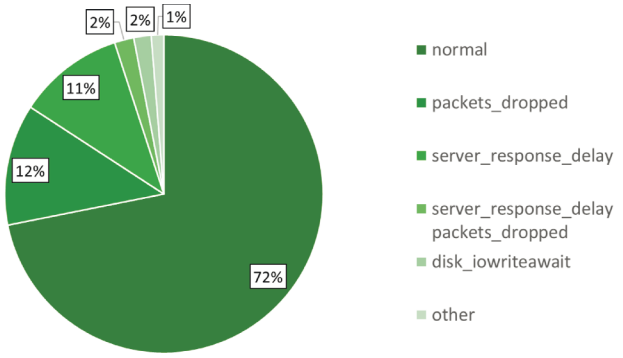

Fig. 3. Distribution of experimental data by the number of simultaneously violated service level indicators

As can be seen from the diagram, 72% of the experimental data is occupied by the CN state of the normal functioning of the CN, which gives rise to the problem of class imbalance. For the computational experiment, the first 200 thousand records of the initial experimental data were taken [21]. The amount of experimental data was chosen based on the available computing resources.

The specified attributes – *ping_avg, network_outdropped, disk_ioreadmergespersec, server_response_timetotal* – are converted to the corresponding CN states and excluded from further analysis. Thus, the specified secondary attributes CN become latent variables.

The following attributes are chosen as primary attributes for illustrative purposes: *disk_await, disk_writebytes, network_outbytes, network_inbytes, ping_max*.

Since one record can be associated with several CN states at the same time, the method of reducing multi-label class labels to a single-label form was chosen - *Label Powerset* (LP, [28]), generates a new class for each possible combination of labels by unitary coding of the alphabet of all possible combinations of CN states and then solves the multi-label analysis problem as a single-label multi-class analysis problem.

To improve the objectivity of the classification, the accuracy was assessed by cross-validation: the sample was divided into 10 equal parts; alternately one of the parts became a test. The classification efficiency metric is Mean accuracy (which is the standard metric for all algorithms provided by the scikit-learn.org library).

The experiment was carried out with standard hyperparameters set for the default algorithms. Hyperparameter optimization was not performed. For pairs "single-label classification algorithm X – multi-label classification algorithm X", the same hyperparameters were set.

The results of the computational experiment are given in Table 2. The table shows the name of the algorithm, the results for the single-label and multi-label cases. The cell with the highest value of the Mean accuracy metric among all types of classification is highlighted in light color.

Table 2

Comparative analysis of single-label and multi-label classifiers in a computational experiment

Name of the classification algorithm	Mean accuracy metric value for single-label case of multiclass classification	Mean accuracy metric value for multi-label classification case
Tree.DecisionTreeClassifier	0,52	0,75
Tree.ExtraTreeClassifier	0,66	0,69
Ensemble.ExtraTreesClassifier	0,64	0,81
Neighbors.KNeighborsClassifier	0,64	0,91
Ensemble.RandomForestClassifier	0,70	0,13

As can be seen from the table, 80% of single-label classifiers were inferior in classification accuracy according to the Mean accuracy multi-label metric to their counterparts, which may indicate a strong influence of multi-label class labels on the models under consideration. Despite the fact that multi-label plots are only 3% (see Table 2), the gain in accuracy reaches 23% in terms of the Mean accuracy metric for MLL algorithms.

The conducted experiment allows us to form the following conclusions. The LP method used to markup single-label data leads to high classification errors for boosting algorithms during cross-validation.

The data structure of [21] is affected by the multi-label problem much more than can be estimated by the standard frequency check performed in Table 1, 2. One of the possible reasons for such a strong influence is the use of primary attributes as arguments that are not directly related to the classified CN states.

Since the predictive power of frequency testing of the effect of multi-label class label results on the classification results of single-label classifiers is low, further research on this topic is planned. Conducting research in the field of multi-label analysis can lead to an increase in the accuracy of both static and dynamic fault detection in CN and network attacks [29].

Conclusion

The results of the study of estimating the characteristics of CN states of a distributed computer system consisting of six hosts for given indicators of the service level SLO are analyzed.

Class labels (CN states) generated as a result of CN operation, in the general case, are multi-label as a result of the removal and analysis of information on several CN attributes (from several system sensors). The nature of multi-label CN states is different from the nature of multi-label occurrence in the analysis of text corpora or social network data.

Anomalies associated with violation of the established SLO thresholds regularly occur simultaneously for several analyzed attributes. The results of the computational analysis made it possible to judge the nonlinear dependence of the frequency distribution of multi-label class labels on the degree of influence of multi-label on the classification results, which, in turn, directly affects the security of information circulating in the CN.

In connection with the results obtained, if there is a priority in the classification of certain class labels (which is important for information security tasks), multi-label classifiers are proposed for use.

References

[1] A. Kuznetsov, V. Babenko, K. Kuznetsova, S. Kavun, O. Smirnov, O. Nakisko "Malware correlation monitoring in computer networks of promising smart grids", *Proceedings of the IEEE 6th International Conference on Energy Smart Systems, ESS 2019*, 2019, pp. 347-352. DOI: 10.1109/ESS.2019.8764228

[2] A.S. Bol'shakov, D.I. Rakovskii "An efficient multiple-criteria decision analysismethod in the field of information security", *Legal Informatics*, 2020, no 4. pp. 55-66. DOI 10.21681/1994-1404-2020-4-55-66.

[3] I.V. Kotenko, S.S. Khmyrov "Analysis of models and techniques used for attribution of cyber security violators in the implementation of targeted attacks", *Voprosy kiberbezopasnosti*, 2022, vol 50, no 4, pp. 52-79. DOI 10.21681/2311-3456-2022-4-52-79.

[4] D.A. Gaifulina, I.V. Kotenko "Application of deep learning methods in cybersecurity tasks", *Voprosy kiberbezopasnosti*, 2020, vol 37, no 3. pp. 76-86. DOI 10.21681/2311-3456-2020-03-76-86.

[5] M. Alrammal, M. Naveed, S. Rihawi "Using heuristic approach to build anti-malware", Proceedings of the ITT 2018 - Information Technology Trends: Emerging Technologies for Artificial Intelligence. 5, *Emerging Technologies for Artificial Intelligence*, 2019, pp. 191-196. DOI: 10.1109/CTIT.2018.8649499.

[6] A.S. Bol'shakov, D.I. Rakovskii "Software for modelling information security threats in information systems", *Pravovaya informatika*, 2020, no 1, pp. 26—39. DOI: 10.21681/1994-1404-2020-1-26-39. E.Y. Pavlenko, N.V. Gololobov, D.S. Lavrova, A.V. "Kozachok Recognition of cyber threats on the adaptive network topology of large-scale systems based on a recurrent neural network", *Voprosy kiberbezopasnosti*, 2022, vol. 52, no 6, pp. 93 – 98. DOI:10.21681/2311-3456-2022-6-93-99

[7] K.E. Izrailov, M.V. Buinevich, I.V. Kotenko, V.A. "Desnitsky Assessment and prediction of the complex objectsstate: applicatioin for information security", *Voprosy kiberbezopasnosti*, 2022, vol 52, no 6, pp. 2 – 21. DOI:10.21681/23113456-6-2022-2-21

[8] O.I. Sheluhin, A.V. Osin, D.I. Rakovsky "New Algorithm for Predicting the States of a Computer Network Using Multivalued Dependencies", *Automatic Control and Computer Sciences*, 2023, vol. 57, no 1, pp. 48–60. DOI: 10.3103/S0146411623010091

[9] E. Gibaja, S. Ventura "A Tutorial on Multi-Label Learning", *ACM Computing surveys*, 2015, no 47, pp. 1-40. DOI: 10.1145/2716262

[10] A.C.E.S. Lima, L.N. de Castro "A multi-label, semi-supervised classification approach applied to personality prediction in social media", *Neural Networks*, 2014, vol. 58, pp. 122-130.

[11] S.N. Karpovich "Multi-Label Classification of Text Documents using Probabilistic Topic Model ml-PLSI", *Trudy SPIIRAN*, 2016, vol 47, no 4, pp. 92-104 DOI: 10.15622/sp.47.5

[12] I.V. Kotenko, I.B. Saenko, A.A. Branitsky, I.B. Paraschuk, D.A. Gayfulina "Intelligent system of analytical processing of digital network content for its protection from unwanted information", *Informatics and automation*, 2021, vol. 20, no 4, pp. 755-784

[13] G.G. Kulikov, V.V. Antonov, Antonov D.V. "Analysis of the possibility of analytical knowledge extraction of a formal model of subject domain information system by neural network methods", *Neurocomputers,* 2013, no 3, pp. 12-16.

[14] M. Azad, M. Moshkov "A Bi-criteria Optimization Model for Adjusting the Decision Tree Parameters", *Kuwait Journal of Science*, 2022, vol. 49, no 2, pp. 1–14. DOI 10.48129/kjs.10725

[15] A. Niemistö , O. Yli-Harja, I. Shmulevich, V.V. Lukin, A.N. Dolia "Correction of misclassifications using a proximity-based estimation method", *Eurasip Journal on Applied Signal Processing*, vol. 2004, no 8, pp. 1142-1155. DOI: 10.1155/S1110865704402145

[16] A.S. Markov "Cybersecurity and information security as nomenclature bifurcation scientific specialties (Russian text)", *Voprosy kiberbezopasnosti*, 2022, vol 47, no 1, pp. 2-9. DOI 10.21681/2311-3456-2022-1-2-9

[17] Lovtsov D. "Principles of ensuring information security in ergasystems", *Legal Informatics*, 2021, no 1, pp. 36-50. DOI 10.21681/1994-1404-2021-1-36-50

[18] A. S. Bolshakov, R. V. Khusainov, A.V. Osin "Traffic anomaly detection using a neural network to ensure information protection", *I-methods*, 2021, vol. 13, no 4, pp. 1 – 15.

[19] O.I. Sheluhin, D.I. Rakovskiy "Prediction of the profile functioning of a computer system (network) based on multivalued patterns", *Voprosy kiberbezopasnosti*, 2022, no 6, pp. 28-45 (in Russian) DOI:10.21681/2311-3456-2022-6-53-70

[20] O.I. Sheluhin, D.I. Rakovsky "Selection of metric and categorical attributes of rare anomalous events in a computer system using data mining methods", T-Comm. 2021, vol. 15, no. 6, pp. 40-47. (in Russian) DOI: 10.36724/2072-8735-2021-15-6-40-47

[21] B. Raja, K. Ravindranath, B. "Jayanag Monitoring and analysing anomaly activities in a network using packetbeat", International Journal of Innovative Technology and Exploring Engineering, 2019, Vol. 8, No 6, Pp. 45-49.

[22] I.V. Kotenko, A.A. Kuleshov, I.A. Ushakova “System for collecting, storing and processingsecurity information and events based on elasticstack tools”, Informatics and Automation (SPIIRAS Proceedings), 2017, vol. 54, no 5, pp. 5-34. DOI 10.15622/sp.54.1(in Russian)

[23] V.V. Petrov, K.V. Bryukhanov, E.Y. Avksentieva “Network monitoring: network traffic analysis using ELK”, In Modern Science: actual problems of theory & practice, 2020, no 5, pp. 102-105. DOI 10.37882/2223-2966.2020.05.34. (in Russian)

[24] G. Calderon, G. Del Campo, E. Saavedra, A. Santamaria “Management and Monitoring IoT Networks through an Elastic Stack-based Platform”, Proceedings of 2021 International Conference on Future Internet of Things and Cloud, FiCloud 2021. Virtual, Online, 2021, Pp. 184-191. DOI 10.1109/FiCloud49777.2021.00034.

[25] I.V. Kotenko, A.A. Kuleshov, I.A. Ushakov “Aggregation of elastic stack instruments for collecting, storing and processing of security information and events”, Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). California, USA: Institute of Electrical and Electronics Engineers, 2017, pp. 1 – 8. DOI 10.1109/UIC-ATC.2017.8397627.

[26] U. Chaudhuri, S. Dey, B. Banerjee, A. Bhattacharya, M. Datcu “Interband Retrieval and Classification Using the Multilabeled”, Sentinel-2 BigEarthNet Archive. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2021, vol. 14, pp. 9884-9898. DOI 10.1109/JSTARS.2021.3112209

[27] L. Maltoudoglou, A. Paisios, H. Papadopoulos, L. Lenc, J. Martínek, P. Král “Well-calibrated confidence measures for multi-label text classification with a large number of labels”, Pattern Recognition, 2022, vol. 122, pp. 108271. DOI: 10.1016/j.patcog.2021.108271

[28] O.I. Sheluhin, S.Yu. Rybakov, A.V. Vanyushina “Modified Algorithm for Detecting Network Attacks Using the Fractal Dimension Jump Estimation Method in Online Mode”, Proceedings of Telecommunication Universities, 2022, vol. 8, no 3, pp. 117-126. (in Russian) https://doi.org/10.31854/1813-324X-2022-8-3-117-126