Proposal Bachelor's Thesis

Development and Evaluation of a Hybrid Approach for Automated Error Detection and Classification in LoRaWAN-Based IoT Data Pipelines Using Log Data

Emir Hamulic

University of Applied Sciences Technikum Wien

Student ID: wi23b168
Study Program: B.Sc. Business Informatics
Advisor: Rohatsch Lukas, MSc
Version: 0.1

Date: January 17, 2026

1. Problem Area

1.1. Operational Context

LoRaWAN-based IoT deployments are typically operated as distributed, multi-component systems. End devices communicate with gateways, which forward packets to a LoRaWAN Network Server (LNS). The LNS then routes data to application components or ETL-Pipelines which process data into structured data for databases or dashboards. This "star-of-stars" topology and the central role of the Network Server are core characteristics of LoRaWAN networks. [1]

1.2. Log Volume and Monitoring Limit

Modern IT and IoT backends generate very large volumes of logs. Multiple studies show that manual inspection becomes impractical at scale and is only possible to archive with automation of log analysis and diagnosis.

Practitioner evidence reinforces this: in a large survey on log anomaly detection, practitioners explicitly cite the need to "efficiently analyze large volumes of log data," and many expect tools to handle at least 100,000 logs while still delivering near-real-time results. [2]

1.3. Heterogenity and Missing Structure

Operational logs are typically semi-structured, highly heterogeneous, and produced by many components in an interleaved manner. A widely adopted first step in automated log analysis is log parsing (template extraction) to convert raw messages into structured events.

Research further shows that diverse formatting and log types create significant challenges when applying ML methods directly to real-world logs, and that preprocessing/structuring decisions materially impact downstream analysis quality.[3]

1.4. Missing Prioritization

Raw logs typically provide neither a consistent prioritization scheme nor decision-support artifacts such as confidence estimates, rationale, or action guidance. In practice, this forces operators manually looking through and link alerts, driving up the MTTR and operational cost. [2]

1.5. Rule-Based Detection Systems

Rule-based systems and expert-defined patterns are effective for known signatures and for encoding domain knowledge, but they require ongoing effort and are limited when log messages vary substantially or new patterns appear. Early log anomaly detection often relied on rule-based or pattern-driven approaches, while recent work increasingly applies deep neural networks to learn normal behavior and handle patterns in log data.[4]

1.6. ML-Based Detection

The ML component is primarily used as a mechanism to detect and prioritize *unknown* events. Instead of requiring an explicit signature for each failure, anomaly detection techniques learn a baseline of normal log behavior and flag deviations for operational attention. [4, 5] This is particularly relevant for long-tail failures such as new stack traces, unexpected integration errors, or previously unseen combinations of events.[6]

A Random Forest is a comparatively lightweight classifier for tabular log features. It aggregates many decision trees via majority vote and can output class probability estimates, which makes it suitable for operational classification without the complexity of deep models. [7, 8] In a hybrid pipeline, these probability estimates can be used to implement a reject option for unknown events: if the maximum predicted class probability falls below a threshold, the event is tagged as *Unknown* and routed to aggregation and rule-base enrichment. [7, 9]

To handle unknown-event in a hybrid classifier, the ML subsystem should not force every input into one of the known error classes. In open-set recognition, inputs from unseen classes must be rejected or mapped to an "unknown" category based on confidence or risk-aware decision rules. [10]

2. Task Description

3. Research Questions / Hypotheses

Main Research Question: Sub-Questions:

4. Methodology

5. Kind of expected results

5.1. Overview Research Questions - Methodology - Expected Results

Research questions / (hypotheses)	**Method(s) per research question**	**Expected kind of result per method**
RQ1: ...	...	...
RQ1: ...	...	...
RQ2: ...	...	...
RQ3: ...	...	...
RQ4: ...	...	...

Table 1: Mapping research questions to methods and expected results.

6. Timetable

6.1. Milestone Plan

Nr	Name	Plan	Adapted by	Actual Date
M1	...	...	...	...
M2	...	...	...	...
M3	...	...	...	...
M4	...	...	...	...
M5	...	...	...	...

Table 2: Milestones plan vs. actuals.

References

[1] LoRa Alliance, *LoRaWAN® Specification v1.1*, Online technical specification, Accessed 2026-01-17. [Online]. Available: `https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1`

[2] X. Ma et al., *Practitioners' expectations on log anomaly detection*, arXiv preprint, 2024. arXiv: `2412.01066 [cs.SE]`. [Online]. Available: `https://arxiv.org/abs/2412.01066`

[3] P. He, J. Zhu, Z. Zheng, and M. 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` [Online]. Available: `https://jiemingzhu.github.io/pub/pjhe_icws2017.pdf`

[4] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," *Machine Learning with Applications*, vol. 12, p. 100 470, 2023. DOI: `10.1016/j.mlwa.2023.100470` [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2666827023000233`

[5] M. Du, F. Li, G. Zheng, and V. 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)*, 2017, p. 1285. DOI: `10.1145/3133956.3134015`

[6] X. Ma et al., *Practitioners' expectations on log anomaly detection*, Dec. 2024. DOI: `10.48550/arXiv.2412.01066` [Online]. Available: `https://www.researchgate.net/publication/386375463_Practitioners'_Expectations_on_Log_Anomaly_Detection`

[7] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324 [Online]. Available: https://link.springer.com/article/10.1023/A:1010933404324

[8] scikit-learn developers, *Randomforestclassifier - scikit-learn documentation*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassif html, Accessed 2026-01-17.

[9] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, “Open-set recognition: A good closed-set classifier is all you need?” In *International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://arxiv.org/abs/2110.06207

[10] W. J. Scheirer, L. P. Jain, and T. E. Boult, “Probability models for open set recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014. DOI: 10.1109/TPAMI.2014.2321392 [Online]. Available: https://ieeexplore.ieee.org/document/6809169