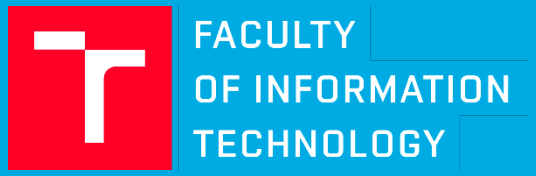

Log File Analysis and Anomaly Detection
PDS project
Academic year 2023/2024

Doc. Ing. Petr Matoušek, PhD., MA
Brno University of Technology, Faculty of Information Technology
Bozetechova 1/2, 612 66 Brno - Kralovo Pole
matousp@fit.vutbr.cz

What are log files?

- An ordered collection of system or application events (textual representation).
- Usually contains timestamp, src IP/port, event category, description, etc.
  - Event features: categorical, quantitative, unstructured text
- Generated by an operating system or an application
  - Unix log files (auth, messages, maillog), Windows logs
  - Services: web, ssh, mail, firewall, ids, dns, dhcp, radius

maillog:

Feb 12 04:14:55 pcmatousek sendmail[63661]: 41C3EtkO063661: from=root, size=2283, class=0, nrcpts=1, msgid=<202402120314.41C3EtkO063661@PCMATOUSEK.fit.vutbr.cz>, relay=root@localhost

messeges:

Jan 1 18:17:10 pcmatousek sshd[85693]: warning: /etc/hosts.allow, line 11: can't verify hostname: getaddrinfo(122.145.155.27.broad.fz.fj.dynamic.163data.com.cn, AF_INET) failed

auth.log:

Jan 28 07:13:08 pcmatousek sshd[85693]: refused connect from 45-79-168-172.ip.linodeusercontent.com (45.79.168.172)

http_log.log:

2005-05-04 17:16:12 2 45.110.2.82 200 TCP_HIT 941 729 GET http www.inmobus.com /wcm/assets/images/imagefileicon.gif - - DIRECT 38.112.92.20 image/gif "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)" PROXIED none - 192.16.170.42 SG-HTTP-Service - none –

radius-auth.log:

Sat Nov 28 03:34:36 2020: Login OK: [novotny@vut.cz] (User-Name novotny@vut.cz Client-Addr 147.229.122.3 CSID A2-38-3C-0B-99-86 CEID BC-EA-FA-F1-47-60: NAS wck/147.229.122.3) NAS-IP 147.229.122.1

Log file and log events analysis

- Description of the log dataset
  - Format, source, purpose, duration.
  - Number of events, event types, occurrence, uniqueness.
  - Feature selection, value range, uniqueness, distribution, importance.
  - Occurrence of events in time – frequency, distribution.

- Representation of events for machine processing
  - Preprocessing of features: log parsing, feature extraction, categorisation, dimension reduction.
  - Selection of features based on their uniqueness and importance.
  - Mapping of feature values to specific categories.

- Event annotation/labelling
  - Based on type, flags, system-assigned severity – error, failure, denied (syslog, IDS, firewall).
  - Supervised learning if an annotated dataset is available (labelled data).
  - Unsupervised learning if a dataset is not annotated -> clustering, labelling the clusters, scoring components (e.g., based on entropy).

Modeling log event behavior using ML methods

- One model vs. multiple sub-models
  - One model representing all events with their features.
  - Partitioning: dividing events into categories based on their type and creating independent behaviour models for each category.

- ML processing
  - The choice of a method depends on the nature of your dataset.
  - Classification models built on labelled datasets assign the class to an incoming event.
  - Outlier detection models built on unlabelled datasets assign the anomaly score or detect whether an event is an anomaly or not.

- ML methods applied to log event processing
  - Statistical methods: Gaussian distribution – 3 sigma, box plot, moving average [2].
  - Clustering and classification: k-means + XGBoost [3], DBSCAN + LSTM [4].
  - Formal languages: clustering + FSM [5].
  - Outlier detection: Isolation Forest [6].
  - Neural networks: Autoencoder [7], LSTM [8].

Modeling log event behavior using ML methods

- Building an ML model to represent log events
  - Selecting an ML model
  - Setting parameters of the model
  - Selecting a metric to measure whether an event fits the model
  - Choosing a threshold
  - Validating the model
  - Anomaly detection and evaluation

- Process of anomaly detection using ML model
  - Log collection
  - Log event preprocessing [9]
    - Splitting unstructured event records into keys or tokens.
    - Data normalization: min-max, Z-score, dimension reduction using PCA.
  - Feature extractions – manual or using automated methods (information gain, gain ratio)
  - Clustering – grouping of events based on their features, labelling of clusters
  - Detection: classification (for annotated data), anomaly score (for unannotated data)

Where to get log files for experiments

- Logs obtained from the local system – anonymization needed:
  - syslog, web access/error logs, maillog, ssh, dns, dhcp
  - unix log files at /var/log/ or windows logs
- Publicly available log datasets:
  - https://www.kaggle.com/datasets
  - https://ieee-dataport.org/
  - https://www.unb.ca/cic/datasets/index.html
  - https://www.westpoint.edu/centers-and-research/cyber-research-center/data-sets
  - https://onlineacademiccommunity.uvic.ca/isot/2022/11/25/cloud-security-datasets/
  - https://www.netresec.com
  - https://www.secrepo.com/
  - https://csr.lanl.gov/data/
  - http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
  - https://logpai.com/

PDS Project

Title: Log file analysis and anomaly detection.

Goal: To propose and implement a tool that analyses a large log file and detects anomalies using an ML method.

- The project will be carried out by the following steps:
  1. Create or select a log file dataset. Analyse the dataset, explore the log events and the features.
  2. Preprocess the data for machine learning.
  3. Select features to represent normal behaviour. Select an ML method to model the dataset.
  4. Implement a classification model using available libraries, tune its parameters.
  5. Select a metric and a threshold for classification and anomaly detection. Validate the model.
  6. Provide experiments with the model. Describe its advantages and limitations.
  7. Write the project report (see the recommended structure below).
  8. Submit the project (source code + report in PDF + dataset) via BUT IS.

PDS Project

- Project deadline: 22nd April 2024 (hard deadline).
- Individual project registration via BUT IS by 29th February.
- Maximum points: 25
- Online consultation possible via Moodle News.
- Individual project – each student creates their own solution.
- Partial solutions will be accepted. Parts not implemented must be explained in the Readme.txt.
- Required deliverables:
  1. Log analyser *log-monitor* -> running as a CLI application on a Unix system.
  2. Log files for training and testing.
  3. Report in PDF format following the required structure.
- Plagiarism is prohibited – see Copyright and Publication Policy.

1) Log file description and analysis

1. Create or download a log file for analysis and experimentation.
   - The log file should cover at least 1 week of communication.
2. Describe the log file, analyse the log events.
   a) Format of event records, event types, their occurrence, uniqueness.
   b) Available features: type, value distribution, uniqueness, importance.
3. Select features to represent log events in the model.
   a) Feature selection, normalisation, reduction.
   b) Feature representation for machine learning.
4. Implement log file preprocessing -> tool *log-monitor.*
   - *Can be implemented in C/C++/Python on Unix/Linux OS.*
   - *The tool should run on the Unix command line (CLI).*
5. Log event annotation/labelling
   - Supervised learning – annotation based on assigned event types or flags.
   - Unsupervised learning – working with an unlabelled dataset.

2) Implement a classification tool

1. Select an ML method for classification based on recommended resources, see references.
2. Implement a *log-monitor* that trains the model and detects anomalies.
3. The tool reads log files (training and testing data) provided with the project.
4. The tool has the following syntax (mandatory format):

log-monitor -training <file> -testing <file> -<params>

-training <file>: a data set used to train the model

-testing <file>: a data set used for testing the classification

-<params>: a list of parameters required for the specific model (threshold, time window) etc. in the format par1=val1, par2=val2, …

output: a list of anomalies with their score or classified log events

5. Additional files: *Makefile, Readme.txt*
   - A list of submitted files with a short description.
   - Description of how to compile and run the tool, explanation of the parameters.
   - Example of running the tool
   - The Makefile with installation of required libraries, if needed.

3) Write the report in EN/CZ/SK (5-10 pages)

Suggested document structure:

1. Introduction to the log file analysis (purpose, methods).
2. Description and analysis of the log dataset.
   - Analyse and explore log events, select features.
3. Modelling log events
   - Data pre-processing, feature selection, event representation.
   - ML model selection, its parameters and threshold.
4. Tool implementation and experiments.
   - Description of the implemented tool, its behaviour, input parameters.
   - Experiments: training, validation, threshold setting, detection.
5. Testing of the application on available datasets, results, evaluation.
6. Discussion of the results.
7. Conclusion and contribution.

To create a document, use the BSc/MSc template, see https://www.fit.vut.cz/study/theses/bachelor-theses/.

4) Project submission

1. Submit a zip file named *xlogin.zip* containing the following files:
   - Readme.txt – with your name, login, a list of files submitted, description of how to install and run your tool.
   - The project report in PDF format (*xlogin.pdf* file).
   - The source code of your tool *log-monitor*.
   - Datasets used for testing and training.
     - If a dataset is too large, share it online (Google drive, BUT drive) and provide a link in the Readme.txt file.

Concluding remarks

- The aim of the project is to demonstrate your ability to automatically process and analyse large log files using advanced ML techniques.
- The focus is on individual solution, data processing and analysis.
- The project includes
  - experimental part,
  - analysis part,
  - proposal of a detection method,
  - implementation part,
  - testing, evaluation, and discussion.
- Innovative approaches in any of these parts are highly appreciated.

- Any external tools, code, information sources must be properly referenced, otherwise the work will be considered as plagiarism.

References

[1] Jiawei Han, Micheline Kamber, and Jian Pei. Data Mining: Concepts and Techniques (3rd. ed.). 2011. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[2] Siwoon Son, Myeong-Seon Gil and Y. -S. Moon, "Anomaly detection for big log data using a Hadoop ecosystem," *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju, Korea (South), 2017, pp. 377-380

[3] Henriques, J.; Caldeira, F.; Cruz, T.; Simões, P. Combining K-Means and XGBoost Models for Anomaly Detection Using Log Datasets. Electronics 2020, 9.

[4] C. Egersdoerfer, D. Zhang and D. Dai, "ClusterLog: Clustering Logs for Effective Log-based Anomaly Detection," 2022 IEEE/ACM 12th Workshop on Fault Tolerance for HPC at eXtreme Scale, USA, 2022.

[5] Q. Fu, J. -G. Lou, Y. Wang and J. Li, "Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis," 2009 Ninth IEEE International Conference on Data Mining, USA, 2009.

[6] Amir Farzad, T. Aaron Gulliver, Unsupervised log message anomaly detection, ICT Express, Volume 6, Issue 3, 2020, Pages 229-237, ISSN 2405-9595.

[7] Marta Catillo, Antonio Pecchia, Umberto Villano: AutoLog: Anomaly detection by deep autoencoding of system logs, Expert Systems with Applications, Volume 191, 2022, ISSN 0957-4174.

[8] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. 2017. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). ACM, New York, NY, USA.

[9] S. He, J. Zhu, P. He and M. R. Lyu, "Experience Report: System Log Analysis for Anomaly Detection," *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, Ottawa, ON, Canada, 2016, pp. 207-21