Peter the Great St. Petersburg Polytechnic University

Published in manuscript form

Uspenskij Mikhail Borisovich

Development and research of the diagnostic information processing models and methods for the storage system failure detection and isolation

Specialty 05.13.01 – System analysis, information management and processing

Abstract

thesis for the degree
of Candidate of technical sciences

St. Petersburg – 2020

The work was carried out in Peter the Great St. Petersburg Polytechnic University.

Scientific advisor:	Candidate of technical sciences, associate professor **Itsykson Vladimir Mikhailovich** SPbPU, director of the Higher school of intelligent systems and supercomputer technologies
Official opponents:	
Doctor of Physico-Mathematical Sciences, Boris Alexandrovich Kulik	Institute for Problems in Mechanical Engineering of the Russian Academy of Sciences (IPME RAS), laboratory of intellectual electromechanical systems, lead researcher
Candidate of technical sciences, associate professor, Melekhova Anna Leonidovna	MIPT, assistant professor
Lead organization:	Joint Stock Company Concern Avrora Scientific and Production Association

Defense of the thesis will take place on 26th November 2020 at 16:00 at the meeting of the dissertation council У.05.13.01 of Peter the Great St. Petersburg Polytechnic University (Russia, 195251, St.Petersburg, Polytechnicheskaya, 29, III educational building, room 506). The dissertation can be found in the library and on the website http://www.spbstu.ru of Peter the Great St. Petersburg Polytechnic University

Abstract was sent out on «__» ____________2020 г.

Academic secretary of the dissertation council У.05.13.01.

Doctor of technical sciences, associate professor A.E. Vasiliev

GENERAL THESIS DESCRIPTION

Relevance of the research topic. The problem of timely detection of failures in the data storage systems (DSS) is currently very important due to the sharp increase in the volume of information stored in various ways. According to a joint report by experts from IDC and Seagate, the total amount of stored data may exceed 160 zettabytes by 2025. At the same time, there is a steady tendency to increase the share of data located centrally in corporate and commercial data storage and processing centers (DPC) using storage systems of various levels.

DSS are widely used in the banking and telecommunications sectors, government agencies, enterprises of the military-industrial and fuel-energy complexes, in the field of education and science for storing personal data, financial and regulatory documents, project documentation, training materials, etc.

To increase the safety of data and ensure their constant availability, DSS developers are using increasingly sophisticated hardware and software solutions aimed at creating redundancy and caching schemes at different levels of topology; therefore, modern DSS are complex hardware and software systems that include many interconnected elements. The complexity of diagnostics of such systems is constantly increasing, because as the volume of stored data increases, the complexity of the solutions used increases also. As a result, the importance of timely and accurate failure detection is growing, not only in data carriers (hard drives, solid-state drives), but also in elements that are not intended directly for data storage (storage controllers, network infrastructure elements), as well as failures that occur as a result of inter-element interaction.

To detect such failures, it is necessary to develop new methods and models that provide a comprehensive approach to diagnostics based on the analysis of an extended set of storage parameters. In particular, this work is devoted to solving problems of monitoring the technical condition, as well as locating source of failures and determining the failures root cause.

The degree of scientific development of the problem.

Actual approaches to computer systems diagnostics, based on the usage of models and processing tools for monitoring data are applicable to storage systems. In this case, the diagnostic model is determined in accordance with GOST 20911-89 as a formalized description of an object necessary for solving diagnostic problems.

Recent works in this field: papers of S. Chen, L. Queiros in the field of computer systems diagnostics using models; studies of J. Ding, G. Wang, D. Dasgupta, D. Lee, F. Gonzalez, C. Eiras-Franco in the field of diagnostics based on the analysis of monitoring data. Diagnostics using a combined approach in various technical objects is described in the works of A. Slimani, J. Luo, D. Jung, C. Sundstrem, S. Frank, M. Heaney.

Methods for diagnosing storage media failures are studied in the works performed in the laboratories of Microsoft, Google, Facebook, etc., as well as by independent researchers: I. Narayanan, D. Wang, J. Murray, F. Mahdisoltani and others.

One of the most promising areas of research is the diagnosis of computer systems using software logs. Various aspects of such diagnostics are considered in the works of R. Vaarandi, F. Kiyang, J. Ming, S. Bertero, S. Messoudi and others.

Up-to-date technical solutions in the field of storage diagnostics are presented in the software packages of IBM, Fujitsu, HP, Dell, Zabbix and others.

The analysis of the state of the subject matter shows the need for further research of the approaches that allow expanding the number of classes of detected failures due to the ability to process heterogeneous diagnostic information and formalize expert knowledge about the operation of storage, including through the joint application of diagnostic models and methods based on the processing of monitoring data.

The scientific task of the thesis is to develop models, methods and algorithms that extend the set of possible failure classes in DSS.

To achieve these goals, the following tasks are solved:

1) Analysis of a DSS as an object of diagnostics, determination of requirements for failure detection methods in DSS and software that implements them. Analysis of the recent papers and solutions in the field of the computer systems diagnostics, with a comparative analysis of their features.

2) Development of a method for constructing diagnostic models for the DSS, defining the relations between the parameters and possible states of the system and its elements, where relations can be specified both as deterministic relations between diagnostic entities and as a functions of machine learning.

3) Development of the methods and software for converting a diagnostic model to a simplified graph view used as a part of the diagnostic software.

4) Development of the methods and tools for detecting failures in DSS based on the analysis of textual information obtained during the DSS monitoring using text classification machine learning methods.

5) Experimental verification of the developed models, methods and tools on the target storage platform.

The object of the research is DSS and the relations between diagnostic parameters, the state of individual elements of DSS and DSS in general.

The subject of the research are the models and methods for the failure detection in the DSS.

Academic novelty of the statements to be defended:

- The paper proposes and applies a method for building diagnostic models of DSS based on the usage of an ontological model and methods for processing and analyzing expert information, which differs from the existing ones by the ability to describe relations between ontology objects using machine learning algorithms.
- The paper proposes and applies an approach to DSS failure detection, which differs from the existing ones by using a classification algorithm for partially structured DSS software textual monitoring data based on the machine learning.
- The paper proposes an algorithm for analysis, transformation and processing of the textual information obtained during monitoring of DSS software, which differs from the existing ones in that it allows to detect failures using classification of partially structured texts without a detailed analysis of the structure, format and sequence of monitoring messages.

- The paper proposes a method for detecting DSS failures, which differs from the existing ones by the joint use of the ontological model and machine learning algorithms for processing text information obtained during the monitoring of DSS.

Theoretical and practical significance of the research.

The approach proposed in this paper develops the scientific basis for building diagnostic models designed for automatic and automated failure detection in DSS, using ontological model to describe the relations between the state of an object, its elements and diagnostic parameters, expanding the apparatus of ontological modeling methods by adding the ability to describe the relation between classes, named individuals and attributes using external procedures based on machine learning algorithms.

The practical significance of the study lies in the development and implementation of the failure detection method for DSS as part of diagnostic software. This software will improve the reliability of data storage and ensure high availability. The resulting solution can be used to diagnose a wide range of storage configurations by changing the set of elements of the ontological model. Moreover, if the storage configurations differ only in the application of various quantitative characteristics of redundancy schemes, then no additional actions are required to adapt the ontological model, and the resulting solution remains operational for the scalable DSS.

Methodology and methods of dissertation research are based on an interdisciplinary approach with the use of diagnostic methods based on diagnostic models and methods for the analysis of patterns in monitoring data, including methods of graph theory, pattern recognition theory, ontological modeling and semantic networks, methods of natural language processing using machine learning algorithms.

Statements to be defended:

1) A method for constructing diagnostic models of storage systems that differ in hardware configuration, software stack, and parameters of the implemented redundancy schemes, which allows to detect more failure types compared with existing solutions by providing the possibility of simultaneous usage of various diagnostic parameter types.

2) A method and algorithm for the analysis, transformation and processing of textual information obtained during the monitoring of storage systems, which, unlike existing solutions, allows to detect failures without a detailed analysis of the structure of monitoring data, format and sequence of text messages.

3) A comprehensive method for detecting failures in DSS, based on the joint use of the DSS diagnostic model and the method of processing textual DSS monitoring data using machine learning algorithms, which allows scaling of the ontological model of the storage and provides an increase in the number of detected failure types compared with existing methods.

The validity and reliability of the scientific results is achieved by using a proven mathematical apparatus, the correspondence of experimental data to theoretical assumptions and also by successful implementation of the developed methods and models in an experimental sample of the fault detection hardware and software complex in DSS.

Implementation of the results. The results of the research were used in the development of the experimental sample of the DSS fault prevention software and hardware complex carried out with the financial support of the Ministry of Science and Higher Education of Russian Federation within the framework of the Federal Program "Research and Development in Priority Areas for the Development of the Russian Science and Technology Complex for 2014-2020", unique identifier: RFMEFI58117X0023.

Approbation of the results. The results were presented at the seven russian and international conferences: Topical Problems of Architecture, Civil Engineering and Environmental Economics, Moscow, Russia, 2018; XXIII International scientific-practical conference «System analysis in research and management» (St.Petersburg, Russian Federation, 2018); International Conference on Soft Computing and Measurement (St. Petersburg, Russia, 2018); International Conference Cyber-Physical Systems and Control (St. Petersburg, Russia, 2019); 17th IEEE International Symposium on Intelligent Systems and Informatics (Subotica, Serbia, 2019); 33rd International Business Information Management Association Conference (IBIMA) (Madrid, Spain, 2019); 2019 International Scientific Conference on Energy, Environmental and Construction Engineering (EECE) (St. Petersburg, Russia, 2019).

Papers. The main results on the topic of the dissertation are published in 13 scientific papers, including 3 from the list, recommended by VAK and 5 in the journals, indexed in the Scopus and Web of Science databases. Also, 9 software programs were patented in Russian Federation.

Personal contribution. All the results presented in this thesis were obtained by the author personally.

Structure and volume of the thesis. The thesis consists of introduction, five chapters, conclusion and two annexes. The size of the main part of the thesis – 150 pages, overall size – 153 pages, including 27 tables and 24 pictures. Bibliography consists of 154 references.

THESIS CONTENT

The introduction substantiates the relevance of the research topic, defines the purpose and tasks to be solved, objects and subject of research, formulates the statements to be defended, their theoretical and practical significance and scientific novelty.

A comparative analysis of the most relevant software and hardware tools designed to detect, predict and prevent the occurrence of computer systems failures that can be used to diagnose DSS is carried out. Their classification according to the scope of application, the principle of data collecting and data type of diagnostic parameters, obtained as a result of the monitoring process and used in the diagnostic process is proposed.

A set of performance criteria for the analyzed tools is defined, and their comparison is carried out in accordance with these criteria.

Based on the identified list of approaches that are most suitable for detecting failures in DSS, the analysis of modern scientific papers aimed at determining the most relevant methods for implementing these approaches, including promising algorithms for detecting anomalies in diagnostic data, classification and clustering algorithms, is carried out.

The need for the development of new methods and tools for detecting DSS failures aimed at a more efficient use of data logs for storage systems, also as an element of the diagnostic model of storage systems, is identified.

The **second Chapter** provides a description of the method for constructing a diagnostic model designed to detect failures in DSS. In fact, a diagnostic model is constructed to formalize and determine the relations between the state of an object, its elements and diagnostic parameters, including, in particular, textual monitoring data. Textual monitoring data is accumulated in the storage software logs in the form of messages that are sequentially recorded as events occur in the system.

As part of the study, an approach to creation of the ontological diagnostic DSS model is proposed. An ontological model is understood as a knowledge base based on the ontology of reliability of DSS, created using the traditional methodology, interpreted using new diagnostic algorithms and analysis of the system logs to determine the technical condition of the system.

The developed model has the following features:

- The ontological model defines formal description of the relations between heterogeneous knowledge about the behavior of DSS and storage elements obtained from expert assessments, from the analysis of statistical data and historical data on the functioning of storage systems, simulation results and various diagnostic parameters.
- The ontological model defines formal description of the hierarchical structure of storage systems, their components and subsystems;
- The ontological model satisfies the scalability requirement. It provides the ability to adapt to various configurations and architectures of DSS, while the model is scaled with minimal program code changes for the diagnostic procedure.

Thesis defines the basic concepts (classes) of the ontological model, which formally define the structure of the diagnostic object and its elements, possible states of the diagnostic object and its elements, possible events in the diagnostic object and its elements, as well as diagnostic parameters.

“Storage parameter” class *(P)* describes the diagnostic parameters of storage. The values of the storage system parameters obtained during monitoring are classified depending on the possible values of the parameter *{V}* defined in the model as related to a particular event.

"Storage component" class (K) defines the basic structural elements of DSS. At the model level, the component is atomic and indivisible, meaning that the maximum depth of searching for a failure is limited by the component level.

"Storage subsystem" (Ss) class describes a group of storage elements united by some functional feature. A storage subsystem can be used to group classes not only for the components, but also for subsystems, including subsystems of the same level.

"System" class defines diagnostics object (that is, storage as a whole) and is intended to describe a group of top-level subsystems.

"State" class *{D}* is used to define the health of the subsystem, storage component, and storage as a whole and can have one of four values: "operational state", "pre-failure state", "vulnerable state", and "complete failure".

"Event in storage" class *{F}* is a set of values of storage parameters, states of components or subsystems of storage that characterize the occurrence of a failure. Each event is classified as one of the *{D}* States, depending on its level of criticality.

To determine the state of D_{E_i} in the E_i element, the presence of events in the child elements is determined, starting with the elements of the lowest nesting level - parameters and further up the hierarchy of elements from components to subsystems.

Formally, the ontological model is described as follows:

Parameter ***P_i*** of DSS:

$$P_i = \{Rp_j, V_j\}_{j=1,..,N} \tag{1}$$

where N – number of relations of the i-th DSS parameter, $< Rpj, Vj >$ - touple relation/value.

Component ***K_i*** of DSS:

$$K_i = <\{P_{ij}\}_{j=1,..,N}; \{Fe_k, De_k, Rep\}_{k=1,..,M} > \tag{2}$$

where N – number of parameter, M – number of possible events of the i-th DSS, *<Fe_k, De_k, Rep>* - tripple event/state/parameter relation, *{P_{ij}}* – DSS component parameter set.

DSS subsystem ***S_i***:

$$S_i = <\{K_{ij}\}_{j=1,..,N}; \{S_k\}_{k=1,..,M}; \{Fs_t, Ds_t, [Rs_t(K_{ij}), Rs_t(S_k)]\}_{t=1,..,L} > \tag{3}$$

where N – number of components, M – included subsystems, and L – events – of the i-th DSS subsystem, *<Fs, Ds, {Rs_t, Rs_t}>* - triple <event/state/relation, defining component or subsystem – event source>, *{K_{ij}}* – DSS component set.

Than system overall:

$$Storage = <\{S_i\}_{j=1,..,N}; \{Fst_j, Dst_j, Rst_j\}_{j=1,..,M} > \tag{4}$$

where N – number of the parameters, M – number of the possible events, *<Fst, Dst, Rst>* - triple event/state/subsystem relation. Evaluation of the DSS current state:

- Set of the DSS component states ***{S_c}***, DSS subsystem states ***{S_s}*** and DSS as a whole ***{S}*** (from the list: "operational", "pre-failure", "malfunction", "total failure")*;*
- a set of detectable events in the storage {F}, each of which corresponds to one of the storage States.

Two types of relations *{R}* are used to describe relations between a diagnostic object, its elements, states, and diagnostic parameters (table 1): relations based on ontological properties of objects and data properties, hereinafter referred to as deterministic relations, and relations that require an external procedure, hereinafter referred to as conditional relationships (figure 1).

Table 1: Relation types in the knowledge base

Relation name	Comment
manifests_in (if_fatal_manifests_in, if_warning_manifests_in)	Defines component health level of the failure
shows_in_parameter	Defines parameters of the failure detection
is_normal (is_normal_when, is_normal_below, is_normal_above)	Defines range of the diagnostic parameter normal values in a simple event
solves_with/described_by	For defining the link to the external rule
depends_on (strongly_depends_on, majorly_depends_on и depends_with_ecc_on)	Defines how the state of the storage subsystem depends on the states of the elements of this subsystem
consists_of	Description of the hierarchy of components and subsystems of a specific storage configuration as a tree
causes (if_failed_causes, if_warning_causes)	Indicates the names of system events corresponding to subsystem states
interprets_as	Allows definition of the DSS health state

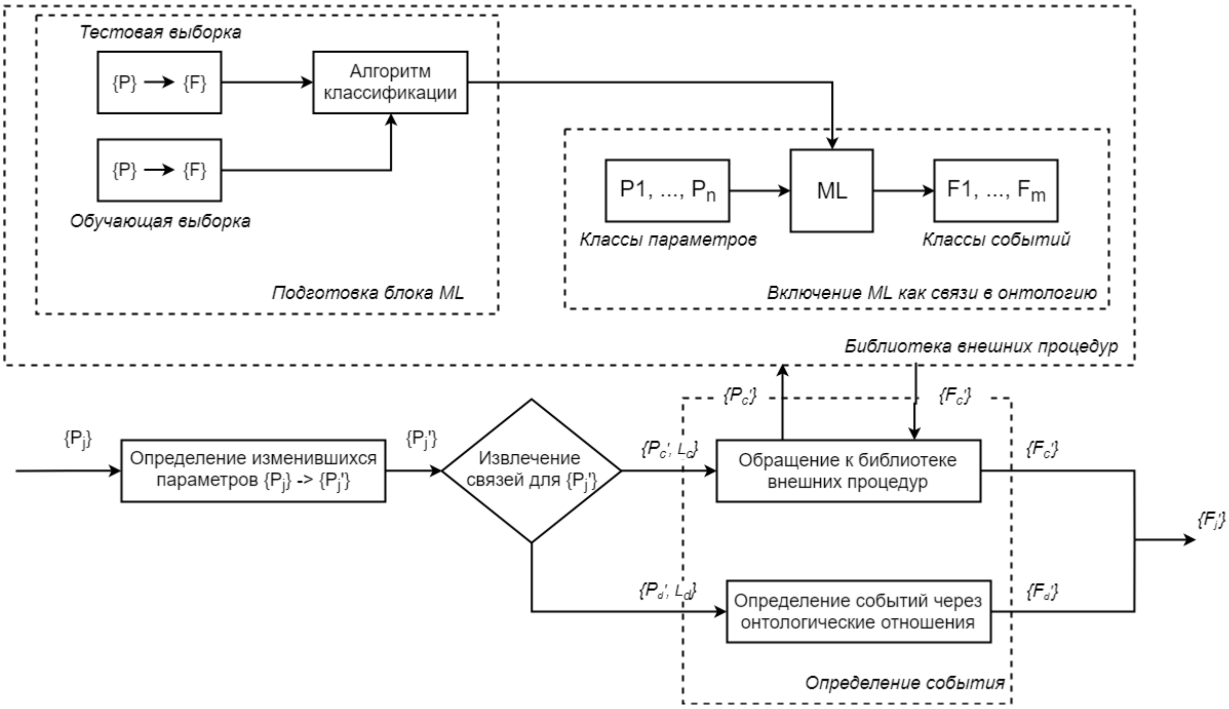
Тестовая выборка
{P} → {F}
Алгоритм классификации
{P} → {F}
Обучающая выборка
Подготовка блока ML
P1, ..., Pn
ML
F1, ..., Fm
Классы параметров
Классы событий
Включение ML как связи в онтологию
Библиотека внешних процедур
{Pc'}
{Fc'}
{Pj}
Определение изменившихся параметров {Pj} -> {Pj'}
{Pj'}
Извлечение связей для {Pj'}
{Pc'; Lc}
Обращение к библиотеке внешних процедур
{Fc'}
{Fj'}
{Pd'; Ld}
Определение событий через онтологические отношения
{Fd'}
Определение события

Fig. 1 – Conditional relations

A pre-trained machine learning algorithm is specified as an external procedure. It takes an input vector of diagnostic parameter values and classifies them as one of the related classes. The procedure is called in the diagnostic process to determine the event associated with the specified diagnostic parameters.

After extracting from the complete list of parameters *{P$_i$}* those parameters {*P$_i$*'} whose values have changed, their possible relations *{L}* with events in the components are determined. After that, the search for events that occurred is performed both for conditional relations and their corresponding parameters (*{P_c', Lc}*) and deterministic relationships and their corresponding parameters (*{P_d', Ld}*). Parameters *{P_c'}* are passed to external procedures determined according to *{Lc}*.

Figure 2 shows ontology fragment, defining an event on the various levels of DSS topology.

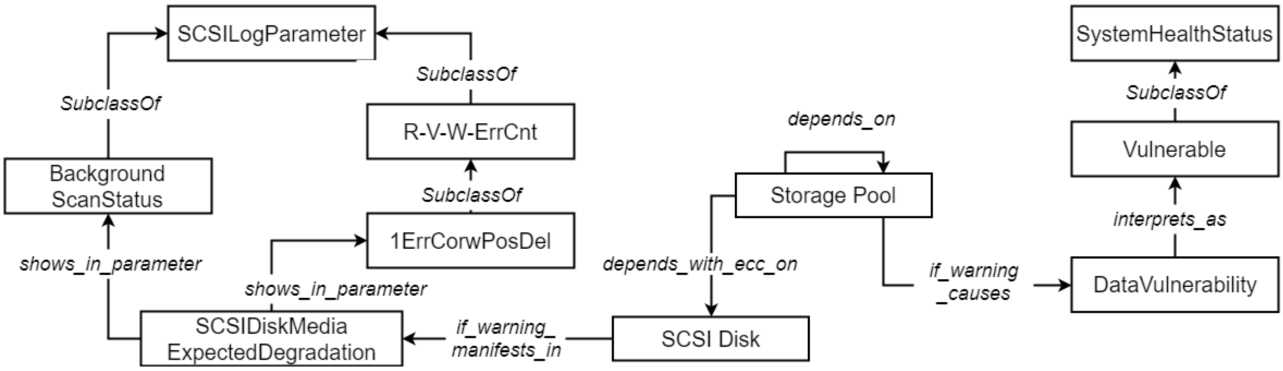

Fig. 2 – Event on the various levels of DSS topolgy

Thus, the basic ontological model described in the second chapter includes three main sections:

- section describing the storage topology;
- section describing the DSS parameters;
- failure description section.

In the conclusion of the second Chapter, the storage system logs as a possible source of diagnostic information for the ontological model are evaluated. Requirements are defined for the log types and message formats suitable for the developed models, methods and algorithms.

The third chapter contains the description of the development of an algorithm for the dynamic failure detection in DSS, which can be used as an ontological conditional connection, with message blocks from the DSS software logs as input.

The developed algorithm assumes a combined approach to the analysis of DSS software logs. Each message in the log is defined as a combination of structured and unstructured text. Each message is split into a heading part and a text part according to a set of rules defined within the helper section of the ontological model.

Text classification methods based on the machine learning, are used for the failure detection. Input data for such methods is composed from the unstructured text parts of the messages within a given time interval.

Algorithm of the failure detection contains following steps:

1) Splitting of the log messages in to the heading and textual parts.
2) Preprocessing of the textual message parts.
3) Building of the relevant software log texts by merging of the message texts withing a given time interval.
4) Building of a text vector representation and the definition of additional numerical parameters for the evaluation of machine learning algorithms.
5) Usage of a pre-trained classification algorithm for the DSS software logs text classification using data, obtained in clause 4.

6) Localization of the sources of failures using ontological conditional relations.

The procedure for splitting messages into the text part and the header is based on the combinatorial parsing method with only bind operation being used:

bind :: Parser a -> (a -> Parser b) -> Parser b
p ‘bind‘ f = \inp -> concat [f v inp’ | (v,inp’) <- p inp]

In this case, the parsing of the string S by the set of parsers L_i, $i\epsilon[0, N]$ is performed as follows: a string is passed to the input L_i of the primitive parser, the remainder of the string is passed to the input L_{i+1} of the primitive parser, etc.

This method allows the separation of the header part of the messages usage of its contents (for example, timestamp) in the diagnostic procedure.

Pre-processing the text part of a message consists of the following actions: tokenization of the text, letter case normalization, non-letter tokens and stop words filtering, stemming and message filtering.

Failure detection in a raw text procedure assumes application of the classification algorithm for the X_u vector which is a union of the log text vectorization V_t and the additional numeric parameters vector $X_{m.}$

To improve the efficiency of building a vector representation of the text, two types of coefficients were used together:

- the $W_{tf\text{-}idf}$ weighting factor that characterizes the importance of a word within a context and represents the ratio of the frequency of occurrence of a word in a document to the frequency of use of the word in all documents in the sample;
- weight coefficient W_{err}, which characterizes the importance of the word for fault identification, determined by the position of the word in the time window.

Thus, The final V_W vector describing each word has the following form (5):

$$V_w = W_{tf-idf}W_{err}[V]; \qquad (5)$$

where *[V]* – word vector representation.

Accordingly, the V_T vector describing the current time window should be defined as (6):

$$V_t = \frac{\sum_{i=1}^{n} W_{tf-idf}W_{err}[V_i]}{n}; \qquad (6)$$

where n – number of words in a set.

Additional X_m parameters describe the quantitative characteristics of the log text:
- Number of tokens in the *i-th* log;
- Nuber of messages in the *i-th* log;
- Average message length in the *i-th* log;
- Presence of messages in the *i-th* log;
- Average number of tokens per second in the *i-th* log;
- Average number of messages per second in the *i-th* log.

The importance of additional parameters was evaluated using two methods. The first method is to use the Random Forest algorithm and calculate the Gini criterion at each step:

$$G(k) = \sum_{i=0}^{J} P(i) * \left(1 - P(i)\right); \qquad (7)$$

where *P(i)*-probability of the classification *i* for the feature *k.*

An alternative way is to calculate the F-criterion on a variety of features. The F-criterion is the ratio of intergroup dispersion to intragroup:

$$F = \frac{Var_b}{Var_w} \tag{8}$$

where Var_b – intragroup, Var_w – intergroup dispersion.

The check was performed on the available marked-up data obtained from the results of storage testing. The total sample contains 5904 log packages (which is about 350 GB of log files) placed on storage controllers, including 1574 corresponding to failures that occurred during the operation of the storage, indicating the approximate time interval for the occurrence of failures. A total of 41 types of various faults were identified in the available data.

An assessment of the importance of the features made it possible to exclude the feature "presence of messages in the *i-th* log", others showed their importance for classification.

The fourth chapter describes the combined algorithm using the ontological model and the methods of machine learning and its application as part of the built-in diagnostic software. The diagnostic procedure includes the following main steps, repeated in accordance with the specified interval for starting the failure detection procedure:

1) Monitoring data collection.

2) For each component, its determinate state is determined by putting the parameter values in the rules for symptoms evaluation (through both conditional and deterministic relationships)

3) For each component, the states of the associated subsystems are determined.

4) For storage systems as a whole, a state is determined as a function of the states of its subsystems.

At the first stage of the procedure for building and applying the ontological model as part of the embedded diagnostic software of the target storage (figure 3), the ontological model is prepared by filling it with expert data in the Stanford Protégé ontology editor. For the practical application of the ontological model as part of the software, it is proposed to transform it to a simplified graph form, which involves the conversion of traditional rdf to rdf-nquad format, and a model description in the format [node] - <link> [node] [context]. Such a transformation eliminates the cumbersome inheritance constructs used in the OWL ontology and reduces them to a simplified form.

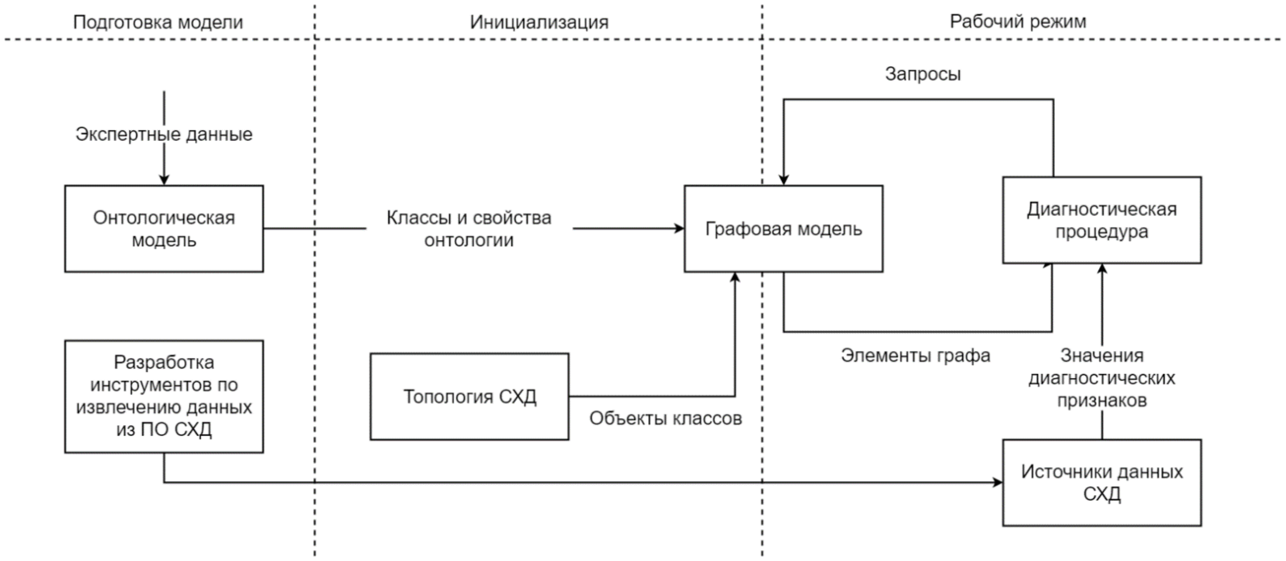

Рисунок 3 – Схема применения моделей

Conversion to a simplified graph view is performed at the initialization stage. The model structure, its classes, object properties, and data properties, are obtained by converting the ontological model, and instances are obtained by polling the embedded DSS software. After that, the resulting model is stored in a graph database and used by the diagnostic procedure to obtain information about the relations of the diagnostic parameter values and DSS failures.

The architecture of the software implementation of the diagnostic procedure assumes obtaining the current value of the storage parameters, including the texts of storage logs, from the database, populated by the monitoring and data collection module. The software instance is run by the cluster tools on one of the storage controllers from the cluster, and determines the state of the system as a whole and its individual subsystems and components.

In the fifth Chapter, an experimental test of the diagnostic software was performed on an array of log texts with a built-in model and pre-trained text classification algorithms. The purposes of the experiment were:

1) Selection of elements of the classification procedure with the best classification results, including the classifier algorithm itself (classical Random Forest algorithms, naive Bayesian classifier, KNN, logistic regression, the support vector method and deep learning neural networks LSTM, RNN and LSTM with an attention mechanism are considered), the method of the text vectorization and the method of the log text combining.

2) Evaluation of the entire DSS failure detection approach performance.

The considered set of classification algorithms was chosen based on the analysis of the results of their application to the multi-class text classification problem presented in current scientific publications (see Kovasari K. et al., Yu S. et al.) and applied research (Li S.). Algorithms hyperparameter tuning was performed by applying a combination of random search method and a complete search in the vicinity of the assumed optimal values.

The results of experiments have shown that the most effective method is to build a separate vector representation for each log and combine them together with the vectors of additional features defined for each log into a single feature space. The results of comparing the quality of classification are presented in table 2. The Random Forest algorithm showed the greatest accuracy in solving the problem of classifying log texts.

Table 2 – Comparison of the classification performance

Classificator type	Average precision	Average recall	Average f1-score	Average model training time
Random Forest	0,74	0,71	0,71	183 s
Naïve Bayes	0,48	0,27	0,28	205 s
KNN	0,38	0,39	0,37	166 s
Logistic regression	0,28	0,33	0,28	498 s
SVM	0,23	0,26	0,20	430 s
LSTM	0,47	0,44	0,45	~4 hours
GRU	0,24	0,22	0,23	~7 hours
LSTM with attention	0,42	0,39	0,40	~ 6 hours

The experiment was conducted using software to simulate failures, since the frequency of their occurrence in the process of regular operation of the DSS is too small to make any conclusions about the developed approach performance. The following results were obtained from the experiment:

- for the failures, detected using deterministic ontological relations, the percentage of identified is 0.99 on a data set of 10,000;
- for the failures, detected using conditional ontological relationships, the percentage of identified is 0.71 on a data set of 10,000.

RESULTS

In the thesis, the actual scientific and technical problem of expanding the set of detected DSS failures is solved by applying the proposed method of diagnostics based on the joint use of an ontological diagnostic model and machine learning methods for analyzing diagnostic parameters.

The following results were obtained:

1) DSS as an object of diagnostics, determination of requirements for failure detection methods in DSS and software that implements them was analyzed. Recent papers and solutions in the field of the computer systems diagnostics, with a comparative analysis of their features were analyzed.

2) A method for constructing diagnostic models for the DSS, which define the relations between the parameters and possible states of the system and its elements, where relations can be specified both as deterministic relations between diagnostic entities and as a functions of machine learning was developed.

3) The methods and software for converting a diagnostic model to a simplified graph view used as a part of the diagnostic software were developed.

4) The methods and tools for detecting failures in DSS based on the analysis of textual information obtained during the DSS monitoring using text classification machine learning methods were developed.

5) Developed models, methods and tools were experimentally verified on the target storage platform.

The research topic may be further developed by researching an impact of the implementation of the more complex vectorizers for the DSS software logs based on the neural networks, and by possible simultaneous usage of different algorithms for solving classification problem.

The results obtained correspond to paragraph 4 "Development of methods and algorithms for solving problems of system analysis, optimization, management, decision-making and information processing", paragraph 5 "Development of special mathematical and algorithmic support for systems of analysis, optimization, management, decision-making and information processing", paragraph 11 "Methods and algorithms for forecasting and evaluating the effectiveness, quality and reliability of complex systems.", paragraph 12 "Methods for obtaining, analyzing and processing expert information" of the passport specialty 05.13.01 System analysis, information management and processing (by industry).

LIST OF PUBLICATIONS ON THESES TOPIC

In the journals recommended by ВАК:

1. Uspenskij M.B. A survey of the approaches to storage systems fault detection / Uspenskij M.B. // Computing, Telecommunications and Control Peter the Great St. Petersburg Polytechnic University – 2019. - №4 – P.145-158.

2. Uspenskij M.B. Application of the ontological model and text classification algorithms in the data storage systems failure detection / Uspenskij M.B. // Izvestia of Samara Scientific Center of the Russian Academy of Sciences – 2020. - №1 - P.107-113.

3. Uspenskij M.B. Automatic failure detection in data storage systems using system software logs / Uspenskij M.B. // Information and space – 2020. - №1 – P. 90-96.

In the journals, indexed in Web of Science and Scopus

4. Uspenskij, M. B. (2019). Log mining and knowledge-based models in data storage systems diagnostics. //E3S Web of Conferences, Vol.140, №03006.

5. Shirokova S.V., Bolsunovskaiya M.V., Loginova A.V., Uspenskiy M.B. Developing a procedure for conducting a security audit of a software package for predicting storage system failures // MATEC Web Conf, 2018. International Scientific Conference on Energy, Environmental and Construction Engineering (EECE-2018). Volume 245, № 10007.

6. Uspenskij M., Makarov A., Sochnev A., Shirokova S., Petrov V. Development of a software structure for monitoring the working capacity of the data storage system for predicting failures and preventing critical situations // Proceedings of the 33rd International Business Information Management Association Conference, IBIMA 2019: Education Excellence and Innovation Management through Vision 2020, 2019. Pp. 8508-8514.

7. Mamoutova O.V., Shirokova S.V., Uspenskij M.B., Loginova A.V. The ontology-based approach to data storage systems technical diagnostics // E3S Web of Conferences, Vol. 91, № 08018.

8. Mamoutova O.V, Uspenskiy M.B., Sochnev A.V., Smirnov S.V. Bolsunovskaya M.V. Knowledge Based Diagnostic Approach for Enterprise Storage Systems // Proceeding of IEEE 17th International Symposium on Intelligent Systems and Informatics, 2019

Other journals

9. Makarov A.S., Bolsunovskaya M.V., Shirokova S.V., Uspenskij M.B., Kuzmichev A.A. Analysis of approaches to the diagnosis of data storage systems // Soft Computing and Measurements: Proceedings of the XXI International Conference, 2018. Vol. 2. Pp. 61-64

10. Pridanova E.V., Uspenskij M. B., Itsykson V. M. Monitoring and analysis of the parameters of a data storage system to assess its condition // Collection of scientific papers of the XXIII International scientific-practical conference “System analysis in design and management”, 2019. Pp. 170-177.

11. Ivanov O.I., Mikhailov E.A., Pustovetov V.I., Uspenskij M.B. The subsystem for the preparation of test projects for control and diagnostic systems // Voprosy radioelektroniki. 2013.Vol. 1. No. 1. Pp. 99-105.

12. Berlik S.A., Ivanov O.I., Uspenskij M. B., Pustovetov V.I. The architecture of the graphic environment of the hardware-software complex KDK //Voprosy radioelektroniki. 2013.Vol. 1. No. 1. Pp. 73-80.

13. Mikhailov A. N., Ivanov O. I., Uspenskij M. B. Signature analyzer for multifunctional hardware and software complex KDK / / Voprosy radioelektroniki. 2014. Vol. 1. No. 2. Pp. 106-112.

Certificates of State registration of computer programmes

1. Uspenskij M.B. Program for collecting data storage system parameters / M.B. Uspenskij, V.D. Petrov, A.V. Sochnev, V.I. Pustovetov. - Certificate of state registration of a computer program No. 2018660284 of 08.21.2018.

2. Uspenskij M.B. A program for simulating the functioning of the hardware component of a data storage system - an information carrier / M.B. Uspenskij, M.E. Karpov. - Certificate of state registration of computer programs No. 2018665078 of 11/30/2018.

3. Uspenskij M.B. A program for simulating the functioning of the hardware component of a data storage system - controller of a PCI-Express factory / M.B. Uspenskij, K. Arzymatov. - Certificate of state registration of a computer program No. 2018665160 dated 03.12.2018.

4. Uspenskij M.B. A program for simulating the functioning of the hardware component of a data storage system - a data storage controller / M.B. Uspenskij, V.S. Belavin. - Certificate of state registration of a computer program No. 2018665676 of December 6, 2018.

5. Uspenskij M.B. The program for collecting and displaying climatic parameters of data storage systems / M.B. Uspenskij, S.V. Smirnov. - Certificate of state registration of a computer program No. 2019614476 of 04/05/2019.

6. Uspenskij M.B. A program for diagnosing a data storage system / M.B. Uspenskij, M.I. Gushchin. - Certificate of state registration of computer programs No. 2019618328 of 06/27/2019.

7. Uspenskij M.B. The program for setting the parameters of the simulation model of the functioning of the data storage system / M.B. Uspenskij, V.S. Belavin. - Certificate of state registration of the computer program No. 2019618010 of June 25, 2019.

8. Ivanov O.I. Testing and diagnostics program for digital electronic equipment using multi-channel signature analysis / O.I. Ivanov, V.I. Pustovetov, M.B. Uspenskij - Certificate of state registration of computer programs No. 2015661325 of 10.23.2015.

9. Ivanov O.I. The program for the preparation of combined test projects / O.I. Ivanov, V.I. Pustovetov, M.B. Uspenskij. - Certificate of state registration of computer programs No. 2015661326 of 10.23.2015.