Explainable AI for Log-Based Anomaly Detection in Security Monitoring: Reasoning Pipelines and Cross-Dataset Evaluation.

University Of Oulu
Faculty of Information Technology and Electrical Engineering
Master's Thesis

Emmanuel Ikwunna
10th November 2025

Abstract

Abstract is needed to sum the master's thesis up. The abstract is to be uploaded into Optima before the final grading of the thesis. Please find the current information about the format given in Optima.

The guide includes instructions for students. It is written keeping in mind the idea that the user may utilise it e.g. by pasting his or her text on the current text. The contents include information about formatting the text, positioning tables and figures, among other things. In addition, the use of proper literature is instructed. Even if there is no strict structure for the thesis, a recommendation is offered in this guideline.

One important guideline for the text is that do not write too short paragraphs. For instance, if there is only one sentence in a paragraph, the sentence must be really important and influential to form a paragraph of its own.

It is not possible to provide information in a guideline like this for all issues related to master's thesis. For example, the research process, ways to acquire research material and its analysis are excluded in the guideline. On the other hand, a structure for a research plan is provided in the appendices.

Keywords
first keyword, second keyword, other keywords

Supervisor
Title, position First name Last name

Foreword

The foreword is not instructed by the supervisors. In other words, the student may write in this section what she or he wants to share with readers. However, it is a custom to thank all those who have contributed to the research somehow. When acknowledging people, their affiliations are given (e.g. Professor, University Lecturer, Adjunct Professor, Mrs.) This guideline is based on the previous version that was written in Finnish and finalised by Dr. Lasse Harjumaa in January 2007. This version is to replace the earlier version. I want to thank all those people who have contributed to the earlier versions and this newest version, the first written in English. Hopefully this guideline will serve both students and faculty with its instructions that include both formal and informal regulations and recommendations. In the first phase, the constructive comments are received with pleasure by raija.halonen@oulu.fi. Oulu, January 10, 2011

Raija Halonen Oulu, March 10, 2020

Contents

1 Introduction

In the thesis we follow the style introduced by The American Psychological Association (APA). The APA style can be found easily in the Internet and some sites provide a quick guide, too. E.g. `http://www.waikato.ac.nz/library/learning/g_apaguide.shtml` and `http://owl.english.purdue.edu/owl/resource/560/01/` are useful links.

It is important to follow given instructions. In academic theses, not only the content but also the format is important. Generally every academic publication forum requires that the publications follow their guidelines. In the theses accepted in the Department of Information Processing Science the format is APA. Currently there are several editions published from APA. The general rule is that the latest available edition is applied. Currently the newest edition is 6th. If a thesis is already in process it is not needed to transfer it into a newer edition of APA. Whichever you apply, do it consistently.

In addition to teach the students to follow given formal instructions, the guideline aims to unify and standardise the outlook of the theses made in the department. The guideline also enables the supervisors to focus on the content of the theses as the students already consider the outlook and format themselves. In this sense, it is a question of available resources for supervision and guidance.

The use of language and grammar cannot be discussed in detail in this kind of guide. However, the writing style should meet the general academic writing styles in the sense that no causeries are accepted or other lightweight texts such as jokes or rumblings. In other words, in academic theses all writing must be appropriate and reasonable. There are several guidebooks for academic writing available in the Oulu University Library, for example, and in the Internet. For those who write their thesis in Finnish there are books such as Tieteellinen kirjoittaminen. The style reference by APA (American Psychological Association, 2010) offers fruitful practical hints for writing thesis in English.

As the guideline is written according to the instructions, it enables the students to copy their text (without format) on the document and thus get their text into the right format. The format is to be used in the Bachelor's Theses and in the Master's Theses. In case of other theses, essays or reports it is recommended that the students inquire their teachers if the guideline is to be followed or not.

The structure of the guideline is as follows. The formal instructions for different topics are presented next. This is followed by examples of references and their use. After that the structure of theses and its writing style is discussed briefly. The guideline ends with a summary.

2 Background

2.1 Introduction to Log Anomaly Detection

2.1.1 Overview and Challenges

The information within system log files is fundamental to monitoring the stability and security of networked systems, which generate these logs ubiquitously [1, 2]. Logs provide this value by offering a detailed, chronological record of both runtime system events and user intentions [3]. Consequently, as system complexity grows, these logs have become a critical asset for operations such as performance monitoring, security auditing, transaction tracing, and fault diagnosis [4].

The primary purpose of log anomaly detection is to protect digital infrastructures by identifying abnormal activities, such as network intrusions, from the enormous volumes of event logs. In this context, anomalies represent log patterns that significantly deviate from the expected behavior of the system. Detecting these deviations is crucial for maintaining system reliability and preventing severe disruptions or financial losses, as global cybercrime costs are estimated to reach trillions of euros annually [5].

Log analysis, and consequently anomaly detection, faces several significant challenges primarily driven by the nature and scale of the data:

- **Volume:** System logs are large-scale data collected in real-time [6]. The sheer volume of logs has grown rapidly, often reaching 50 GB (120–200 million lines) per hour for large-scale services, making manual inspection and traditional processing infeasible [3].

- **Variety and Complexity:** Logs are typically unstructured or semi-structured text files generated by logging statements in source code [2]. Because developers are allowed to write free-text messages, the format and semantics of logs vary significantly across systems, leading to high-dimensional features with complex interrelationships. This complexity and diversity increase the difficulty of accurate anomaly detection [7].

- **Velocity (Timeliness):** For anomaly detection to be useful, it must be timely, requiring decisions to be made in a streaming fashion to allow users to intervene in ongoing attacks or performance issues. Offline methods that require multiple passes over the entire log data are thus unsuitable for real-time security monitoring [2].

Due to the challenges of volume and complexity,

the adoption of automated log analysis has become imperative to efficiently process and interpret vast corpora of logs.

2.1.2 Traditional Methods

Early log anomaly detection efforts relied heavily on human expertise [4]. As the volume of logs grew, research shifted toward automated, data-driven methods, broadly categorized into rule-based systems and statistical approaches, many of which depend on logs first being converted into a structured format through a process known as log parsing [7].

Log parsing is a critical precursor step where raw, unstructured log messages are

transformed into structured data, typically by extracting a constant part, called the log template (or log key), and identifying the variable parts (parameters).

The parser Spell, for example, is an online streaming parser that utilizes the Longest Common Subsequence (LCS) technique to dynamically identify and update log patterns. Tools like DeepLog rely on log parsing methods like Spell to generate log templates for their inputs [2, 8].

2.1.2.1 Rule-Based Systems (Regex, Signatures)

Rule-based methodologies were among the first attempts to automate log analysis to reduce human error. These methods typically rely on explicitly defined rules, patterns, or known indicators of abnormal behavior, often requiring specific domain knowledge from human experts.

- **Keyword Matching and Regular Expressions (Regex):** Early rule-based systems focused on matching specific keywords (e.g., ”error,” ”failed”) or using regular expressions to flag anomalous log entries [9]. However, relying solely on keywords or structural features often prevents a large portion of log anomalies from being detected and can lead to unnecessary alarms (alarm fatigue) if the system constantly evolves [9, 10]. Furthermore, manually designing and maintaining regular expressions is prohibitive given the rapid increase in log volume and frequent system updates [7].

- **Invariant Mining (IM):** Invariant mining is another traditional approach that captures co-occurrence patterns between different log keys [2, 10]. This method defines a window (time or session based) and detects whether certain mined quantitative relationships, or invariants, hold true within that window (e.g., ensuring that the count of ”file open” logs equals the count of ”file close” logs in a normal condition) [10]. IM is typically characterized as an unsupervised offline method [2].

2.1.2.2 Statistical Methods (Clustering, PCA)

Statistical methods leverage mathematical principles to identify normal patterns from data volumes and flag deviations statistically likely to be anomalous. These methods generally operate on a generated numeric vector representation of the logs, often discarding parameter values and only using log keys and their counts [2]. Most statistical methods rely on initial log parsing, where raw log messages are converted into structured, numeric representations, such as event count vectors [11].

Principal Component Analysis (PCA):

PCA is a widely used statistical technique in log anomaly detection. PCA is a linear transformation technique used to transform a set of correlated variables into a set of uncorrelated variables, known as principal components [6, 12].

Application in Log Analysis:

- **Vectorization:** PCA is applied at the session level, where log entries are grouped by an identifier (e.g., block_id in HDFS or instance_id in OpenStack) [11, 2]. Each session is converted into an event count vector that records how often each unique log key occurs. These vectors together form a feature matrix, with rows as sessions and columns as log keys [11, 2].

- **Dimensionality Reduction:** PCA projects the high-dimensional counting matrix into a lower-dimensional space by identifying components that capture the most variance [4]. It produces two subspaces: the normal space (S_n), formed by the first k principal components, and the anomaly space (S_a), composed of the remaining ones [11].

- **Anomaly Detection:** PCA generates a normal space (S_n) using the first k principal components and an anomaly space (S_a) using the remaining dimensions [11]. An abnormal session is detected by measuring its projection length (quantified by the Squared Prediction Error, SPE) onto this residual subspace S_a [11, 9, 2].

Clustering Methods

Clustering algorithms aim to group data instances that are similar to each other. In log analysis, these methods are used to group log messages to generate event templates (log parsing) or to detect anomalies by identifying outliers that do not belong to normal clusters [13, 12].

- **LogCluster:** LogCluster is an unsupervised clustering method that groups textually similar log messages to detect frequent line patterns and abnormal events in textual logs [10, 6]. It vectorizes log sequences using Inverse Document Frequency (IDF) scaling and builds a knowledge base of normal and abnormal clusters [11, 4]. A new sequence is classified as an anomaly if its distance to the nearest cluster centroid exceeds a threshold [4].

- **Density-Based Methods:** The Simple Logfile Clustering Tool (SLCT) uses a density-based approach to group log messages that occur frequently, forming clusters that represent common line patterns [14]. Log entries that do not fit into these clusters are considered outliers and may indicate rare or abnormal events [14]. Related methods, such as HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), are also used in some cases to provide pseudo-labels for unlabeled data in semi-supervised learning [4].

2.1.3 Machine Learning Approaches

Following traditional statistical and rule-based methods, log anomaly detection quickly adopted general Machine Learning (ML) algorithms. These approaches rely heavily on the preceding log parsing and vectorization steps to transform unstructured log files into numerical features [1]. Traditional ML techniques are generally categorized into supervised models, which require labeled data, and unsupervised models, which are crucial given that log data is overwhelmingly unlabeled in real-world scenarios [9, 6].

2.1.3.1 Classification Models (SVM, Random Forest)

Supervised classification models require labeled log data that indicate normal and anomalous behavior [13].

- **Support Vector Machines (SVM):** SVM is a flexible model capable of linear or nonlinear classification, regression, and outlier detection. For log analysis, event count matrices are commonly used as input. SVMs can be adapted for multi-class problems using strategies like one-versus-the-rest [12]. The **One-Class SVM (OCSVM)** is an unsupervised variant trained only on normal data to identify anomalies that fall outside a learned boundary [15].

- **Random Forests (RF):** Random Forest is an ensemble method that builds multiple decision trees on random subsets of features and data, then combines their predictions [21]. RF models are effective with minimal parameter tuning and do not require input scaling [16]. In log anomaly detection, Random Forest often outperforms other classifiers and can be integrated into systems like **HuntGPT** [17] alongside Explainable AI techniques for enhanced threat analysis.

2.1.3.2 Unsupervised Methods (Isolation Forest)

Unsupervised methods are critical for log analysis because acquiring labeled data is expensive, and most real-world log data is inherently unlabeled [9, 6].

- **Isolation Forest (iForest):** iForest is a tree-based unsupervised learning algorithm specifically designed for anomaly detection [18]. Unlike clustering methods that seek dense regions, iForest focuses on isolating observations that are distinct from the remaining input data [6, 12]. This is achieved by forming an ensemble of decision trees that partition the data; outliers are isolated more quickly (with fewer partitions) than normal instances [6].

2.1.4 Deep Learning Models

Deep Learning (DL) approaches use neural networks to automatically extract features and capture complex patterns from logs, offering advantages for high-dimensional and large-scale data [4]. DL methods usually follow a pipeline of preprocessing, parsing, vectorization, and neural network classification [19].

2.1.4.1 LSTM-based Models

Recurrent Neural Networks (RNNs), especially with **Long Short-Term Memory (LSTM)** units, are widely used for sequential log data analysis [9].

- **DeepLog (Du et al., 2017):** DeepLog is an early deep learning framework for online log anomaly detection. It treats logs as sequences similar to natural language and uses an LSTM trained on log keys derived from parsing tools like Spell [2, 8]. The model predicts the next log key in a sequence to learn normal patterns, and any log key with low prediction probability is flagged as anomalous [2]. Detection occurs at the level of individual log entries, and the model can incorporate parameter values and time intervals to identify performance-related anomalies [1, 2].

- **LogAnomaly and LogRobust:** LogAnomaly combines LSTM with *template2vec* embeddings to capture sequential and quantitative anomalies. LogRobust uses an attention-based Bi-LSTM with Word2Vec representations to capture bidirectional and semantic relationships among log events.

2.1.4.2 Transformer-based Models

Transformer architectures, originally from NLP, capture long-range dependencies and semantic context in logs.

- **LogBERT:** Applies the BERT architecture to encode log messages into semantic embeddings. These embeddings are then used by a Transformer-based classifier to detect anomalies. Similar approaches, such as **NeuralLog**, bypass traditional parsing and work directly on raw logs.

- **LogFiT:** Fine-tunes pre-trained BERT models on system logs using self-supervised objectives like Masked Token Prediction combined with centroid distance minimization.

2.1.4.3 Attention Mechanisms

Attention mechanisms enable models to prioritize the importance of different log entries, thereby enhancing feature extraction for anomaly detection.

- **Multi-Head Attention:** Enables the model to track multiple patterns simultaneously.
- Attention is used in Bi-LSTM models like LogRobust and hybrid architectures like **LogCTBL** (CNN-TCN-Bi-LSTM + BERT) to enhance feature representation.
- Attention scores can also aid Explainable AI (XAI) by highlighting which log events contribute most to anomaly predictions.

2.1.4.4 Performance

Deep learning models typically achieve high F1 scores based on their ability to model sequential and semantic patterns. Representative results include:

Model	Architecture	Dataset	F1 Score
LogCTBL	Hybrid (CNN-TCN-Bi-LSTM + BERT)	BGL	0.9987
LogCTBL	Hybrid (CNN-TCN-Bi-LSTM + BERT)	Thunderbird	0.9978
SiaLog	Siamese Network (LSTM)	HDFS	0.99
SiaLog	Siamese Network (LSTM)	BGL	0.99
OneLog	HCNN (Character-based)	HDFS	0.99
OneLog	HCNN (Character-based)	BGL	0.99
DeepLog	LSTM	HDFS	0.85
DeepLog	LSTM	BGL	0.86
LogAnomaly	LSTM + Template2Vec	BGL	0.88
LogBERT	Transformer (BERT)	BGL	0.91

Table 1: Comparison of deep learning log anomaly detection models.

References

[1] Crispin Almodovar, Fariza Sabrina, Sarvnaz Karimi and Salahuddin Azad. 'Can Language Models Help in System Security? Investigating Log Anomaly Detection using BERT'. In: *Proceedings of the 20th Annual Workshop of the Australasian Language Technology Association*. Ed. by Pradeesh Parameswaran, Jennifer Biggs and David Powers. Adelaide, Australia: Australasian Language Technology Association, Dec. 2022, pp. 139–147. URL: https://aclanthology.org/2022.alta-1.19/.

[2] Min Du, Feifei Li, Guineng Zheng and Vivek Srikumar. '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. Dallas, Texas, USA: Association for Computing Machinery, 2017, pp. 1285–1298. ISBN: 9781450349468. DOI: 10.1145/3133956.3134015. URL: https://doi.org/10.1145/3133956.3134015.

[3] Hongcheng Guo, Jian Yang, Jiaheng Liu, Jiaqi Bai, Boyang Wang, Zhoujun Li, Tieqiao Zheng, Bo Zhang, Junran peng and Qi Tian. *LogFormer: A Pre-train and Tuning Pipeline for Log Anomaly Detection*. 2024. arXiv: 2401.04749 [cs.LG]. URL: https://arxiv.org/abs/2401.04749.

[4] Hong Huang, Wengang Luo, Yunfei Wang, Yinghang Zhou and Weitao Huang. 'LogCTBL: a hybrid deep learning model for log-based anomaly detection'. In: *The Journal of Supercomputing* 81 (Jan. 2025). DOI: 10.1007/s11227-025-06926-3.

[5] European Parliament. *Cybercrime in the EU: Threats, Trends and Policy Responses*. Tech. rep. According to an EU briefing, the annual global cost of cybercrime was estimated at approximately €5.5 trillion in recent years. Accessed November 9, 2025. European Parliamentary Research Service, 2024. URL: https://www.europarl.europa.eu/RegData/etudes/BRIE/2024/760356/EPRS_BRI(2024)760356_EN.pdf.

[6] Yukyung Lee, Jina Kim and Pilsung Kang. 'LAnoBERT: System log anomaly detection based on BERT masked language model'. In: *Applied Soft Computing* 146 (2023), p. 110689. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2023.110689. URL: https://www.sciencedirect.com/science/article/pii/S156849462300707X.

[7] Pinjia He, Jieming Zhu, Zibin Zheng and Michael R. Lyu. 'Drain: An Online Log Parsing Approach with Fixed Depth Tree'. In: *2017 IEEE International Conference on Web Services (ICWS)*. 2017, pp. 33–40. DOI: 10.1109/ICWS.2017.13.

[8] Min Du and Feifei Li. 'Spell: Streaming Parsing of System Event Logs'. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 2016, pp. 859–864. DOI: 10.1109/ICDM.2016.0103.

[9] Harold Ott, Jasmin Bogatinovski, Alexander Acker, Sasho Nedelkoski and Odej Kao. *Robust and Transferable Anomaly Detection in Log Data using Pre-Trained Language Models*. 2021. arXiv: 2102.11570 [cs.AI]. URL: https://arxiv.org/abs/2102.11570.

[10] Weibin Meng, Ying Liu, Yichen Zhu, Shenglin Zhang, Dan Pei, Yuqing Liu, Yihao Chen, Ruizhi Zhang, Shimin Tao, Pei Sun and Rong Zhou. 'LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs'. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 4739–4745. DOI: 10.24963/ijcai.2019/658. URL: https://doi.org/10.24963/ijcai.2019/658.

[11] Shilin He, Jieming Zhu, Pinjia He and Michael R. Lyu. 'Experience Report: System Log Analysis for Anomaly Detection'. In: *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*. 2016, pp. 207–218. DOI: 10.1109/ISSRE.2016.21.

[12] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd. Sebastopol, CA: O'Reilly Media, 2019. ISBN: 978-1-492-03264-9. URL: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/.

[13] Jay Alammar and Maarten Grootendorst. *Hands-On Large Language Models: Language Understanding and Generation*. Sebastopol, CA: O'Reilly Media, 2024. ISBN: 978-1-492-08828-3. URL: https://www.oreilly.com/library/view/hands-on-large-language/9781492088283/.

[14] Risto Vaarandi. 'Mining event logs with SLCT and LogHound'. In: May 2008, pp. 1071–1074. DOI: 10.1109/NOMS.2008.4575281.

[15] Mennatallah Amer, Markus Goldstein and Slim Abdennadher. 'Enhancing one-class Support Vector Machines for unsupervised anomaly detection'. In: Aug. 2013, pp. 8–15. DOI: 10.1145/2500853.2500857.

[16] Andreas C. Müller and Sarah Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. Sebastopol, CA: O'Reilly Media, 2016. ISBN: 978-1449369415.

[17] Tarek Ali. 'Next-Generation Intrusion Detection Systems with LLMs: Real-Time Anomaly Detection, Explainable AI, and Adaptive Data Generation'. Master's thesis. Oulu, Finland: Faculty of Information Technology and Electrical Engineering, University of Oulu, 2025.

[18] Dong Xu, Yanjun Wang, Yulong Meng and Ziying Zhang. 'An Improved Data Anomaly Detection Method Based on Isolation Forest'. In: Dec. 2017, pp. 287–291. DOI: 10.1109/ISCID.2017.202.

[19] Sayedshayan Hashemi Hosseinabad. 'Data-Driven Software System Log Anomaly Detection'. Acta Univ. Oul. A 808. Doctoral dissertation. Oulu, Finland: University of Oulu, Faculty of Information Technology and Electrical Engineering, 2025. ISBN: 978-952-62-4502-7.

[20] Runqiang Zang, Hongcheng Guo, Jian Yang, Jiaheng Liu, Zhoujun Li, Tieqiao Zheng, Xu Shi, Liangfan Zheng and Bo Zhang. *MLAD: A Unified Model for Multi-system Log Anomaly Detection*. 2024. arXiv: 2401.07655 [cs.SE]. URL: https://arxiv.org/abs/2401.07655.

[21] Chiung Ko, Jintaek Kang, Chaejun Lim, Donggeun Kim and Minwoo Lee. ‘Application of Machine Learning Models in the Estimation of Quercus mongolica Stem Profiles’. In: *Forests* 16.7 (2025). ISSN: 1999-4907. DOI: 10.3390/f16071138. URL: https://www.mdpi.com/1999-4907/16/7/1138.

Appendix A Structure for the research plan

A research plan can be reported according to the next structure. The order of the items is important.

Introduction

The topic is introduced on general level. The context of the research is described and the research problem is explained and justified. The problem is situated in its larger environment. Note references when needed. The researcher may reason the topic also by describing his or her personal motivation.

Research problem and research methods

The problem under study is explained as explicitly as possible. The research problem can be divided into sub problems or presented as hypotheses. The research methods and analysis are described.

Limitations

The planned limitations and known shortcomings are reported. The reasons for them – if known – are explained from the viewpoint of the current research.

Preliminary earlier research

The prior literature is presented briefly with full sentences. All required references are included. Its relevance in the current research is described and limitations recognised in prior research are identified if possible. List of main prior literature in relation to the background theory Main background references are listed in the required format (APA).

Lähteet

Timetable

A plan to describe the planned research related to calendar time. It is recommended that the plan is discussed with supervisor to ensure enough milestones for checking thoroughly the status of the thesis.

Preliminary structure of contents

1. Introduction
2. Glossary
3. Prior research
   (a) First
   (b) Second
       Subsecond
4. Sources