Mälardalen University Licentiate Thesis 369

SMART-TROUBLESHOOTING IN INDUSTRY 4.0 LEVERAGING LOG FILES AND PRODUCT INFORMATION

Sania Partovian

Mälardalen University

Mälardalen University Press Licentiate Theses
No. 369

SMART-TROUBLESHOOTING IN INDUSTRY 4.0 LEVERAGING LOG FILES AND PRODUCT INFORMATION

Sania Partovian

2025

School of Innovation, Design and Engineering

Copyright © Sania Partovian, 2025
ISBN 978-91-7485-694-1
ISSN 1651-9256
Printed by E-Print AB, Stockholm, Sweden

“To my family
& Nader”

Abstract

Connected internet of things devices are becoming more powerful, yet the challenge of effectively managing them to prevent failures remains ongoing. In some scenarios, such as with devices produced by a single company, it is possible to use the same fault detection provided by the manufacturer and follow the instruction for resolving the threat. In other cases, such as with devices produced by different companies, the heterogeneity of devices necessitates a more detailed and complex approach to fault detection. Log files, which are records of events or processes generated by a device's software or hardware, are crucial for monitoring device behavior. It is important to consider the diversity of log files and data to detect threats, identify their root causes, and provide effective solutions. In the realm of troubleshooting interconnected internet of things devices, current solutions predominantly address homogeneous device environments, which limits their scalability and adaptability to diverse device types and configurations. Instead, a more flexible approach is needed; one that can accommodate a variety of connected devices while minimizing reliance on specific company instructions. One such method is Smart-troubleshooting which involves a 4-step cycle which include prevention, detection and diagnosis, recovery, and evolution of threats. Given these premises, the ultimate goal of this research is to define a smart-troubleshooting approach based on log files and product information. By leveraging a generalized methodology, this approach seeks to enhance the management of internet of things systems in complex, multi-manufacturer environments. This thesis focuses on a systematic review of log files and the state of the art in troubleshooting methodologies. During the research, the scarcity of publicly available log files for troubleshooting purposes was identi-

fied. Consequently, a method was proposed for generating synthetic log files using generative adversarial networks. The proposed methodology leverages these log files along with product information to enhance smart-troubleshooting. To validate the approach and gather industry feedback, questionnaires and interviews was conducted. Following this, machine learning algorithms will be employed to implement and refine the proposed method. By leveraging a generalized methodology, this approach seeks to improve the management and fault detection of internet of things systems in complex, heterogeneous environments.

Sammanfattning

Uppkopplade internet of things-enheter blir mer kraftfulla, men utmaningen att effektivt hantera dem för att förhindra fel kvarstår. I vissa scenarier, till exempel med enheter som produceras av ett enda företag, är det möjligt att använda samma feldetektering som tillhandahålls av tillverkaren och följa instruktionerna för att lösa hotet. I andra fall, t.ex. med enheter som produceras av olika företag, kräver enheternas heterogenitet en mer detaljerad och komplex metod för att upptäcka fel. Loggfiler, som är register över händelser eller processer som genereras av en enhets mjukvara eller hårdvara, är avgörande för att övervaka enhetens beteende. Det är viktigt att överväga mångfalden av loggfiler och data för att upptäcka hot, identifiera deras grundorsaker och tillhandahålla effektiva lösningar. Inom området för felsökning av sammankopplade internet of things-enheter, adresserar nuvarande lösningar huvudsakligen homogena enhetsmiljöer, vilket begränsar deras skalbarhet och anpassningsförmåga till olika enhetstyper och konfigurationer. Istället behövs ett mer flexibelt förhållningssätt; en som kan rymma en mängd olika anslutna enheter samtidigt som man minimerar beroendet av specifika företagsinstruktioner. En sådan metod är smart felsökning som involverar en 4-stegscykel som inkluderar förebyggande, upptäckt och diagnos, återhämtning och utveckling av hot. Med tanke på dessa premisser är det slutliga målet med denna forskning att definiera en smart felsökningsmetod baserad på loggfiler och produktinformation. Genom att utnyttja en generaliserad metodik försöker detta tillvägagångssätt att förbättra hanteringen av internet of things-system i komplexa miljöer med flera tillverkare. Denna avhandling fokuserar på en systematisk genomgång av loggfiler och den senaste tekniken inom felsökningsmetoder. Under forskningen identifierades bristen på allmänt

tillgängliga loggfiler för felsökningsändamål. Följaktligen föreslogs en metod för att generera syntetiska loggfiler med hjälp av generativa kontradiktoriska nätverk. Den föreslagna metoden utnyttjar dessa loggfiler tillsammans med produktinformation för att förbättra smart felsökning. För att validera tillvägagångssättet och samla in feedback från branschen genomfördes frågeformulär och intervjuer. Efter detta kommer maskininlärningsalgoritmer att användas för att implementera och förfina den föreslagna metoden. Genom att utnyttja en generaliserad metodik försöker detta tillvägagångssätt förbättra hanteringen och feldetekteringen av internet of things-system i komplexa, heterogena miljöer.

Acknowledgment

First and foremost, I would like to express my deepest gratitude to my supervisor, Francesco Flammini, for their invaluable guidance, support, and encouragement throughout the course of my thesis work. Their insights and expertise have been instrumental in shaping the direction and quality of this research.

I am profoundly thankful to my supervisor, Alessio Bucaioni, whose constructive feedback and suggestions greatly contributed to refining this work. This achievement would have been impossible without his help, dedication, and support.

Moreover, I extend my heartfelt gratitude to Johan Thornadtsson, the Sigma Technology Information president, for supporting this research. Despite being involved in other activities, he continued to supervise and support me throughout this journey, ensuring the success of this work. I am always thankful to him and believe that reaching this step would have been challenging without his support.

My deepest appreciation goes to my family for their unwavering support and patience.

Lastly, and most importantly, thank you, Nader, for your love and encouragement throughout this journey.

The work presented in this thesis is funded by the Swedish Knowledge Foundation through INDTECH (20200132) and INDTECH+ (20220132) research schools

Sania Partovian
Eskilstuna, January, 2025

List of Publications

Papers included in this thesis[1]

Paper A: Sania Partovian, Alessio Bucaioni, Francesco Flammini, and Johan Thornadtsson *Analysis of log files to enable smart-troubleshooting in Industry 4.0: a systematic mapping study.* Journal of IEEE Access, 2024.

Paper B: Sania Partovian, Francesco Flammini , and Alessio Bucaioni. *Leveraging GANs to Generate Synthetic Log Files for Smart-Troubleshooting in Industry 4.0.* IEEE International Conference on 50th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA 2024).

Paper C: Sania Partovian, Alessio Bucaioni, Francesco Flammini, and Johan Thornadtsson. *A vision for leveraging product information and log files to enable smart-troubleshooting of heterogeneous interconnected devices.* Submitted (IEEE ICHMS 2025).

[1]The included papers have been reformatted to comply with the thesis layout.

Publications not included in this thesis

Paper D: Sania Partovian, Francesco Flammini , Alessio Bucaioni, and Johan Thornadtsson. *Synthesizing log files using generative and Larg Language Models.* Springer, Lecture Notes in Computer Science.

Contents

I Thesis **1**

1 Introduction **3**

2 Preliminaries **7**
2.1 Smart-troubleshooting 7
2.2 Log files 9
2.3 Product information 10
2.4 Generative Adversarial Networks 11

3 Research plan **13**
3.1 Overall research goal 13
3.2 Research challenges 14
3.3 Research contributions 16
3.4 Paper contributions 18
3.4.1 Paper A 19
3.4.2 Paper B 20
3.4.3 Paper C 21
3.5 Research process and methods 22
3.6 Threats to Validity 24

4 Related work **25**
4.1 Log files and preprocessing 25
4.2 Smart-troubleshooting 27
4.3 Synthetic log files 28

5 Summary and future work **31**

Bibliography **33**

II Included Papers **41**

6 Paper A:
Analysis of log files to enable smart-troubleshooting in Industry 4.0: a systematic mapping study **43**
6.1 Introduction 41
6.2 Preliminaries 43
6.2.1 Industry 4.0 43
6.2.2 Smart-troubleshooting 44
6.2.3 Log files 45
6.3 Research method 47
6.3.1 Research goal and questions 48
6.3.2 Search and selection strategy 49
6.3.3 Classification framework and data extraction 51
6.3.4 Data analysis and synthesis 53
6.3.5 Threats to validity 53
6.4 Results 55
6.4.1 Publication trends (RQ1) 55
6.4.2 Log file attributes (RQ2) 59
6.4.3 Log file analysis techniques (RQ3) 62
6.4.4 Challenges in analyzing log file (RQ4) 67
6.4.5 Data attributes vs. analysis techniques 69
6.4.6 analysis techniques vs. analysis challenges 71
6.5 Related work 71
6.6 Discussion 75
6.7 Conclusions and Future Work 76
Bibliography 77
Bibliography 87

7 Paper B:
Leveraging GANs to Generate Synthetic Log Files for Smart-Troubleshooting in Industry 4.0 **93**
7.1 Introduction . . . 98
7.2 Preliminaries and related work . . . 99
7.2.1 Log Files . . . 100
7.2.2 Generative Adversarial Networks . . . 100
7.2.3 Related work . . . 102
7.3 Methodology . . . 104
7.4 Experimental Setup . . . 106
7.4.1 Dataset . . . 107
7.4.2 Architecture of proposed GANs . . . 107
7.5 Results . . . 109
7.5.1 Visualization Results . . . 109
7.5.2 Empirical Evaluation . . . 110
7.6 Discussion . . . 113
7.7 Conclusions and Future Work . . . 115
7.8 Acknowledgement . . . 116
Bibliography . . . 116

8 Paper C:
A vision for leveraging product information and log files to enable smart-troubleshooting of heterogeneous interconnected devices **121**
8.1 Introduction . . . 121
8.2 Research process . . . 123
8.2.1 Research goal and questions . . . 124
8.2.2 Data extraction . . . 125
8.2.3 Data synthesis and analysis . . . 125
8.2.4 Threats to validity . . . 126
8.3 A novel methodology for smart-troubleshooting . . . 127
8.4 Survey and interview results . . . 132
8.4.1 Survey definition . . . 132
8.4.2 General findings . . . 134
8.4.3 Findings about product information . . . 137

8.4.4 Real-world challenges for smart-troubleshooting . . . 140
8.4.5 Expert Interviews . . . 142
8.5 Related work . . . 143
8.6 Discussion . . . 144
8.6.1 Smart-troubleshooting system . . . 144
8.6.2 Documentation and Continuous Improvement . . . 145
8.6.3 Challenges and limitations . . . 145
8.6.4 Future of smart-troubleshooting . . . 145
8.7 Conclusions and Future Work . . . 146
Bibliography . . . 147

I

Thesis

Chapter 1

Introduction

Industry 4.0 represents the fourth industrial revolution, characterized by the integration of digital technologies into manufacturing processes [1]. Industry 4.0 comprises Cyber-Physical Systems (CPS) that merge physical operations with computational capabilities for real-time oversight and management. It also includes the Industrial Internet of Things (IIoT), which consists of a network of connected devices and sensors that improve communication and operational effectiveness. Furthermore, it highlights the idea of smart factories, which are defined by advanced automation, connectivity, and the capacity for self-optimization, self-configuration, and self-adaptation [2].

As Internet of Things (IoT) deployments grow quickly in both number and complexity, the demand for strong security solutions becomes essential. This requires scalable and standards-compliant mechanisms to ensure the long-term support, maintainability, and secure management of these connected devices [3]. Effective troubleshooting is crucial for maintaining system resilience and performance in the quickly advancing IoT environment [4]. Smart-troubleshooting is the (semi)automated process for pinpointing system failures, associating them with relevant troubleshooting information, and implementing suitable recovery and reconfiguration steps [5]. This approach includes elements of resilience and self-healing, viewed from a comprehensive system-of-systems standpoint [6]. It integrates multiple fields such as artificial intelligence, formal modeling, and data analytics to facilitate the real-time detection of failure patterns, extraction of

troubleshooting data from both natural language and machine-readable sources, and the deployment of solutions using organized workflow management models and procedures [5].

Log files are integral to smart-troubleshooting since they are key in monitoring the status of systems, applications, or devices and capturing significant events. This information helps administrators accurately identify and diagnose anomalies such as system bugs, failures, and errors [7]. Analyzing this data by hand is both time-consuming and demands expert skills. Additionally, the vast amount of data makes manual analysis nearly impossible [8]. For a single device, log files are defined and organized. However, when dealing with diverse interconnected devices, the log files lack structures [9].

Despite extensive research and numerous methods aimed at enabling IoT devices to detect anomalies through self-supervised training, these approaches frequently struggle to elucidate the significance and underlying causes of the anomalies they identify. This shortcoming is particularly evident in rare situations or scenarios involving multiple simultaneous errors where the causes are highly intricate. Moreover, current techniques typically lack the capacity to adapt to the diverse and heterogeneous nature of IoT environments, making them unsuitable for complex, real-world applications.

For this purpose, a thorough review of log file analysis in Industry 4.0 and manufacturing is essential to understand how log files are utilized for failure detection, anomaly detection, prediction, and troubleshooting. So, first, we will begin with a systematic mapping study of log files for troubleshooting purposes. This study will help us categorize and evaluate the existing research and practices related to the use of log files in troubleshooting within the context of Industry 4.0 and manufacturing.

The scarcity of publicly accessible and usable log file datasets makes it challenging to validate new anomaly detection methods based on log file analysis, rendering them less reliable. This issue is particularly significant due to the lack of large-scale benchmark datasets in software engineering research, a problem that has been emphasized in numerous studies [10]. To address challenges in obtaining real data, synthetic data generation using GANs (Generative Adversarial Networks) [11] such as VanillaGAN, CTGAN(Conditional Tabular Generative Adversarial Network), and SeqGAN (Sequential Generative Adversarial Net-

work) can be employed to create artificial log datasets. This approach facilitates research and experimentation in log analysis despite data scarcity or privacy constraints.

Every system has a piece of very detailed product information that holds expert knowledge. It can be mentioned that expert knowledge can be updated through online access and also forums on related sites. By utilizing this knowledge, we can create and implement collections of efficient ML models, which can find the anomaly and give the root cause of the anomaly and solution extracted from the product information [12].

[colback=yellow!10!white,colframe=yellow!50!black] Therefore, the ultimate goal of this thesis is to propose a method that leverages product information and log files, using machine learning techniques, to enable smart-troubleshooting in interconnected devices. In other words, integrating machine learning components into the traditional troubleshooting has allowed it to become autonomous. Thus, after proposing the methodology, collaborating with industry experts such as site reliability engineers is crucial for demonstrating its practical application in the industry. This collaboration will provide valuable insights and help address any validity concerns regarding our methodology.

In what follows are Chapter 2 introducing the main concepts and their associated terminologies used throughout the thesis, Chapter 3 describing in detail the research goals, challenges, contributions and the adopted research methodology, Chapter 4 comparing our work with existing related literature published over the years, and finally, Chapter 5 presenting a summary the thesis along with a brief overview of the future work.

Chapter 2

Preliminaries

This section briefly introduces the main concepts and terminologies used in this thesis proposal.

2.1 Smart-troubleshooting

Smart-troubleshooting is a problem-solving approach that leverages data, technology, and analytical techniques to efficiently identify and address issues [5]. Smart-troubleshooting comprises four phases as outlined below [13]:

1. The prevention phase aims to foresee and address potential failures before they arise. This stage is crucial in smart-troubleshooting. In this study, we use log file analysis as a forecasting tool to detect faults, involving the collection and evaluation of data to identify potential issues accurately.

2. The detection and diagnosis phase focuses on identifying and diagnosing failures as they happen. This stage is dedicated to spotting errors within a system and supplying necessary information to the recovery and adaptation phases. These inputs help in addressing threats, resolving errors or failures, reconfiguring the system, and enhancing its overall resilience.

3. The recovery phase involves addressing and resolving failures. Various methods and techniques can be employed for effective error handling and

failure recovery, based on the information gathered during the detection phase. One approach is to integrate product information with log file analysis, a method that is elaborated upon in our proposed solution.

4. The adaptation phase focuses on making adjustments to the system to prevent future failures. Its objective is to enhance the evolutionary and adaptive capabilities of interconnected devices to better address emerging threats. This phase involves continuous real-time monitoring of the system and its environment, allowing for timely modifications to maintain optimal system performance.

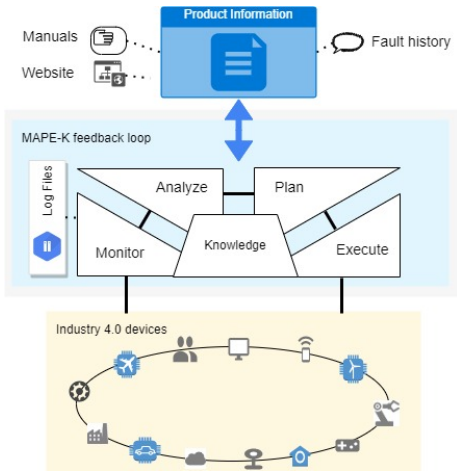

Figure 2.1. Smart-troubleshooting in Industry 4.0 [14]

Figure 2.1 illustrates smart-troubleshooting within the framework of Industry 4.0. In our examination of self-adaptive systems, the management of these systems typically follows the Monitor-Analyze-Plan-Execute with a shared Knowledge (MAPE-K) framework [14]. While Industry 4.0 focuses on the integration of cyber-physical systems, IoT, and automation to enhance industrial efficiency, Industry 5.0 marks a shift towards collaboration between humans and machines, emphasizing personalized production. In the Industry 5.0 paradigm,

human workers are placed at the center of the production process, working alongside advanced technologies such as AI, robotics, and smart systems to create tailored products [15]. The transition from Industry 4.0 to Industry 5.0 highlights the importance of human creativity, ethics, and sustainability in the future of manufacturing, moving beyond efficiency-driven goals to those that are more user-centric and environmentally consciouscts both the technological advancements and the human-centered shift outlined in Industry 5.0.[16]

2.2 Log files

As software systems become more complex, monitoring their behavior is crucial to ensure they function correctly. One effective approach is to use system logs, which document significant events and system states at various times [17]. The National Institute of Standards and Technology (NIST) defines a log as a recorded account of events occurring within an organization's systems and networks [18]. These logs are valuable for monitoring, auditing, and debugging, providing detailed information about the system's operational status, including parameters, events, and process execution details. They contain a wide range of information about the running status of a system (such as parameters, events, process execution details, etc.) [19].

Here are some common types of log files:

System Logs. These logs capture events related to the operating system, such as startup and shutdown sequences, hardware malfunctions, and resource usage [20].

Application Logs. Generated by software applications, these logs document user interactions, including errors, warnings, and other significant events within the application [21].

Access Logs. These logs record every request made by users or systems, detailing access to individual files and system interactions [22].

Security Logs. Security logs track system status and notable security events,

aiding security professionals in detecting intrusions and vulnerabilities, and providing insights into the system's overall security posture [23].

Server Logs. These logs automatically document the activities of a server, such as web servers maintaining a history of page requests [24].

2.3 Product information

While both log files and product information are valuable tools for troubleshooting, they serve different purposes. Log files are dynamic records generated during the operation of a product or system, capturing real-time events, errors, and performance data. These files are critical for diagnosing issues as they provide insight into what happens during system failures or irregularities. However, they do not contain the structured, comprehensive data needed to understand the core design, functionality, and specifications of a product.

This is where product information comes in. The information provided by the product which includes documentation, user manuals, and technical specifications, plays a crucial role in troubleshooting. According to Abraham, product information management is an integral part of a successful e-commerce strategy, as it provides customers with accurate and detailed information about products [12]. It is important to have product information not only for customers, but also for technicians who are responsible for troubleshooting product issues. Khandpour emphasizes that having access to complete and accurate product information can help technicians understand the product's design, function, and specifications, which in turn can help them identify potential issues and their possible solutions [25]. Log files point to the issues, while product information contains the answers. According to Mazur, product information can assist in diagnosing problems and determining whether a part needs to be replaced or repaired [26]. In the absence of accurate product information, troubleshooting can be time-consuming and frustrating. For this reason, companies should prioritize accurate and detailed product information in order to ensure successful troubleshooting. We consider product information from different companies, but it is written by experts and technical writers who follow a structured approach. For example, they adhere to the principles of Information Mapping for Learning

and Reference [27]. This method organizes categories of information and displays them effectively for both learning and reference purposes, ensuring clarity and consistency.

2.4 Generative Adversarial Networks

Synthetic data generation creates artificial datasets to overcome challenges like privacy concerns or data scarcity, such as the lack of public log files. Traditional methods often struggle to replicate the original data's key statistical properties. Machine Learning (ML) and Deep Learning (DL) provide innovative solutions by automatically learning patterns from existing data, capturing distributions and correlations with statistical fidelity [28]. These models are flexible, scalable, and aid in privacy preservation, especially with customer data. However, challenges like overfitting and data diversity must be considered. A promising approach is using GANs, where a generative model (G) mimics data distribution and a discriminative model (D) distinguishes between real and generated samples. G improves by maximizing the likelihood that D misclassifies the samples, creating a two-player minimax game until reaching Nash equilibrium [11]. Figure 2.2

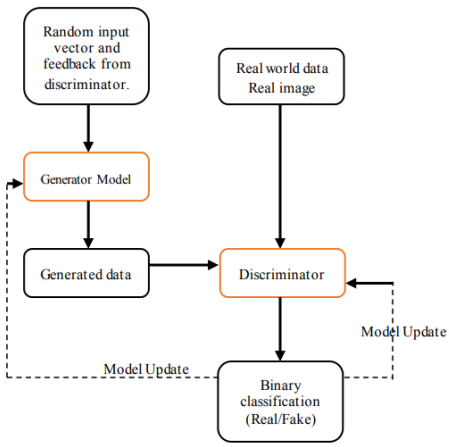

Figure 2.2. General approach of GAN [29].

shows a flowchart of a GAN, depicting the general schematic of the approach. The loss functions for the generator and discriminator in a GAN are defined as follows:

- For the generator, the loss function L_G is given by:

$$L_G = -E_{z \sim p(z)}[\log D(G(z))]$$

  where:

  - z is a random noise vector sampled from a prior distribution $p(z)$.
  - $G(z)$ represents the data generated by the generator.
  - $D(G(z))$ denotes the discriminator's output for the generated data.

- For the discriminator, the loss function L_D is:

$$L_D = -E_{x \sim p_{data}(x)}[\log D(x)] - E_{z \sim p(z)}[\log(1 - D(G(z)))]$$

  where:

  - x represents real data samples.
  - $D(x)$ is the discriminator's output for real data.

In this formulation, the discriminator aims to maximize L_D, while the generator aims to minimize L_G.

Chapter 3

Research plan

This section outlines the research plan, detailing the research goal, challenges, contributions, methodologies, and the overall process.

3.1 Overall research goal

Smart-troubleshooting has introduced as a crucial capability in managing the complexities of modern interconnected devices, especially within the rapidly expanding IoT devices [5]. As devices become increasingly heterogeneous, with varying protocols, data formats, and operational contexts, the challenge of maintaining system reliability and quickly diagnosing and resolving issues has grown significantly [30]. A key component in addressing these challenges is the effective utilization of log files and product information, which can offer invaluable insights into system behavior, anomalies, and potential faults. To enhance the effectiveness of troubleshooting in such environments, there is a need for a robust, generalized methodology that leverages log file analysis and integrates product-specific information. This methodology would not only facilitate more accurate fault detection and diagnosis but also support the development of self-healing capabilities in IoT systems, ultimately leading to more resilient and autonomous operations. A critical challenge in log file analysis is the limited availability of publicly accessible log files that include faults, errors, and failures. Due to privacy concerns, these log files are often not released [10]. Furthermore,

the integration of machine learning into smart-troubleshooting not only enhances its capabilities but also introduces a more advanced layer of analysis and decision-making. Given the above considerations, our research is focused on establishing a practical and comprehensive methodology for smart-troubleshooting in heterogeneous IoT systems, with a specific emphasis on the utilization of log files and product information. Therefore, our primary research goal is as follows:

Research Goal (RG): *Develop a comprehensive methodology for smart- troubleshooting in Industry 4.0 that utilizes log files and product information to accurately detect, diagnose, and resolve issues in interconnected systems.*

The goal is to propose a smart-troubleshooting framework that leverages data from log files and product information to improve threats detection and resolution in Industry 4.0 environments. The scope of the framework will cover the entire troubleshooting process, from identifying anomalies and patterns within log files to diagnosing root causes and suggesting corrective actions. This methodology will include the creation of a practical test bed to simulate real-world Industry 4.0 scenarios, where log files can be generated, analyzed, and tested for anomaly detection and issue resolution. Additionally, the methodology will emphasize adaptability to different types of IoT devices and manufacturing systems, and industry collaboration will be integral to validate the framework's applicability, ensuring its relevance and effectiveness in diverse industrial settings.

3.2 Research challenges

We divided the above goal into three distinct research challenges (RCs), each addressing unique and complementary aspects of the goal.

Research Challenge 1 (RC_1): *Collect and structure the latest techniques for analyzing log files in Industry 4.0.* The rapid evolution of Industry 4.0 technologies has led to the development of numerous log file analysis techniques aimed at enhancing system reliability and troubleshooting efficiency. However,

these techniques vary widely in their approaches, effectiveness, and application contexts. With the diversity of methods emerging from different industrial sectors, there is a lack of a unified framework to systematically categorize and evaluate these techniques. This challenge is further compounded by the heterogeneous nature of Industry 4.0 environments, which involve varying protocols, data formats, and operational conditions. Therefore, the challenge here is to systematically identify the studies showing the log file analysis techniques for advancing smart-troubleshooting practices, ensuring that the most appropriate and effective methods are employed across different scenarios within Industry 4.0.

Research Challenge 2 (RC_2): *Address the shortage publicly available log files for Industry 4.0.* A significant obstacle in advancing smart-troubleshooting methodologies is the limited availability of publicly accessible log files that contain realistic data, including faults, errors, and failures. This scarcity arises primarily due to privacy concerns and the proprietary nature of much industrial data. Without access to diverse and comprehensive datasets, researchers face difficulties in developing, validating, and benchmarking log analysis techniques. This shortage hinders progress in creating robust, generalizable solutions that can be applied across various Industry 4.0 environments. Tackling this challenge requires innovative approaches to data generation, such as synthetic log file generation.

Research Challenge 3 (RC_3): *Introduce a methodology for smart-troubleshooting that leverages both product information and log files.* The primary aim of this challenge is to create a unified and adaptable approach that can be seamlessly integrated into the diverse and complex environments typical of Industry 4.0. Current methodologies often lack the flexibility needed to handle the variability in system architectures, data sources, and operational contexts found across different industrial sectors. Additionally, these methodologies must address the challenge of correlating product information with log data to provide actionable insights for troubleshooting. The methodology should encompass the three phases of detect, diagnose, and resolve issues. This phased approach ensures a comprehensive troubleshooting process that can effectively identify

faults, determine their root causes, and implement solutions. Lastly, the heterogeneous nature of different devices make it challenging to establish a unique solution for fault detection, diagnosis, and resolution processes. The ultimate goal is to introduce a methodology that not only works theoretically but is also validated through rigorous industry feedback, ensuring its effectiveness in real-world applications. This includes ensuring that the methodology is scalable and robust enough to handle the dynamic and heterogeneous nature of modern manufacturing systems.

3.3 Research contributions

The following section outlines the three primary research contributions (RCOs) that address the aforementioned challenges and goals. Table 3.1 illustrates the alignment between these research contributions and the identified challenges.

	RC$_1$	**RC**$_2$	**RC**$_3$
RCO$_1$	X		
RCO$_2$		X	X
RCO$_3$			X

Table 3.1. Mapping of research contributions to challenges.

Research Contribution 1 (RCO_1**):** *Systematic identification, classification, and analysis of log file analysis techniques for smart-troubleshooting in Industry 4.0.* This contribution presents a structured and detailed review of existing research focused on log file analysis techniques for smart-troubleshooting in Industry 4.0. Specifically, it includes: i) an overview of publication trends, including the number of publications, key venues, and research focus over time; ii) a categorization of research types and validation methods according to established taxonomies; iii) an identification of key attributes and challenges related to log file analysis; iv) a comparative analysis of the most commonly used techniques, such as machine learning approaches, for log file analysis; and v) an exploration of the challenges associated with implementing these techniques in real-world industrial environments. This contribution is crucial as

it provides a comprehensive framework for understanding the current state of log file analysis in Industry 4.0, highlights gaps in existing research, and offers insights into the strengths and limitations of various approaches. It serves as a foundation for further research and development of more effective and efficient smart-troubleshooting methodologies. RCO_1 corresponds to Paper A [6] and tackles RC_1.

Research Contribution 2 (RCO_2): *Addressing Data Availability Challenges in Log File Analysis for Industry 4.0.* This contribution addresses the significant challenge of limited publicly available log data for developing and testing smart-troubleshooting techniques in Industry 4.0 environments. Leveraging GANs, this work proposes a methodology for generating synthetic log files that accurately mimic real-world industrial log data. The process involves utilizing different GAN models, such as VanillaGAN, CTGAN, and SeqGAN, to produce synthetic logs that maintain the statistical properties and patterns of authentic log files. A comprehensive comparative analysis of these GAN models is conducted to determine the most effective approach for generating high-quality synthetic logs. The generated logs are designed to include various faults, errors, and failure patterns, making them highly valuable for evaluating and refining smart-troubleshooting techniques. These synthetic logs are crucial not only for addressing data scarcity but also for providing a test-bed to validate and refine the proposed smart-troubleshooting methodology. By enabling extensive and realistic testing of anomaly detection and troubleshooting techniques, this contribution supports the practical implementation of smart-troubleshooting solutions. Therefore, RCO_2 directly addresses RC_2 by overcoming data availability limitations and also contributes to RC_3 by facilitating the evaluation and improvement of the proposed methodology through a robust test-bed. RCO_2 corresponds to Paper B [31] and tackles RC_2 and RC_3.

Research Contribution 3 (RCO_3): *Introducing a methodology for smart-troubleshooting leveraging product information and log files.* This contribution focuses on proposing and validating an intelligent and holistic troubleshooting approach by integrating data from log files and product information across a

wide array of interconnected devices. The concept of smart-troubleshooting, originating from industry practices, involves the systematic collection and analysis of failure data, aligning these insights with troubleshooting instructions and software solutions to ensure more effective issue resolution. The methodology involves identifying potential faults by analyzing log data for deviations from expected behavior. Once issues are detected, the next step is to diagnose the problem by correlating log data with product specifications and operational parameters to determine the root causes. Finally, actionable solutions are implemented based on the diagnosis to address the faults and restore the system to optimal performance. To realize this vision, we designed a smart-troubleshooting methodology that enhances traditional troubleshooting processes by utilizing the rich data contained in log files and detailed product information. We conducted a mixed-methods empirical study to assess the practicality and effectiveness of our proposed methodology. This study involved planning, executing, and analyzing an online survey, complemented by qualitative insights from in-depth, semi-structured interviews with industry professionals, including site reliability engineers. By engaging with industry experts, we addressed potential gaps between academic research and practical implementation, ensuring that our methodology is both theoretically sound and practically viable. RCO_3 directly supports RC_3 by offering a validated, industry-informed smart-troubleshooting methodology and corresponds to Paper C.

3.4 Paper contributions

In this section, we present the papers included in the licentiate thesis. Table 3.2 illustrates the mapping between the papers and the RCOs discussed in Section 3.3.

Papers	Venue	Article Type	Research Contribution	Research Challenge	Status
$\mathbf{P}_A$	IEEE Access	Journal Article	RCO_1	RC_1	Published
$\mathbf{P}_B$	IEEE SEAA 2024	Conference Paper	RCO_2, RCO_3	RC_2	Published
$\mathbf{P}_C$	IEEE ICDCS 2025	Conference Paper	RCO_1, RCO_3	RC_1	Submitted

Table 3.2. Papers mapped onto contributions and challenges

3.4.1 Paper A

Title: Analysis of Log Files to Enable Smart-Troubleshooting in Industry 4.0: A Systematic Mapping Study
Authors: Sania Partovian, Alessio Bucaioni, Francesco Flammini, Johan Thornadtsson
Abstract: A crucial element of Industry 4.0, is the utilization of smart devices that generate log files. Log files are key components containing data on system operations, faults (unexpected glitches or malfunctions), errors (mistakes or incorrect actions), and failures (complete breakdowns or non-functionality). This paper presents a systematic mapping study analyzing research conducted on log files for smart-troubleshooting in Industry 4.0. To the best of our knowledge, this is the study that aims to identify research trends, log file attributes, techniques, and challenges involved in log file analysis for smart-troubleshooting. From an initial set of 941 potentially relevant peer-reviewed publications, 74 primary studies were selected and analyzed using a meticulous data extraction, analysis, and synthesis process. The results of the study demonstrate that the majority of research has focused on developing algorithms for log analysis, with machine learning being the most commonly used approach. The smart-troubleshooting encompasses a range of activities and tools that are essential for collecting failure data generated by diverse interconnected devices, conducting analyses, and aligning them with troubleshooting instructions and software remedies. Moreover, the study identifies the need for further research in the areas of real-time log analysis, anomaly detection, and the integration of log analysis with other Industry 4.0 technologies. In conclusion, our study provides insights into the current state of research in log analysis for smart-troubleshooting in Industry 4.0 and identifies areas for future research. The use of smart devices generating log files in Industry 4.0 highlights the importance of log file analysis for troubleshooting purposes. Further research is needed to address the challenges and opportunities in this field to integrate log analysis with other Industry 4.0 technologies for performing more efficient and effective troubleshooting.
Personal contribution: I led the development of this research as the primary author, responsible for crafting the research questions, designing the search strategy, and conducting data extraction and synthesis on log file analysis techniques for smart-troubleshooting in Industry 4.0. This involved analyzing and catego-

rizing studies related to the use of log files and machine learning techniques over the past decade. My co-authors contributed by reviewing and providing feedback on drafts, ensuring the accuracy and clarity of the paper before final submission.
Status: Published in IEEE Access, vol. 12, pp. 147640-147658, 2024, doi: 10.1109/ACCESS.2023.3342365.

3.4.2 Paper B

Title: Leveraging GANs to Generate Synthetic Log Files for Smart-Troubleshooting in Industry 4.0
Authors: Sania Partovian, Francesco Flammini, and Alessio Bucaioni
Abstract: In this paper, we tackle the challenge of generating synthetic log files using generative adversarial networks to support smart-troubleshooting experimentation. Log files are critical for implementing monitoring systems for smart-troubleshooting, as they capture valuable information about the activities and events occurring within the monitored system. Analyzing these logs is crucial for effective smart-troubleshooting, enhancing the overall efficiency, reliability, and security of smart manufacturing processes. However, accessing public log data is difficult due to privacy concerns and the need to protect sensitive information. Moreover, for the purpose of effective troubleshooting, it is essential to have datasets that include fault, error, and failure logs as well as standard logs. In recent years, synthetic log files have emerged as a promising solution to augment limited real-world datasets and facilitate the development and evaluation of anomaly detection techniques. Building on this concept of synthetic data, we have developed a specific log generation technique and dataset tailored for testing smart-troubleshooting techniques in heterogeneous connected systems environments, such as industrial cyber-physical systems and the internet of things. First, we propose a methodology that generates synthetic log files based on generative adversarial networks. Later, we instantiate this methodology using different Generative Adversarial Network implementations and present a validation and a comprehensive comparative analysis of their performance. Eventually, we provide a robust dataset for anomaly detection and threat analysis in cyberspace security. Based on the results of our comparison, CTGAN has

shown superior performance in generating high-quality synthetic log files.
Personal contribution: As the primary driver and main author of this research, I designed the research question and led the development of a methodology utilizing Generative Adversarial Networks (GANs) to generate synthetic log files tailored for smart-troubleshooting in Industry 4.0. My responsibilities included GAN model selection, training, and evaluating synthetic log quality. All the co-authors help improve the quality of the paper by constantly reviewing and commenting on the presented drafts.
Status: Published in 50th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA) 2024.

3.4.3 Paper C

Title: A vision for leveraging product information and log files to enable smart-troubleshooting of heterogeneous interconnected devices
Authors: Sania Partovian, Alessio Bucaioni, Francesco Flammini, Johan Thornadtsson
Abstract: The concept of smart-troubleshooting originates from the necessity to manage the increasing complexity of interconnected devices in environments like smart cities, Industry 4.0, and cyber-physical systems. Smart-troubleshooting refers to the semi-automatic process that identifies system failures, matches them with relevant troubleshooting information, and applies recovery or reconfiguration actions. Smart-troubleshooting is not yet fully realized, and although there are some works addressing parts of the problem, a comprehensive solution has not been achieved. To bridge this gap, we propose a vision for smart-troubleshooting that integrates product information and log files. By utilizing both product information and log files, this approach enables the collection, analysis, and synthesis of data from diverse sources, empowering the system to self-heal and enhance its resilience. In this paper, first we propose the methodology that implements the vision of smart-troubleshooting by integrating product information and log files to automate failure detection and resolution in interconnected systems. The methodology introduces key innovations by automating the correlation of failures through the integration of product data and log files, detecting real-time patterns using data mining techniques, and

incorporating a feedback loop that allows the system to continuously improve its response to future issues. We validated the methodology using a mix-method approach consisting surveys and interviews. In particular, we distributed the survey to 40 practitioners in the maintenance domain. We extracted, analyzed, and synthesized the survey data, complementing it with qualitative insights from four online, semi-structured, in-depth interviews. To address the limitations of our vision and gather feedback from the industry, we engaged in extensive discussions with recognized professionals, such as site reliability engineers. This collaborative approach not only helped mitigate potential threats to the validity of our vision but also facilitated a valuable exchange of insights and perspectives from experts in the field and identified the gap between industry and academia.

Personal contribution: I was the main driver of this research and the primary contributor to the writing of the paper. My responsibilities included designing surveys, gathering feedback, conducting interviews, and extracting and analyzing results. Moreover, based on this, I proposed the "Smart-troubleshooting" methodology. The co-authors helped improve the quality of the paper by reviewing and providing the necessary comments.

Status: Submitted to 45th IEEE International Conference on Distributed Computing Systems, IEEE ICDCS 2025

3.5 Research process and methods

In this section, we detail the research process and methods used in our study. The term process refers to the overarching workflow that structured our approach to achieving our research objectives. Methods refers to the specific research techniques applied in the development of our contributions. Our research process is influenced by the Action Design Research (ADR) methodology, a widely recognized framework that provides structured guidance for creating IT artifacts [32]. Figure 3.1 presents the key steps of the process we followed, inspired by ADR.

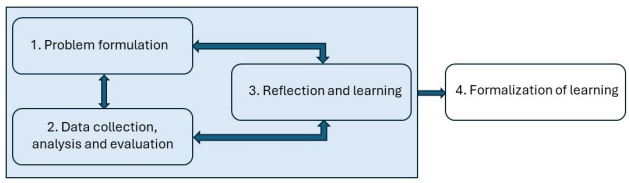

Figure 3.1. Research process adapted from [32]

Step # 1 Problem Formulation: Problem Formulation: To define our overall RG, we began by conducting a thorough review of existing scientific literature to identify gaps in the field of smart-troubleshooting in interconnected devices. Our focus was on leveraging log files and product information to enhance the troubleshooting process in Industry 4.0 environments. Through an iterative process, we developed three distinct research challenges, each addressing a specific aspect of this overarching goal. For instance, after analyzing the initial findings from our literature review, we formulated RC1 to classify and organize the latest techniques in log file analysis. As our research progressed, the results from RCO1 led us to define RC2, which focused on addressing the scarcity of publicly available log data. Finally, the insights gained from RCO2 shaped the development of RC3, aimed at creating a comprehensive methodology for practical smart-troubleshooting.

Step # 2 Data collection, analysis, and evaluation: To address each of the defined research challenges, we collected relevant data using systematic approaches, guided by the methodologies of Kitchenham et al. [33] and Petersen et al. [34]. For instance, to tackle RC1, we conducted a systematic mapping study to classify log analysis techniques. The data collected was then analyzed using thematic synthesis and grounded theory techniques, as detailed in our peer-reviewed publications. For RC2, we applied descriptive statistics to evaluate the effectiveness of GAN-based methods in generating synthetic log files, which were essential for overcoming the challenge of data scarcity. The findings were represented graphically to provide a clear and comprehensive overview of our results.

Step # 3 Reflection and Learning: After analyzing the results from Step 2, we reflected on our findings to refine our contributions and address each research challenge more effectively. For example, while evaluating the outcomes of RC1, we recognized the need to explore more advanced machine learning techniques for log file analysis, which led to the development of RCO3. Similarly, the analysis of synthetic log data in RC2 highlighted the importance of integrating industry feedback, prompting us to refine our methodology in RCO3 to ensure its practical applicability.

Step # 4 Formalization of Learning: The insights and findings obtained from Step 3 were formalized and documented in our peer-reviewed publications. For example, the refined methodology for smart-troubleshooting developed in RCO3 was detailed in a publication that outlined the practical steps for implementing this approach in industrial settings.

3.6 Threats to Validity

The chosen research methodology may be influenced by personal bias, given the researcher's extensive involvement and the qualitative nature of the results. To mitigate this validity threat, the entire research process was carefully documented to ensure transparency. Additionally, external reviews were sought through the peer-review process of the submitted articles presenting the research findings.

Chapter 4

Related work

In this section we summarise some of the research works that are related to our goal and contributions.

4.1 Log files and preprocessing

Here, we highlight several related studies that explore log file analysis solutions. Unlike our research, which applies log file analysis specifically for smart-troubleshooting across heterogeneous devices, many of these works primarily focus on log analysis within a single domain and are not tailored for troubleshooting in diverse, interconnected systems.

A 2021 paper offers an in-depth examination of the challenges and advancements in log-based software monitoring. The study emphasizes the crucial role of application logs and runtime environment data in identifying and diagnosing unexpected behaviors, thereby enhancing system reliability. It also outlines the difficulties developers face in making logging decisions, managing and storing log data, and extracting valuable insights. Additionally, the authors provide a structured overview of the current state-of-the-art in technology transfer, framing the research within the context of the log data lifecycle [35].

Behera et al. explore various techniques for detecting anomalies within log analysis. They cover a range of methods, including console log analysis, program invariants, data mining techniques, recurrent neural networks, deep

neural networks, and deep learning frameworks. The paper offers a comparative analysis, examining the strengths and weaknesses of each approach, and discusses the algorithms and models utilized for anomaly detection. The survey emphasizes the significance of analyzing unstructured log data, reviews different detection methods, and highlights the challenges and possible solutions in this area [36].

Abdalla et al. explore the widespread applications of machine learning algorithms in log file analysis, categorizing these applications into several distinct areas. Their survey addresses multiple facets of log analysis, such as prediction, reporting, profiling, anomaly detection, and user behavior analysis. Additionally, the authors highlight the key challenges and limitations in the field, emphasizing the need for continued advancements [37].

Chen et al. provide an in-depth survey that addresses the challenges in log instrumentation and suggests potential solutions. Through a systematic review, they identify key difficulties associated with the process of log instrumentation, which involves embedding logging statements into code to produce log messages. It's important to differentiate between log instrumentation and log management; while the former pertains to generating logs, log management focuses on collecting, analyzing, and interpreting these logs to understand and enhance system performance. Unlike Chen et al.'s focus on instrumentation, our work is centered on log management, emphasizing the analysis and interpretation of log data to diagnose issues and optimize system performance. [38].

Log Parsing and Preprocessing techniques are essential for the efficient and effective analysis of system logs [39]. So, it is a well-known technique to prone the data. Many methods presented by researchers for example a formalised method for pre-processing and transforming ICS (Industrial Control System) device logs into event logs [40], or designed and implemented a parallel log parser (namely POP) on top of Spark, a large-scale data processing platform [41] or used more simple methods and extracted the log event templates based on the log print statements in the source code [42]. Some research used simple regular expressions to parse some commonly-used variables (e.g., IP address, timestamp and number) and used this method to convert logs to a structured and usable format [43]. Online preprocessing method is used frequently to develop other techniques for example Drain is an online log parsing approach with

Fixed Depth Tree [44] which is used frequently to expand anomaly detection methods [45] [46]. Jadidi et al. did preprocessing phase using Wireshark[1] and tshark[2] software to convert the Pcap (network traffic packet captures) files to CSV (Comma-Separated Values) files that required for the anomaly detection phase [47].

4.2 Smart-troubleshooting

This section reviews related works on smart-troubleshooting. Unlike previous studies, our research takes a step further by proposing a practical methodology and testbed for smart-troubleshooting in heterogeneous interconnected devices.

Caporuscio et al. contribute to the field of smart-troubleshooting in several significant ways through their paper. They introduce a conceptual framework that aims to enhance the resilience of systems by enabling self-healing capabilities. In addition, they outline four main phases of smart-troubleshooting: Prevention, Detection, Recovery, and Adaptation. This structured approach not only clarifies the lifecycle of troubleshooting processes but also provides a roadmap for future research and development in this area. Overall, the paper serves as a foundational resource for researchers and practitioners interested in advancing the field of smart-troubleshooting, offering both a theoretical framework and practical insights into the challenges and opportunities that lie ahead [5].

Gu et al. conducted a systematic mapping study to investigate logging practices in software engineering, identifying gaps between existing challenges and proposed solutions. By synthesizing various issues and approaches, their research provides insights into the current state of logging practices. Utilizing the Goal-Question-Metric (GQM) framework, the authors reveal omissions and suggest future research directions, emphasizing the need for improved logging methodologies in the development and maintenance of business software systems [48]. Singh et al. make significant contributions to the field of smart-troubleshooting and self-healing in the IoT by utilizing log analytics and process mining techniques. Their research addresses the challenges of anomaly detection, automatic problem detection, and automated repair within complex and

[1] https://www.wireshark.org/
[2] https://tshark.dev

heterogeneous IoT environments. In their 2020 work, they propose methods for generating consistent event logs across various IoT devices, facilitating process mining to autonomously identify and address system failures [49]. Building on this foundation, their 2022 research combines machine learning with process mining to further enhance the resilience of interconnected cyber-physical systems, offering a novel approach to diagnosing and resolving issues in IoT environments. Together, these studies provide a comprehensive roadmap for implementing effective self-healing capabilities in IoT systems, marking significant progress in this domain [4].

4.3 Synthetic log files

State-of-the-art methods for synthetic log file generation use either systematic approaches or generative models. Systematic methods create synthetic logs by simulating program executions, replicating both normal operations and anomalies to test anomaly detection algorithms. For example, one method generates log datasets that mimic linear program execution, including normal and erroneous behaviors by simulating multiple program runs [50]. However, these synthetic datasets may not fully capture the complexity of real-world data, potentially limiting the applicability of the research.

Generative models offer an innovative solution for generating synthetic log files, addressing the challenge of limited labeled log data for training anomaly detection models. For example, one study used a generative model to create adversarial log files with subtle changes aimed at evading anomaly detection systems. This approach highlights the vulnerabilities in current log parsing tools, demonstrating how logs can be manipulated to bypass detection and compromise system security and reliability [51]. The focus is more on log parsing and structure than on the dataset itself.

Generative models have been increasingly employed to address the challenge of insufficient log data for various applications. For example, Kim et al., used a Deep Neural Network (DNN) model to generate synthetic density log data for wells in the Golden field, Alberta, Canada, to overcome the data shortage in reservoir modeling [52]. Wurzenberger et al. utilized a generative model combining log line clustering and Markov chain simulation to create synthetic

log files, generating realistic Network Event Sequence (NES) data for testing security and network analysis tools offline [53]. However, while Markov chains are effective for simpler systems, they fall short in handling complex data compared to more advanced models like GANs. Chen et al. leveraged GANs to generate synthetic samples for network threat detection, addressing dataset imbalances by augmenting the limited number of malicious logs and network traffic data [54]. These approaches highlight the growing use of generative models to create synthetic log data for diverse applications, from reservoir modeling to cybersecurity.

GANs are not only useful for generating log files but also play a significant role in log anomaly detection. For example, LogGAN improves anomaly detection by utilizing GANs combined with permutation event modeling [55]. Similarly, Duan et al. introduced a method that uses an Encoder-Decoder framework with LSTM networks for the generator and CNNs for the discriminator to detect log anomalies [56].

Chapter 5

Summary and future work

This thesis explored smart-troubleshooting methodologies for heterogeneous interconnected devices, focusing on the integration of product information and log file analysis. The study examined the alignment between industrial practices and academic research, highlighting the potential for automation and data-driven approaches to improve troubleshooting efficiency. The findings suggest that leveraging structured data and automated processes can significantly enhance fault detection, diagnosis, and resolution, reducing downtime and improving system reliability.

The results emphasize that industrial practices often rely on manual troubleshooting methods, which can be time-consuming and error-prone. In contrast, academic research points toward predictive analytics and AI-driven techniques as more scalable and proactive solutions. This comparison highlights the need for industries to adopt smarter troubleshooting strategies that incorporate automation and AI technologies to bridge the existing gaps.

This research also identified several opportunities for advancing troubleshooting methodologies. The integration of product information and log files enables more comprehensive diagnostics, supporting predictive maintenance and reducing operational disruptions. Automated systems driven by AI can further improve scalability, allowing industries to transition from reactive approaches to predictive and adaptive troubleshooting frameworks.

While the proposed methodology demonstrated practical applicability through

empirical studies, certain challenges remain. These include addressing the complexity of troubleshooting heterogeneous systems, handling unforeseen errors, and ensuring compatibility across diverse devices and platforms. The findings suggest that refining AI models and integrating data sources can help overcome these limitations, paving the way for more robust and scalable solutions.

Future research will focus on enhancing automation capabilities by integrating AI-driven approaches for predictive fault detection and adaptive learning. Generating synthetic log data using GAN-based techniques will also be explored to address data scarcity issues and train AI systems effectively. Additionally, developing tools that combine log analysis, product information, and AI techniques will be prioritized to support real-world troubleshooting applications.

Testing the proposed methodology in industrial settings will be critical to validating its scalability and effectiveness. Collaborations between domain experts, AI developers, and engineers will be encouraged to improve data quality, expand datasets, and refine predictive maintenance pipelines. Balancing automation with strategic human oversight will remain essential for ensuring reliability and adaptability in complex troubleshooting scenarios.

By addressing these challenges and opportunities, this research contributes to advancing smart troubleshooting systems and supporting the evolution of Industry 4.0.

Bibliography

[1] Mario Hermann, Tobias Pentek, and Boris Otto. Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 3928–3937, 2016.

[2] Baotong Chen, Jiafu Wan, Lei Shu, Peng Li, Mithun Mukherjee, and Boxing Yin. Smart factory of industry 4.0: Key technologies, application case, and challenges. *IEEE Access*, 6:6505–6519, 2018.

[3] Joel Höglund, Simon Bouget, Martin Furuhed, John Preuß Mattsson, Göran Selander, and Shahid Raza. AutoPKI: public key infrastructure for IoT with automated trust transfer. 23(3):1859–1875.

[4] Prasannjeet Singh, Mehdi Saman Azari, Francesco Vitale, Francesco Flammini, Nicola Mazzocca, Mauro Caporuscio, and Johan Thornadtsson. Using log analytics and process mining to enable self-healing in the internet of things. 42(2):234–250.

[5] Mauro Caporuscio, Francesco Flammini, Narges Khakpour, Prasannjeet Singh, and Johan Thornadtsson. Smart-troubleshooting connected devices: Concept, challenges and opportunities. *Future Generation Computer Systems*, 111:681–697, 2020.

[6] Sania Partovian, Alessio Bucaioni, Francesco Flammini, and Johan Thornadtsson. Analysis of log files to enable smart-troubleshooting in industry 4.0: a systematic mapping study. *IEEE Access*, pages 1–1, 2023.

[7] Bin Xia, Yuxuan Bai, Junjie Yin, Yun Li, and Jian Xu. LogGAN: a log-level generative adversarial network for anomaly detection using permutation event modeling. *Information Systems Frontiers*, 23(2):285–298, 2021.

[8] Steven Locke, Heng Li, Tse-Hsun Peter Chen, Weiyi Shang, and Wei Liu. Logassist: Assisting log analysis through log summarization. *IEEE Transactions on Software Engineering*, 48(9):3227–3241, 2022.

[9] Marcello Cinque, Christian Esposito, and Antonio Pecchia. Security log analysis in critical industrial systems exploiting game theoretic feature selection and evidence combination. *IEEE Transactions on Industrial Informatics*, 16(6):3871–3880, 2020.

[10] Fatma Bozyigit, Tolgahan Bardakci, Alireza Khalilipour, Moharram Challenger, Guus Ramackers, Önder Babur, and Michel RV Chaudron. Generating domain models from natural language text using nlp: a benchmark dataset and experimental comparison of tools. *Software and Systems Modeling*, pages 1–19, 2024.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[12] Jorij Abraham. *Product Information Management: Theory and Practice*. Management for Professionals. Springer International Publishing.

[13] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[14] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[15] Ndala Yves Mulongo. Industry 5.0 a novel technological concept. In *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pages 1–6, 2024.

[16] Dietmar P. F. Möller, Hamid Vakilzadian, and Roland E. Haas. From industry 4.0 towards industry 5.0. In *2022 IEEE International Conference on Electro Information Technology (eIT)*, pages 61–68, 2022.

[17] Qiang Fu, Jian-Guang Lou, Yi Wang, and Jiang Li. Execution anomaly detection in distributed systems through unstructured log analysis. In *2009 Ninth IEEE International Conference on Data Mining*, pages 149–158, 2009.

[18] K Kent and M P Souppaya. Guide to computer security log management, 2006.

[19] Madhu Raut, Sunita Dhavale, Amarjit Singh, and Atul Mehra. Insider threat detection using deep learning: A review. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 856–863.

[20] Akihiro Tomono, Minoru Uehara, and Yuji Shimada. Improvement and evaluation of a method to manage multiple types of logs. In *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, pages 601–606, 2011.

[21] Byron Stuike and Yasaman Amannejad. Pairwise application log classification. In *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 349–350, 2023.

[22] Leila Karimi, Maryam Aldairi, James Joshi, and Mai Abdelhakim. An automatic attribute-based access control policy extraction from access logs. *IEEE Transactions on Dependable and Secure Computing*, 19(4):2304–2317, 2022.

[23] Aidong Xu, Lin Chen, Yixin Jiang, Huahui Lv, Hang Yang, and Bo Li. Finding gold in the sand: Identifying anomaly indicators though huge amount security logs. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 140–144, 2020.

[24] Pratibha Sharma, Surendra Yadav, and Brahmdutt Bohra. A review study of server log formats for efficient web mining. In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pages 1373–1377, 2015.

[25] R. Khandpur. *Troubleshooting Electronic Equipment*. McGraw Hill TAB, 1st edition edition.

[26] Glenn A. Mazur Mazur and Thomas Proctor. *Troubleshooting Electrical/-Electronic Systems*. American Tech Publishers, 3rd edition edition.

[27] Robert E. Horn, Elizabeth H. Nicol, Joel C. Kleinman, and Michael G. A. Grace. Information mapping for learning and reference. 1969.

[28] Douglas Garcia Torres. *Generation of Synthetic Data with Generative Adversarial Networks*. PhD thesis, KTH ROYAL INSTITUTE OF TECHNOLOGY, 10 2018.

[29] Yuktaa Tiwari, Akhtar Rasool, and Gaurav Hajela. Machine learning with generative adverserial network. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 543–548, 2020.

[30] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. 58(1):49–69.

[31] Sania Partovian, Francesco Flammini, and Alessio Bucaioni. Leveraging gans to generate synthetic log files for smart-troubleshooting in industry 4.0. In *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 1–8, 2024.

[32] Stefan Cronholm and Hannes Göbel. Action design research: integration of method support. *International Journal of Managing Projects in Business*, 15(8):19–47, 2022.

[33] Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. *Information and software technology*, 2013.

[34] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *Procs of EASE*, 2008.

[35] Jeanderson Barros Cândido, Maurício Finavaro Aniche, and Arie van Deursen. Log-based software monitoring: a systematic mapping study, 2021.

[36] Anukampa Behera, Chhabi Rani Panigrahi, and Bibudhendu Pati. Unstructured log analysis for system anomaly detection—a study. In Samarjeet Borah, Sambit Kumar Mishra, Brojo Kishore Mishra, Valentina Emilia Balas, and Zdzislaw Polkowski, editors, *Advances in Data Science and Management*, Lecture Notes on Data Engineering and Communications Technologies, pages 497–509. Springer Nature, 2022.

[37] Rawand Raouf Abdalla and Alaa Khalil Jumaa. Log file analysis based on machine learning: A survey: Survey. 6(2):77–84, 2022.

[38] Boyuan Chen and Zhen Ming (Jack) Jiang. A survey of software log instrumentation. 54(4):90:1–90:34, 2021.

[39] Shubham Jain, Amy de Buitléir, and Enda Fallon. *A Review of Unstructured Data Analysis and Parsing Methods*.

[40] Mukhtar Hussain, Ernest Foo, and Suriadi Suriadi. An improved industrial control system device logs processing method for process-based anomaly detection. In *2019 International Conference on Frontiers of Information Technology (FIT)*, pages 150–1505. ISSN: 2334-3141.

[41] Pinjia He, Jieming Zhu, Shilin He, Jian Li, and Michael R. Lyu. Towards automated log parsing for large-scale log data analysis. 15(6):931–944. Conference Name: IEEE Transactions on Dependable and Secure Computing.

[42] Yun Qian, Shi Ying, and Bingming Wang. Anomaly detection in distributed systems via variational autoencoders. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2822–2829. ISSN: 2577-1655.

[43] J. Liu, Z. Hou, and Y. Li. Lopper: An efficient method for online log pattern mining based on hybrid clustering tree. 11706 LNCS:63–78. ISBN: 9783030276140.

[44] 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)*, pages 33–40.

[45] D. Pasha, A.H. Shah, E.H. Zadeh, and S. Konur. Anomaly detection and root cause analysis on log data. 13652 LNAI:333–339. ISBN: 9783031214400.

[46] Q. Du, L. Zhao, J. Xu, Y. Han, and S. Zhang. Log-based anomaly detection with multi-head scaled dot-product attention mechanism. 12923 LNCS:335–347. ISBN: 9783030864712.

[47] Z. Jadidi, E. Foo, M. Hussain, and C. Fidge. Automated detection-in-depth in industrial control systems. 118(7):2467–2479.

[48] Shenghui Gu, Guoping Rong, He Zhang, and Haifeng Shen. Logging practices in software engineering: A systematic mapping study. 49(2):902–923, 2023.

[49] Prasannjeet Singh, Francesco Flammini, Mauro Caporuscio, Mehdi Samanazari, and Johan Thornadtsson. Towards self-healing in the internet of things by log analytics and process mining.

[50] Sabrina Luftensteiner and Patrick Praher. A synthetic dataset for anomaly detection of machine behavior. In Gabriele Kotsis, A. Min Tjoa, Ismail Khalil, Bernhard Moser, Alfred Taudes, Atif Mashkoor, Johannes Sametinger, Jorge Martinez-Gil, Florian Sobieczky, Lukas Fischer, Rudolf Ramler, Maqbool Khan, and Gerald Czech, editors, *Database and Expert Systems Applications - DEXA 2022 Workshops*, pages 424–431, Cham, 2022. Springer International Publishing.

[51] Jingyu Sun, Bingyu Liu, and Yuan Hong. Logbug: Generating adversarial system logs in real time. In *Proceedings of the 29th ACM International*

Conference on Information & Knowledge Management, CIKM '20, page 2229–2232, New York, NY, USA, 2020. Association for Computing Machinery.

[52] Sungil Kim, Kwang Hyun Kim, Baehyun Min, Jungtek Lim, and Kyungbook Lee. Generation of synthetic density log data using deep learning algorithm at the golden field in alberta, canada. *Geofluids*, 2020:Article ID 5387183, 26 pages, 2020.

[53] Markus Wurzenberger, Florian Skopik, Giuseppe Settanni, and Wolfgang Scherrer. Complex log file synthesis for rapid sandbox-benchmarking of security- and computer network analysis tools. *Information Systems*, 60:13–33, 2016.

[54] Long Chen, Yanqing Song, and Jianguo Chen. A framework for few-shot network threats based on generative adversarial networks. In *2023 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2023.

[55] Bin Xia, Yuxuan Bai, Junjie Yin, Yun Li, and Jian Xu. Loggan: a log-level generative adversarial network for anomaly detection using permutation event modeling. *Information Systems Frontiers*, 23(2):285–298, 2021.

[56] Xiaoyu Duan, Shi Ying, Wanli Yuan, Hailong Cheng, and Xiang Yin. A generative adversarial networks for log anomaly detection. *Computer Systems Science & Engineering*, 37(1):135–148, 2021.