Trustworthy anomaly detection for smart manufacturing

Li, Z.

Chapter 8

Conclusions and Future Directions

In this dissertation, we developed advanced anomaly detection approaches with a primary focus on enhancing the manufacturing processes of high-tech systems. However, it is important to emphasize that many of the techniques and findings discussed here are applicable to broader scenarios beyond manufacturing, and in some cases, were even designed for more general-purpose use. Given the complexity of achieving trustworthy anomaly detection in smart manufacturing, we approached this challenge from two complementary perspectives—data-centric AI and model-centric AI—leading to two key research questions, each with several sub-questions.

The first research question focuses on improving anomaly detection from a **data-centric AI perspective**. In this approach, the emphasis is placed on systematically processing and engineering the data that serves as input for the anomaly detection systems. For instance, we explored ways to handle *complex data*—such as system logs, time series, and graphs—by introducing novel methods that effectively manage and extract meaningful patterns from such data. Additionally, we investigated the challenges posed by *high-dimensional data*, which arises even after transforming complex data into simpler forms. High dimensionality often undermines the effectiveness of traditional anomaly detection approaches.

The second research question centers on the **model-centric AI perspective**, which focuses on optimizing the anomaly detection models themselves. One critical issue we explored in this context is the *explainability* of AI models. While neural network-based models can excel in detecting anomalies, their lack of transparency often leaves end-users in the dark about how decisions are made. Furthermore, we examined *the robustness of post-hoc explanations*, particularly in the face of adversarial attacks. In addition to explainability, we explored *generalizability*, which is essential for deploying anomaly detection algorithms in new, unseen environments. This is particularly important when systems undergo updates or modifications, such as changes in robots or software. Finally, we investigated approaches to improving the *automatability* in anomaly detection, particularly in unsupervised settings where labeled data is scarce.

In summary, this dissertation provides a comprehensive framework for developing robust, explainable, and generalizable anomaly detection systems, offering solutions to both data-related and model-related challenges in smart manufacturing. Our approaches contribute to improving the reliability and efficiency of manufacturing processes, while also providing insights applicable to broader fields of anomaly detection.

In the remainder of this chapter, we will begin by summarizing the key findings from each chapter to address the research questions outlined in Chapter 1. This will provide a cohesive overview of how the research outcomes contribute to answering these questions. Afterward, we will explore the limitations and challenges encountered across multiple chapters, highlighting the areas that require further attention. This discussion aims to offer valuable insights into unresolved issues and guide potential future research directions.

8.1 Conclusions

In Chapter 1, we outlined six key research questions. In this subsection, we provide detailed answers to each of these questions, drawing upon the findings presented in Chapters 2 through 7. For each research question, we summarize the main conclusions, highlighting how the results contribute to addressing the research objectives.

8.1.1 Develop and/or Improve Anomaly Detection for Smart Manufacturing from A Data-Centric AI Perspective

Q1.1 **How to deal with complex data in system logs to effectively detect and explain anomalies?**

In response to this problem, we introduced *Logs2Graphs*, a new approach for unsupervised log anomaly detection. It first converts log files to attributed, directed, and edge-weighted graphs, translating the problem to an instance of graph-level anomaly detection. Next, this problem is solved by OCDiGCN, a novel method based on graph neural networks that performs graph representation learning and graph-level anomaly detection in an end-to-end manner. Important properties of OCDiGCN include that it can deal with directed graphs and do unsupervised learning. Moreover, we furnish a concise set of nodes pivotal in OCDiGCN's prediction as explanations for each detected anomaly, offering valuable insights for subsequent root cause analysis. Extensive results on five benchmark datasets reveal that *Logs2Graphs* is at least comparable to and often outperforms state-of-the-art log anomaly detection methods such as DeepLog [65] and LogAnomaly [66].

The above findings are all described in Chapter 2. The main conclusion to this research question is that: by leveraging the rich and expressive power of attributed, directed, and edge-weighted graphs to represent logs, followed by using graph neural networks to effectively detect graph-level anomalies, taking into account both semantic information of log events and structure information among log events, we can better detect anomalies in system logs. Moreover, by decomposing the anomaly score of a graph into individual nodes and visualizing these nodes based on their contributions, we provide understandable explanations for identified anomalies.

Q1.2 **How to handle high-dimensionality in system logs to effectively detect anomalies?**

By regarding each log event as a feature, we developed a feature selection method that aims to select relevant features for log-based fault detection and prediction. In brief, our method consists of three main modules, namely *Log Event Vectorization*, *Selection of Relevant Features* and *Removal of Redundant Features*. Specifically, the *Log Event Vectorization* module aims at converting unstructured log events into time series data; the *Selection of Relevant Features* module attempts to select relevant features for fault detection and prediction by using the variables measured by sensors as target; and the *Removal of Redundant Features* module focuses on eliminating redundant features to further reduce the number of selected features. Extensive experiments on real-world datasets show that our proposed feature selection method can help improve log-based anomaly detection performance. Specifically, based on the selected log features, KNN [124] was able to accurately detect or predict faults in 24 out of 25 machines.

In contrast, KNN can only accurately detect or predict faults in 17 out of 25 datasets based on all log features. One possible reason is that logs from all subsystems are entangled and the inclusion of many irrelevant log events renders the detection/prediction of gradual faults difficult.

The above findings are described in Chapter 3. The main conclusion is that: by considering the associations between sensor data and system logs, we can select relevant and non-redundant log events as features. As a result, the accuracy of log-based fault detection and prediction can be significantly improved, as irrelevant log events often obscure gradual fault patterns.

8.1.2 Develop and/or Improve Anomaly Detection for Smart Manufacturing from A Model-Centric AI Perspective

Q2.1 **How to achieve intrinsic explainability of anomaly detection systems?**

To explore this problem, we for the first time explicitly establish a connection between dependency-based traditional anomaly detection methods and contextual anomaly detection methods. On this basis, we propose a novel approach to contextual anomaly detection and explanation. Specifically, we use Quantile Regression Forests to develop an accurate and interpretable anomaly detection method, QCAD, that explores dependencies between features. QCAD can handle tabular datasets with mixed contextual features and numerical behavioral features. Extensive experiment results on various synthetic and real-world datasets demonstrate that QCAD outperforms state-of-the-art anomaly detection methods in identifying contextual anomalies in terms of accuracy and interpretability. Particularly, the beanplot-based visualizations help to explain why a certain object is (not) considered an anomaly within its context.

These findings are described in Chapter 4. The main conclusion is that: by establishing a connection between dependency-based traditional anomaly detection methods and contextual anomaly detection methods, we developed a self-interpretable anomaly detection method QCAD, which can effectively identify contextual anomalies by exploring dependencies between features and provide beanplot-based visualizations to further explain why an object is (or is not) considered an anomaly.

Q2.2 **Are post-doc explanations for Graph Neural Networks robust to adversarial attacks?**

GNN explanations for enhancing transparency and trust of graph neural networks are becoming increasingly important in decision-critical domains. We showed that existing GNN explanation methods are vulnerable to adversarial perturbations, namely small *prediction-preserving* perturbations can result in largely different explanations generated by post-hoc GNN explainers. To achieve this, we performed perturbations that are both carefully crafted (i.e., *optimized* rather than *random*) and imperceptible (i.e., such that the GNN predictions remain unchanged but their explanations differ substantially). More concretely, we devise *GXAttack*, the first *optimization-based* adversarial attack on post-hoc GNN explanations under this setting. We employ a widely used GNN explainer, PGExplainer [199], as example target when designing our attack algorithm. Results on various datasets demonstrate the effectiveness of our approach. Moreover, our experiments show that other widely used GNN explanation methods, such as GradCAM [197], GNNExplainer [195], and SubgraphX [201], are also fragile under the attacks optimized for PGExplainer.

These findings are described in Chapter 5. The main conclusion is that: existing GNN explanation methods are vulnerable to adversarial perturbations, leading to drastically different explanations while maintaining the predictions unchanged. This is achieved by GXAttack, the first optimization-based adversarial white-box attack on post-hoc GNN explanations. Additionally, other widely used GNN explanation methods, such as GradCAM, GNNExplainer, and SubgraphX, are also fragile under attacks optimized for PGExplainer.

Q2.3 **How to detect graph-level anomalies in an unseen target domain with the help of labeled normal graphs from a different but related source domain?**

Being motivated and supported by domain adaptation theory [237], we propose an unsupervised domain adaptation based graph level anomaly detection method called ARMET. It leverages an adversarial learning approach consisting of four main components. First, to learn graph level representations, it utilizes a two-part feature extractor: a semantic feature extractor to jointly preserve the semantic and topological information of each graph, and a structure feature extractor to extract the structure of each graph domain. Second, a domain classifier is learned to make graph level representations domain-invariant, thereby reducing the domain discrepancy. Third, a one-class classifier is trained using normal source graphs, aiming to make the learned graph level representations label-discriminative. Finally, a class aligner is trained to align normal graphs in both domains while separating anomalous graphs and normal graphs in the target domain. As a result, in an end-to-end manner, ARMET can learn both domain-invariant and label-discriminative graph level representations, and thus effectively identify anomalous graphs from the target domain. Experiments on seven benchmark datasets show that the proposed method largely outperforms state-of-the-art methods.

These findings are described in Chapter 6. The main conclusion is that: by learning domain-invariant and label-discriminative graph-level representations through adversarial learning, ARMET can detect graph-level anomalies in an unseen target domain by leveraging labeled normal graphs from a related source domain, significantly outperforming state-of-the-art methods across multiple benchmark datasets.

Q2.4 **How to automatically tune hyperparamaters in anomaly detection systems without relying on labels?**

Self-Supervised Learning (SSL) has received much attention in recent years, and many recent studies have explored SSL to perform unsupervised graph anomaly detection. However, we found that most existing studies tune hyperparameters arbitrarily or selectively (i.e., guided by labels), and our empirical findings reveal that most methods are highly sensitive to hyperparameter settings. Using label information to tune hyperparameters in an unsupervised setting, however, is label information leakage and leads to severe overestimation of model performance. To mitigate this issue, we introduce AutoGAD, the first automated hyperparameter selection method for SSL-based unsupervised graph anomaly detection. AutoGAD, consists of two parts: 1) an unsupervised performance metric which is based on an internal evaluation strategy, and 2) an effective search method which leverages discretization and grid search that works well in practice. Extensive experiments demonstrate the effectiveness of our proposed strategy. Overall, we aim to raise awareness to the label information leakage issue in the unsupervised graph anomaly detection field, and AutoGAD provides a first step towards achieving truly unsupervised SSL-based graph anomaly detection.

These findings are described in Chapter 7. The main conclusion is that: AutoGAD addresses the critical issue of label information leakage by using an unsupervised performance metric combined with an straightforward grid search strategy, ensuring hyperparameter selection without relying on labels. This approach mitigates overestimation of model performance and paves the way for truly unsupervised graph anomaly detection.

8.2 Limitations and Future Directions

Throughout the research presented in this dissertation, we identified a number of research challenges that may offer opportunities for future research, which we will summarize next.

8.2.1 Limitations of Proposed Methods

First, we discuss the limitations of the proposed methods in each chapter and suggest potential future research directions to mitigate these limitations.

8.2.1.1 Limitations of Logs2Graphs (Chapter 2)

We identify several factors that could potentially impact the validity of our findings related to the Logs2Graphs method introduced in Chapter 2:

- **Limited Datasets.** Our experimental protocol entails utilizing five publicly available log datasets, which have been commonly employed in prior research on log-based anomaly detection. However, it is important to acknowledge that these datasets may not fully encapsulate the entirety of log data characteristics. To address this limitation, our future work will conduct experiments on additional datasets, particularly those derived from industrial settings, in order to encompass a broader range of real-world scenarios.
- **Limited Competitors.** This study focuses solely on the experimental evaluation of eight competing models, which are considered representative and possess publicly accessible source code. However, it is worth noting that certain models such as GLAD-PAW [67] did not disclose their source code and it requires non-trivial efforts to re-implement these models. Moreover, certain other models, such as CODEtect [84], require several months to conduct the experiments on our limited computing resources. For these reasons, we excluded them from our present evaluation. In subsequent endeavors, we intend to re-implement certain models and attain more computing resources to test more models.

- **Purity of Training Data.** The purity of training data is usually hard to guarantee in practical scenarios. Although Logs2Graphs is shown to be robust to very small contamination of the training data, it is critical to improve model robustness by using techniques such as adversarial training [112] in the future.
- **Graph Construction.** The graph construction process, especially regarding the establishment of edges and assigning edge weights, adheres to a rule based on connecting consecutive log events. However, this rule may be considered overly simplistic in certain scenarios. Therefore, application-specific techniques will be explored to construct graphs in the future.

8.2.1.2 Limitations of FS4FDP (Chapter 3)

Several factors have been identified that may influence the validity of our findings concerning the FS4FDP method introduced in Chapter 3:

- **Limited Datasets.** Our experiments were conducted on 25 real-world datasets provided by industrial partners, which may not capture the full range of data diversity encountered in broader applications. In future work, we plan to incorporate additional datasets to further assess the generalizability of FS4FDP across varied contexts and investigate the effects of hyperparameter tuning on performance.
- **Limited Anomaly Detectors.** Currently, we only evaluated FS4FDP using a restricted selection of anomaly detectors, denoted as $\phi(\cdot)$ and $\varphi(\cdot)$. Future efforts will expand this selection to include a broader array of anomaly detection methods, enabling a more comprehensive analysis of FS4FDP's adaptability and effectiveness when paired with different detection techniques.
- **More Studies on Causal Relationships**. Particularly, the Equation (3.4) used in Granger causality requires several explicit and implicit assumptions to effectively identify Granger causal effects. Some of these assumptions may not be fulfilled by our use-case though. Therefore, we will further explore and improve Granger causality test [125] or similar techniques to find log events that can be used to predict sensor time series anomalies. Furthermore, we have not fully explored the causal relationships between different log events. In the future, by constructing a causality graph using techniques such as the PC-algorithm [126] on log events, we can investigate the causal relationships between log events. As a result, it might be possible to pinpoint the root causes of anomalies.

8.2.1.3 Limitations of QCAD (Chapter 4)

The QCAD method presented in Chapter 4 has the following limitations, which we aim to address in future work:

- **Static Features**. Currently, QCAD is limited to static features in both contextual and behavioral spaces. We plan to expand QCAD's capabilities to handle streaming data in future versions.
- **Constraints on Behavioral Space**. At present, QCAD is restricted to numerical features within the behavioral space. Future efforts will focus on extending QCAD to accommodate categorical features and mixed feature types in the behavioral space.

8.2.1.4 Limitations of GXAttack (Chapter 5)

Several factors may influence the validity of our findings concerning the GXAttack method presented in Chapter 5, and we plan to mitigate them in the future:

- **Low Scalability.** GXAttack suffers from low scalability (large requirement of memory), which however can be mitigated by using the Projected Randomized Block Coordinate Descent (PR-BCD) [228].
- **Limited Datasets.** We only employed synthetic datasets. While real-world datasets can provide quantitative evaluation results using metrics like cosine similarity (to measure the consistency of explanations before and after attacks), these metrics may not accurately reflect the true effectiveness of the attacks (i.e., in terms of ΔGEA). This discrepancy suggests that quantitative results can be challenging to interpret, and relying solely on such metrics might not yield meaningful insights. On the other hand, qualitative evaluations are also possible, but as machine learning researchers, our expertise lies not in the realm of such interpretations.
- **Differentiable Target.** GXAttack requires that the target GNN explainer to be differentiable, which is not always the case. However, the attack transferability shows its potential effectiveness on other GNN explainers.
- **Prediction-preserving Assumption.** We (implicitly) impose an assumption that a small L_0 perturbation that is prediction-preserving also preserves the causal reasoning for the GNN prediction. Unlike for images or texts, this is not

easy to verify for graph data. However, we can leverage visualizations to help understand the reasoning process.

8.2.1.5 Limitations of ARMET (Chapter 6)

The ARMET method introduced in Chapter 6 has certain limitations that we intend to explore further in future research:

- **Limited Datasets.** Our experimental protocol included only four publicly available log datasets and three image-based datasets, which do not fully capture the diversity of data characteristics encountered in real-world applications. To overcome this limitation, future work will focus on experiments using additional datasets, especially those derived from molecular, finance, and social networks, to better represent a wide range of practical scenarios.
- **Transferability Across Graph Domains.** Currently, there is limited understanding of how well ARMET generalizes across various graph domains. Future research should focus on evaluating and quantifying the transferability of ARMET across different types of graph-structured data, enabling broader applicability and enhancing its robustness in diverse domains.

8.2.1.6 Limitations of AutoGAD (Chapter 7)

The AutoGAD method presented in Chapter 7 has the following limitations, which we plan to address in future work:

- **Limited Hyperparameters Search Space.** In our experimental setup, we explored a restricted search space for hyperparameters within each SSL-based GAD method, constrained by available computational resources. In future work, we aim to expand this search space to allow for a more comprehensive exploration of hyperparameters, which may uncover more optimal configurations and improve overall performance.
- **Basic Hyperparameters Search Strategy.** Currently, our approach relies on grid search for hyperparameter tuning. Future research will investigate more sophisticated search techniques, such as Bayesian optimization, and genetic algorithms, to increase efficiency and precision in identifying optimal hyperparameter settings.

8.2.2 Other Future Directions

In addition to challenges directly associated with our proposed methods, we identify the following research directions that should receive more attention in the future.

8.2.2.1 Definition of Anomaly and Trustworthy Anomaly Detection

A long-standing problem in anomaly analysis is the lack of a uniform definition of an anomaly, leading to a wide range of anomaly detection methods [46]. The diversity of anomaly definitions and anomaly detection methods leads to the need for a large variety of trustworthy anomaly detection (TAD) techniques. Although is not necessarily problematic on itself (and may be unavoidable), the lack of uniform definitions for anomaly detection and TAD hampers communication of researchers between different (sub)fields, such as computer vision, natural language processing, data mining, and social science. This makes it hard to find related work and leads to the re-invention of methods, causing unnecessary delays in scientific progress. More importantly, the evaluation and comparison of TAD methods becomes difficult and subjective, due to the lack of a uniform, objective, and precise definition of TAD.

8.2.2.2 Enhancing Anomaly Detection for Smart Manufacturing through the Integration of Model-Centric and Data-Centric AI Approaches

Most existing anomaly detection methods (not limited to the context of smart manufacturing) are predominantly developed or enhanced from a model-centric AI perspective [27, 28, 29, 30, 31, 32, 33, 34, 35, 36]. This means that researchers focus on improving anomaly detection systems by refining algorithms to perform more effectively on existing, often fixed, datasets. These efforts involve designing better model architectures, tuning hyperparameters, developing more efficient optimization techniques, and proposing new models types or learning paradigms.

Although there has been a growing shift from model-centric AI to data-centric AI within the broader data mining and machine learning communities [365], research on data-centric anomaly detection remains limited and insufficient [366, 367, 368, 369, 370, 371, 372, 373, 374]. We therefore call for increased research efforts to address the anomaly detection problem from a data-centric AI perspective. However, we also argue that model-centric and data-centric AI approaches are inherently complementary. While data-centric AI deserves more attention, it should not overshadow the importance of model-centric AI. Effectively tackling challenges in smart manufacturing requires both the methods of action (“how-to”) and deep insights hidden from the data (“what-is”) [48]. Thus, advanced research should approach the anomaly detection problem by integrating both model-centric and data-centric AI perspectives, recognizing them as dual forces driving progress in smart manufacturing.

8.2.2.3 Anomaly Detection in the Era of Large Language Models

Large Language Models (LLMs) have recently shown their superior performance not only in traditional natural language processing tasks like language comprehension and summarization but also in a wider array of applications thanks to their advanced comprehension and generative abilities [375, 376]. To unlock the full potential of LLMs beyond textual data, researchers are expanding these models into multi-modal tasks, such as vision-language understanding and generation, which are collectively referred to as Multimodal LLMs (MLLMs) [377]. Leveraging the zero- and few-shot reasoning capabilities of LLMs and MLLMs, researchers are increasingly applying these models to anomaly detection, yielding promising results [378, 379].

The integration of LLMs and MLLMs into anomaly detection has dramatically shifted the traditional learning paradigm, which previously relied mostly on statistical models and traditional machine learning techniques. Xu and Ding [378] categorize these approaches into three groups based on the primary roles played by LLMs in anomaly detection:

- **LLMs for augmentation**: In this approach, LLMs are not directly responsible for detecting anomalies but instead enhance the detection process through their advanced semantic understanding and vast knowledge. LLMs act as data augmenters, generating meaningful information to improve anomaly detection. This includes producing effective text embeddings (using LLMs as feature extractors), generating high-quality synthetic datasets with pseudo-labels, and creating detailed textual descriptions of both normal and abnormal instances.

- **LLMs for detection**: In these methods, LLMs are employed directly as anomaly detectors. This can involve prompting-based approaches in which LLMs generate detection results in response to specific prompts, or contrastive learning approaches that utilize MLLMs pretrained with contrastive objectives to detect anomalies.

- **LLMs for explanation**: Here, LLMs offer detailed explanations and insights regarding the results of anomaly detection. These explanations help address real-world challenges by providing deeper understanding and interpretability of the detected anomalies.

While the application of LLMs and MLLMs in anomaly detection has garnered increasing attention, it remains a relatively underexplored area. Specifically, there are several key directions for future research. First, efforts should prioritize improving the trustworthiness of LLMs and MLLMs in anomaly detection, focusing on aspects such as effectiveness, explainability, and robustness. This will be crucial to ensuring their broader adoption, particularly in critical fields like smart manufacturing. Second, given the diverse range of data modalities involved in smart manufacturing—such as images, videos, time series, and system logs—the potential of MLLMs to handle multimodal data is significant. Harnessing this capability can unlock new opportunities for more advanced and accurate anomaly detection systems within the industry.