ОБНАРУЖЕНИЕ АНОМАЛИЙ БОЛЬШИХ ДАННЫХ НЕСТРУКТУРИРОВАННЫХ СИСТЕМНЫХ ЖУРНАЛОВ

Шелухин О.И.[1], Рябинин В.С.[2]

Файлы журналов,которые сегодня ведутся во всех крупных компьютерных системах, содержат важную информацию о текущих событиях, используемую с целью анализа состояния системы. Поскольку количество записей в подобных журналах обычно очень велико, найти необходимую информацию в этих файлах с помощью традиционных методов чрезвычайно сложно. Поэтому необходимы современные методы анализа журналов для поиска соответствующих данных в файлах журналов. Одним из современных подходов к анализу таких данных является использование методов машинного обучения и интеллектуального анализа больших данных. Проблема при поиске важных, как правило аномальных, событий в файлах журналов состоит в том, что события без какого-либо контекста не всегда обеспечивает достаточную информацию для выявления причины возникающих проблем.

Возрастающее значение анализа журнальных файлов в больших компьютерных системах требует разработки автоматизированных методов обработки неструктурированных данных, позволяющих извлекать соответствую-щую информацию из журнальных файлов большого объема без необходимости вмешательства человека.

Анализ полученных результатов показывает, что наилучшие результаты в рассматриваемой бинарной классификации аномалий больших данных показывает алгоритм SVM. Алгоритм ближайших соседей плохо справляется с высоко размерными данными, а потому не может быть рекомендован задачи обнаружения аномалий на основе неструктурированных данных системных журналов, в которых размерность векторов сопоставима с количеством выделенных шаблонов журнальных событий.

Ключевые слова: *Машинное обучение, обработка текстовых данных, парсинг, журналы событий, лог – файлы, неструктурированные данные.*

DOI: *10.21681/2311-3456-2019-2-36-41*

Постановка задачи

Под аномальными будем понимать экземпляры в наборе данных, которые не соответствуют регулярному поведению системы. Система обнаружения аномалий состоит из четырех последовательных этапов: сбор исходных данных, анализ и обработка журнала, извлечение признаков и числовое представление, построение модели обнаружения аномалий [8, 12, 13, 14].

Целью анализа и обработки данных системных журналов (парсинга) является разделение постоянных и переменных частей сообщений и формирование паттернов (шаблонов) событий [9,10]. Для использования наблюдаемых неструктурированных данных в алгоритмах машинного обучения необходимо осуществить их числовое представление [11]. С этой целью, после обработки журнала применяется один из алгоритмов числового представления [12].

Результатом обработки являются кластеры сообщений, разделенные по типу события. Для этого на первом этапе данные журнала разделяются на различные группы, характеризующие последовательность сообщений. Это делается путем последовательного чтения файла журнала и хранения определенного количества событий в области памяти (называемой окном) при обработке набора данных. Окна могут быть фиксированными, скользящими и сеансовыми [13].

В результате данного шага формируется последовательность векторов в числовом виде каждой последовательности сообщений. Для поиска аномальных (аварийных) состояний системы, полученные векторы обрабатываются методами машинного обучения.

Целью работы является исследование эффективности алгоритмов машинного обучения для обнаружения аномальных (аварийных) состояний крупных компьютерных систем путем автоматизированной обработки неструктурированных данных большого объема системных журналов.

Структура данных

Будем использовать в качестве набора данных журнал суперкомпьютера BlueGene/L представленных в https://www.usenix.org/cfdr-data и работе [15]. Задействованный фрагмент журнала содержит 100000 строк, каждая из которых включает в себя временную метку, имя устройства, само сообщение и метку о том, содержит ли данное сообщение информацию о конкретном типе ошибки. Метки о типе ошибок или аномальных(аварийных) состояний системы рассматриваемого фрагмента приведены в таблице 1.

Данный фрагмент содержит следующие метки о типе ошибки:

Для иллюстрации работы алгоритма обнаружения аварийных состояний на основе данных журнала рассмотрим бинарную классификацию, учитывающую каждый раз только один тип аномалий. В качестве аномального (аварийного) состояния рассмотрим ошибки типа “APPSEV”, “KERNMC” [9]. Количество экземпляров данных типов достаточно велико по сравнению с остальными, представленными в таблице 1.

1 Шелухин Олег Иванович, доктор технических наук, профессор, МТУСИ, Москва, Россия. E-mail: sheluhin@mail.ru

2 Рябинин Владимир Сергеевич, Магистрант, МТУСИ, г. Москва, Россия. E-mail: ryabvs@gmail.com

Рис. 1 Этапы обнаружения аномалий больших данных системных журналов [8,13]

Таблица 1. Метки типа ошибки

Метка типа ошибки	Количество ошибок	Пример сообщения	Тип ошибки или аварийных состояний
APPSEV	7461	ciod: Error reading message prefix after LOGIN MESSAGE on CioStream [...]	Ошибка приложения
KERNTERM	512	rts: kernel terminated for reason 1004rts: bad message header: [...]	Ошибка ядра
KERNMNTF	128	Lustre mount FAILED : bglio11 : block id : location	Ошибка ядра
KERNMC	59	KERNEL FATAL machine check interrupt	Ошибка ядра
KERNPOW	48	KERNEL FATAL Power deactivated: R05-M0-N4	Ошибка ядра (отключение питания)
R_DDR_STR	44	ddr: Unable to steer.*consider replacing the card	Замена карты
KERNRTSP	30	KERNEL FATAL rts panic! - stopping execution	Ошибка ядра
KERNMICRO	16	KERNEL FATAL Microloader Assertion	Ошибка ядра
KERNCON	4	KERNEL FATAL MailboxMonitor::serviceMail boxes() lib ido error: -1033 BGLERR IDO PKT TIMEOUT	Ошибка ядра
Остальные	17		

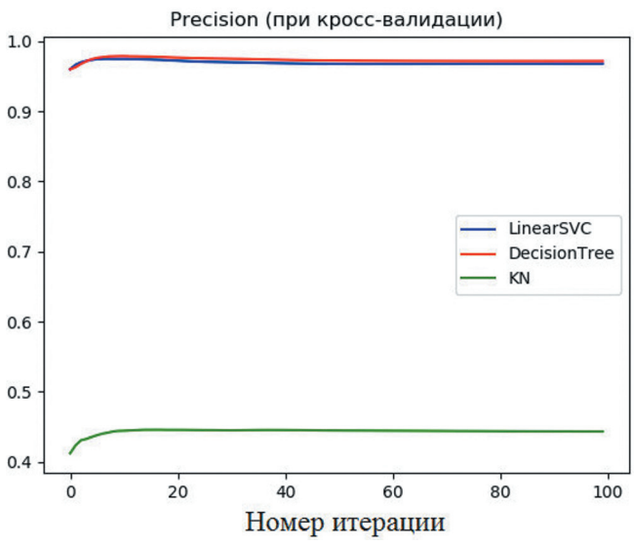
Принимая во внимание, что количество именно аномальных экземпляров в доступной базе данных достаточно мало, необходимо использовать процедуру кросс-валидации (КВ). В подходе КВ, называемом k-Fold, обучающий набор разбивается на k меньших множеств. В начале для каждой из k-частей модель обучается с использованием k-1 частей. Затем алгоритм классификации тестируется на оставшейся части данных. Результатом оценки эффективности, оцениваемой при КВ, является среднее значение, вычисленное по всем итерациям. Для реализации процесса кросс-валидации использовался класс K-Folds из библиотеки sklearn раздела model_selection [https://www.usenix.org/cfdr-data] и значением параметра k=5. Для того, чтобы избежать концентрации аномальных сообщений в одной части набора данных, предварительно производится их случайная перестановка [5,8]. В таблице 2 приведены численные значения объемов выборок для каждой итерации в отдельности, использовавшихся в процессе обучения и тестирования классификатора с учетом 5-блочной КВ.

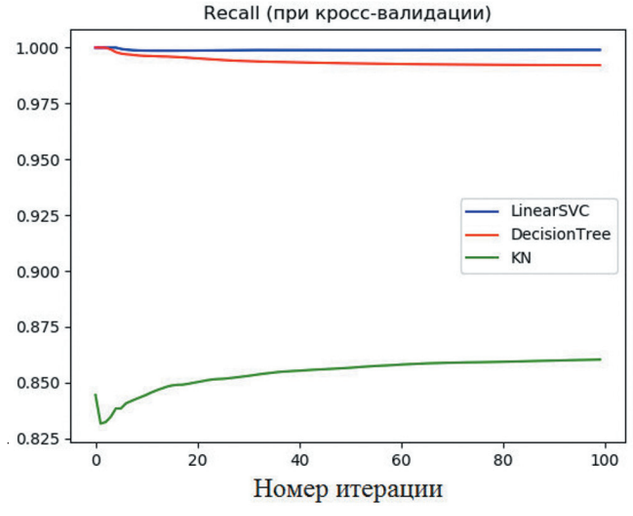
Таблица 2. Характеристики набора данных при кросс-валидации

Аномалия	Номер итерации	Тренировочная выборка		Тестовая выборка	
		Аномальные элементы	Нормальные элементы	Аномальные элементы	Нормальные элементы
KERNMC	1	45	1864	12	466
	2	45	1860	12	466
	3	48	1862	9	468
	4	46	1864	11	466
	5	44	1866	13	464

R_DDR_STR	1	35	1864	9	469
	2	34	1875	10	468
	3	37	1873	7	470
	4	37	1873	7	470
	5	33	1877	11	466
KERNRTSP	1	20	1889	8	470
	2	25	1884	3	475
	3	22	1888	6	471
	4	22	1888	6	471
	5	23	1887	5	472
APPSEV	1	52	1857	12	466
	2	49	1860	15	463
	3	53	1857	11	466
	4	49	1861	15	462
	5	53	1857	11	466

На рисунке 2 представлены зависимости изменения среднего значения для выбранных метрик оценки качества кластеризации при кросс-валидации.





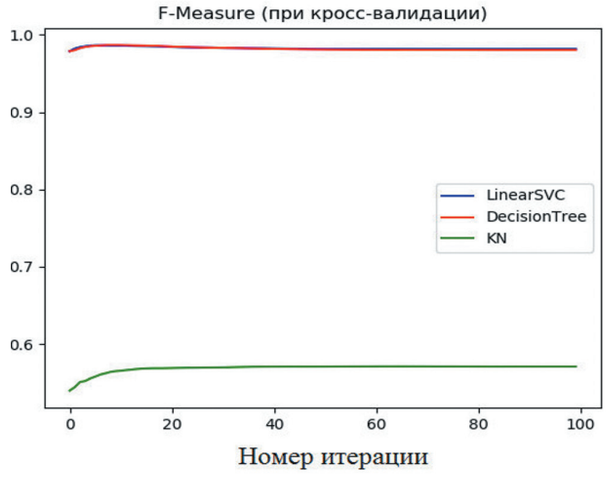


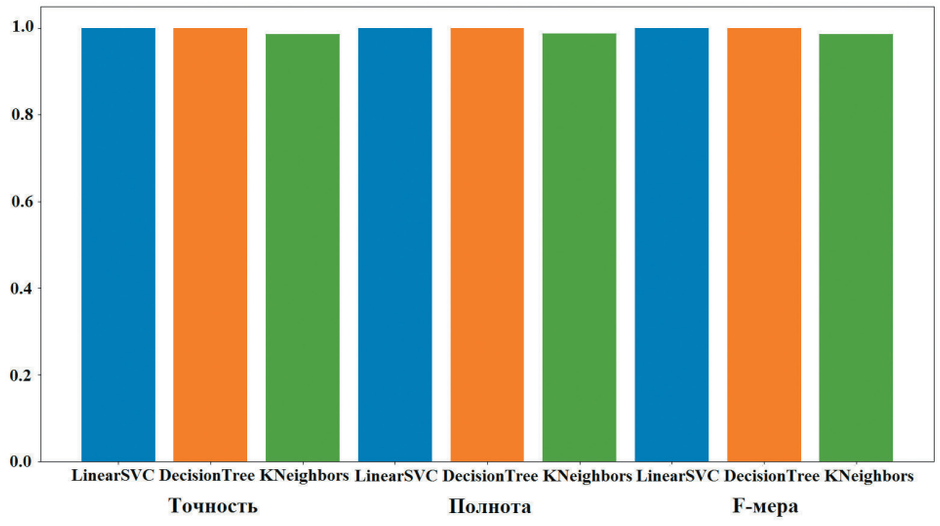
Рис. 2 Метрики при кросс-валидации а)Точность; б)Полнота; в)F-мера

Было проведено 100 итераций обучения и тестирования на одном наборе данных с применением случайной перестановки с помощью функции random.permutation() библиотеки NumPy [http://www.numpy.org/]. При отсутствии кросс-валидации, соотношение между обучающей и тестирующей выборкой составляло 7/3. Сравнение представленных зависимостей показывает, что КВ позволяет получить характеристики более равномерные и устойчивые к колебаниям данных.

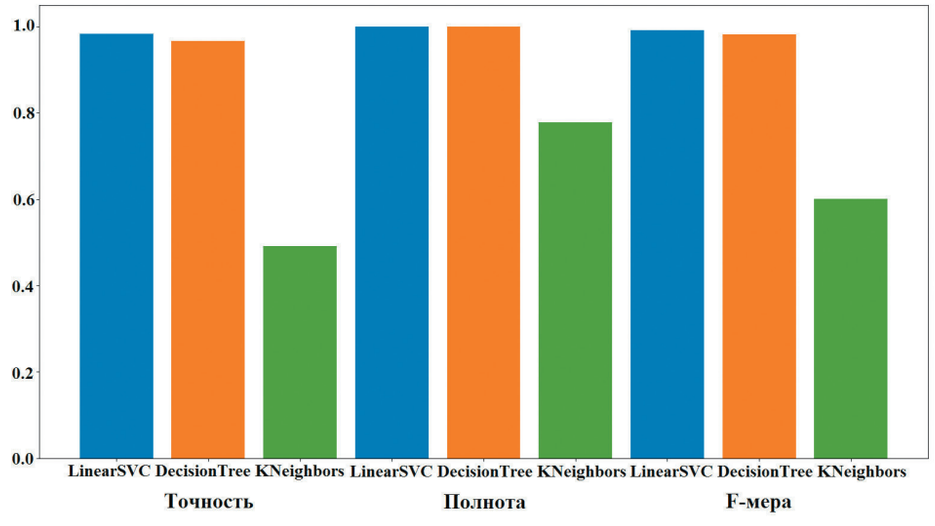
Результаты обнаружения аномалий

Рассмотрим усредненные в процессе КВ результаты оценки трех выбранных методов классификации для рассматриваемых видов аномальных состояний: дерево решений (DecisionTreeClassifier), метод опорных векторов(LinearSVC), K-ближайших соседей(KNeighborsClassifier) [10,11].

Данные методы реализованы в библиотеке для Python – scikit-learn [http://scikit-learn.org/stable/index.html. Результаты классификации приведены на рисунке 4.



а)



б)

Рис. 3 Оценка ошибки типа APPSEV

На основе полученных графиков можно сделать вывод о том, что все три представленных классификатора успешно справляются с задачей классификации аномалии типа APPSEV. Представленные на рис. 4б усредненные в процессе КВ результаты оценки трех выбранных методов классификации для аномалии типа KERNMC показывают, что метод ближайших соседей значительно уступает двум другим алгоритмам классификации SVC и дерево решений. Это может быть вызвано как особенностями данного типа аномалии, так и малым количеством аномальных образцов в выборке на этапе обучения. В данных выборках количество аномальных экземпляров не превышало 1% от общего количества данных доступных для обучения.

Важной характеристикой для визуализации результатов классификации являются ROC-кривые (ROC - Receiver Operating Characteristic), отражающие зависимость истинно положительных(TPR) и ложно положительных (FPR) решений. Как известно, ROC-кривые позволяют оценить качество классификации на основании значения AUC (Area Under the Curve) – площади под ROC-кривой.

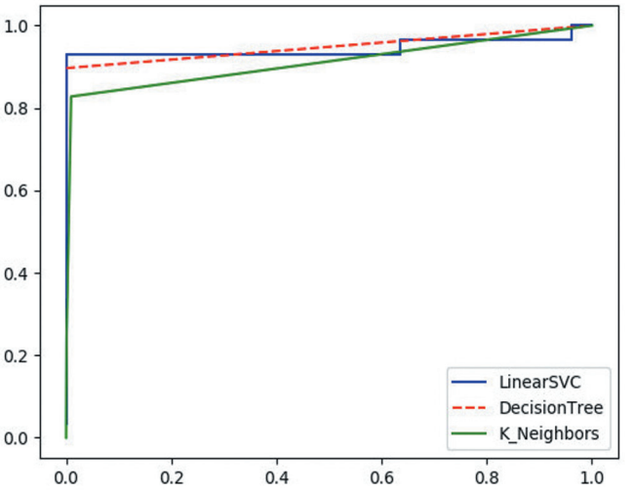


Рис. 4 ROC кривые (ошибка типа “KERNMC”)A

В качестве примера на рис.5 представлены ROC-кривые для бинарной классификации аномалии типа KERNMC, а в таблице 3 соответствующие им значения AUC.

Таблица 3. Значения AUC

Метод	AUC
SVM	0.9993
DecisionTree	0.9230
KNeighbors	0.8808

Как видно из значений, представленных в таблице 3, достоверность обнаружения аномальных событий лежит в пределах 0.92...0.99 при использовании алгоритмов классификации SVM и DecisionTree. Наихудшие результаты у классификатора, обученного методом K-ближайших соседей. Для более точного сравнения выбранных методов, необходимо большее число аномальных элементов в выборке.

Выводы

Анализ полученных результатов показывает, что наилучшие результаты в рассматриваемой бинарной классификации аномалий больших данных показывает алгоритм SVM. Алгоритм «ближайших соседей» плохо справляется с высоко размерными данными, а потому не может быть рекомендован задачи обнаружения аномалий на основе неструктурированных данных системных журналов, в которых размерность векторов сопоставима с количеством выделенных шаблонов журнальных событий.

Рецензент: *Басараб Михаил Алексеевич, доктор физико-математических наук, профессор, МГТУ им. Н.Э. Баумана, Москва, Россия. E-mail: bmic@mail.ru*

Литература:

1. Zou D., Qin H., Jin H., Qiang W., Han Z., Chen X. (2014) Improving Log-Based Fault Diagnosis by Log Classification. In: Hsu CH., Shi X., Salapura V. (eds) Network and Parallel Computing. NPC 2014. Lecture Notes in Computer Science, vol 8707. Springer, Berlin, Heidelberg. DOI https://doi.org/10.1007/978-3-662-44917-2_37
2. Min Du, Feifei Li, Guineng Zheng, Vivek Srikumar. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning", CCS ,17 Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, DOI 10.1145/3133956.3134015
3. Daniel Dias Gonçalves. Automatic Diagnosis of Security Events in Complex Infrastructures using Logs". Instituto Superior Tecnico, Universidade de Lisboa May 2015
4. Christophe Bertero, Matthieu Roy,Carla Sauvanaud, Gilles Tredan.Experience Report: Log Mining Using Natural Language Processing and Application to Anomaly Detection.IEEE 28th International Symposium on Software Reliability Engineering (ISSRE), 2017, DOI: 10.1109/ISSRE.2017.43
5. Kristian Hunt. Log Analysis for Failure Diagnosis and Workload Prediction in Cloud Computing. STOCKHOLM, SWEDEN 2016 URN: urn:nbn:se:kth:diva-189186
6. Berkay Kicanaoglu.Unsupervised Anomaly Detection in Unstructured Log-Data for Root-Cause-Analysis. Tampere University of Technology, 2015 URN: http://URN.fi/URN:NBN:fi:tty-201507291465
7. Qiang Fu, Jian-Guang Lou, Yi Wang, Jiang Li. Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis. ICDM ,09 Proceedings of the 2009 Ninth IEEE International Conference on Data Mining, December 06 - 09, 2009, DOI: 10.1109/ICDM.2009.60.
8. Рябинин В.С., Фармаковский М.А. Обнаружение и диагностика аномальных состояний компьютерных систем средствами интеллектуального анализа данных системных журналов. Сборник трудов: XII Международная отраслевая научно-техническая конференция "Технологии информационного общества", 14-15 марта 2018г.
9. Shilin He, Jieming Zhu, Pinjia He, Michael R. Lyu.Experience Report: System Log Analysis for Anomaly Detection. 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), 23-27 Oct. 2016, DOI: 10.1109/ISSRE.2016.21
10. Pinjia He, Jieming Zhu, Shilin He, Jian Li, Michael R. Lyu. An Evaluation Study on Log Parsing and Its Use in Log Mining. 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 28 June-1 July 2016, DOI: 10.1109/DSN.2016.66.
11. Wei Xu, Ling Huang, Armando Fox, David Patterson, Michael I. Jordan. Detecting Large-Scale System Problems by Mining Console Logs. Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles, October 11 - 14, 2009, DOI: 10.1145/1629575.1629587.
12. Tim Zwietasch. Detecting Anomalies in System Log Files using Machine Learning Techniques. University of Stuttgart, 2014. http://dx.doi.org/10.18419/opus-3454
13. Шелухин О.И., Рябинин В.С., Фармаковский М.А. Обнаружение аномальных состояний компьютерных систем средствами интеллектуального анализа данных системных журналов". Вопросы кибербезопасности №2(26), 2018. DOI: 10.21681/2311-3456-2018-2-33-43
14. P. He, J. Zhu, S. He, J. Li, and M. R. Lyu. An evaluation study on log parsing and its use in log mining," in DSN'16: Proc. of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2016 pp.654-661. DOI 10.1109/DSN.2016.66
15. Jon Stearley, Adam Oliner. What Supercomputers Say: A Study of Five System Logs. Stanford University Department of Computer Science Palo Alto, CA 94305, USA oliner@cs.stanford.edu Sandia National Laboratories Albuquerque, NM 87111 USA, 25-28 June 2007, DOI: 10.1109/DSN.2007.103.

DETECTION ANOMALIES BID DATA IN UNSTRUCTURED SYSLOGS

Sheluhin O.I.[3], Ryabinin V.S. [4]

Log files, which are present in nearly all computer systems contain important information about the current

3 Oleg Sheluhin, Dr.Sc., Professor, MTUCI, Moscow, Russia. E-mail: sheluhin@mail.ru
4 Vladimir Ryabinin, Master, MTUCI, Moscow, Russia. E-mail: ryabvs@gmail.com

events. This information can be used to analyze the state of the system. Since the number of entries in such journals is usually very large, it is extremely difficult to find the necessary information in these files using traditional methods. Therefore, modern log analysis methods are needed to search for relevant data in log files. One of the modern approaches to the analysis of such data is the use of machine learning methods and big data mining. The problem when searching for critical, usually abnormal, the events in the log files is that the events without any context does not always provide sufficient information to identify the cause of problems.

The growing importance of analyzing log files in large computer systems requires the development of automated methods for processing unstructured data that allow extracting relevant information from large log files without the need for human intervention.

Analysis of the obtained results shows that the best results in the binary classification of big data anomalies are shown by the SVM algorithm . The nearest neighbor algorithm does not do well with high dimensional data, and therefore it cannot be recommended for solving the problem of detecting anomalies based on unstructured data from system logs in which the dimension of vectors is comparable to the number of selected log event patterns.

Keywords: *machine learning, text data processing, parsing, event logs, log files, unstructured data.*

References

1. Zou D., Qin H., Jin H., Qiang W., Han Z., Chen X. (2014) Improving Log-Based Fault Diagnosis by Log Classification. In: Hsu CH., Shi X., Salapura V. (eds) Network and Parallel Computing. NPC 2014. Lecture Notes in Computer Science, vol 8707. Springer, Berlin, Heidelberg. DOI https://doi.org/10.1007/978-3-662-44917-2_37
2. Min Du, Feifei Li, Guineng Zheng, Vivek Srikumar. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. CCS ,17 Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, DOI 10.1145/3133956.3134015
3. Daniel Dias Gonçalves. Automatic Diagnosis of Security Events in Complex Infrastructures using Logs. Instituto Superior Tecnico, Universidade de Lisboa May 2015
4. Christophe Bertero, Matthieu Roy,Carla Sauvanaud, Gilles Tredan. Experience Report: Log Mining Using Natural Language Processing and Application to Anomaly Detection. IEEE 28th International Symposium on Software Reliability Engineering (ISSRE), 2017, DOI: 10.1109/ISSRE.2017.43
5. Kristian Hunt. Log Analysis for Failure Diagnosis and Workload Prediction in Cloud Computing. STOCKHOLM, SWEDEN 2016 URN: urn:nbn:se:kth:diva-189186
6. Berkay Kicanaoglu. Unsupervised Anomaly Detection in Unstructured Log-Data for Root-Cause-Analysis. Tampere University of Technology, 2015 URN: http://URN.fi/URN:NBN:fi:tty-201507291465
7. Qiang Fu, Jian-Guang Lou, Yi Wang, Jiang Li. Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis. ICDM ,09 Proceedings of the 2009 Ninth IEEE International Conference on Data Mining, December 06 - 09, 2009, DOI: 10.1109/ICDM.2009.60.
8. Ryabinin V.S., Farmakovskiy M.A. Anomaly detection in computer system by intellectictual analysis of system. Sbornik trudov: XII Mezgdunarodnoia otraslevaya nauchno-tekhnicheskoj konferencii “Technologoia information obshesnva”, 14-15 mart 2018. i. V e: Bezopasnye informacionnye tekhnologii (BIT-2017). Sbornik trudov Vos’moj Vserossijskoj nauchno-tekhnicheskoj konferencii. MGTU im.N.EH.Baumana, 2017
9. Shilin He, Jieming Zhu, Pinjia He, Michael R. Lyu.Experience Report: System Log Analysis for Anomaly Detection. 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), 23-27 Oct. 2016, DOI: 10.1109/ISSRE.2016.21
10. Pinjia He, Jieming Zhu, Shilin He, Jian Li, Michael R. Lyu. An Evaluation Study on Log Parsing and Its Use in Log Mining. 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 28 June-1 July 2016, DOI: 10.1109/DSN.2016.66.
11. Wei Xu, Ling Huang, Armando Fox, David Patterson, Michael I. Jordan. Detecting Large-Scale System Problems by Mining Console Logs. Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles, October 11 - 14, 2009, DOI: 10.1145/1629575.1629587.
12. Tim Zwietasch. Detecting Anomalies in System Log Files using Machine Learning Techniques. University of Stuttgart, 2014. http://dx.doi.org/10.18419/opus-3454
13. Sheluhin O.I., Ryabinin V.S., Farmakovskiy M.A., Anomaly detection in computer system by intellectictual analysis of system journals, Voprosy kiberbezopasnosti, №2(26), 2018. DOI: 10.21681/2311-3456-2018-2-33-43
14. P. He, J. Zhu, S. He, J. Li, and M. R. Lyu. An evaluation study on log parsing and its use in log mining. In DSN’16: Proc. of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2016 pp.654-661. DOI 10.1109/DSN.2016.66
15. Jon Stearley, Adam Oline. What Supercomputers Say: A Study of Five System Logs. Stanford University Department of Computer Science Palo Alto, CA 94305, USA oliner@cs.stanford.edu Sandia National Laboratories Albuquerque, NM 87111 USA, 25-28 June 2007, DOI: 10.1109/DSN.2007.103.