UMEÅ UNIVERSITY

Machine Learning for Anomaly Detection in Edge Clouds

Javad Forough

Doctoral Thesis, February 2024
Department of Computing Science
Umeå University
Sweden

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

javad.forough@cs.umu.se

ISBN 978-91-8070-291-1 (print)
978-91-8070-292-8 (digital)
ISSN 0348-0542
UMINF 24.02

Printed by Cityprint i Norr AB, Umeå, 2024

"As shadows reveal the unseen, anomalies illuminate the path to discovery!"

Abstract

Edge clouds have emerged as an essential architecture, revolutionizing data processing and analysis by bringing computational capabilities closer to data sources and end-users at the edge of the network. Anomaly detection is crucial in these settings to maintain the reliability and security of edge-based systems and applications despite limited computational resources. It plays a vital role in identifying unexpected patterns, which could indicate security threats or performance issues within the decentralized and real-time nature of edge cloud environments. For example, in critical edge applications like autonomous vehicles, augmented reality, and smart healthcare, anomaly detection ensures the consistent and secure operation of these systems, promptly detecting anomalies that might compromise safety, performance, or user experience. However, the adoption of anomaly detection within edge cloud environments poses numerous challenges.

This thesis aims to contribute by addressing the problem of anomaly detection in edge cloud environments. Through a comprehensive exploration of anomaly detection methods, leveraging machine learning techniques and innovative approaches, this research aims to enhance the efficiency and accuracy of detecting anomalies in edge cloud environments. The proposed methods intend to overcome the challenges posed by resource limitations, the lack of labeled data specific to edge clouds, and the need for accurate detection of anomalies. By focusing on machine learning approaches like transfer learning, knowledge distillation, reinforcement learning, deep sequential models, and deep ensemble learning, this thesis endeavors to establish efficient and accurate anomaly detection systems specific for edge cloud environments.

The results demonstrate the improvements achieved by employing machine learning methods for anomaly detection in edge clouds. Extensive testing and evaluation in real-world edge environments show how machine learning-driven anomaly detection systems improve identification of anomalies in edge clouds. The results highlight the capability of these methods to achieve a reasonable trade-off between accuracy and computational efficiency. These findings illustrate how machine learning-based anomaly detection approaches contribute to building resilient and secure edge-based systems.

Sammanfattning

Kantmoln har framträtt som en avgörande arkitektur och revolutionerat datahantering och analys genom att utnyttja beräkningskapacitet närmare datakällor och användare vid kanten av nätverket. Avvikelsedetektering är avgörande i dessa sammanhang för att bibehålla pålitligheten och säkerheten hos dessa kantbaserade system och applikationer trots begränsade beräkningsresurser. Detta spelar en betydande roll i att identifiera oväntade mönster, vilka kan indikera säkerhetshot eller prestandaproblem inom de decentraliserade och realtidsmässiga kantmolnmiljöerna. Inom kritiska applikationer vid nätverkets kant såsom autonoma fordon, augmented reality och smart hälsovård, säkerställer avvikelsedetektion den konsekventa och säkra driften av dessa system, genom att snabbt upptäcka avvikelser som kan kompromissa säkerhet, prestanda eller användarupplevelse. Emellertid innebär införandet av avvikelsedetektion inom kantmolnmiljöer många utmaningar.

Denna avhandling syftar till att bidra genom att ta itu med problemet av avvikelsedetektion i kantmolnmiljöer. Genom en omfattande utforskning av avvikelsedetektionsmetoder med hjälp av maskininlärningstekniker och innovativa tillvägagångssätt, syftar denna forskning till att förbättra effektiviteten och noggrannheten vid detektering av avvikelser i kantmolnmiljöer. De föreslagna metoderna avser att övervinna utmaningar som begränsad beräkningskraft, bristen på märkt data för kantmoln samt oförlitlig identifiering av avvikelser. Genom att fokusera på maskininlärningsmetoder som överföringsinlärning, kunskapsdestillering, förstärkningsinlärning, djupa sekventiella modeller och djup ensembleinlärning, strävar denna avhandling efter att etablera effektiva och noggranna system för avvikelsedetektion som är specifika för kantmolnmiljöer.

Resultaten visar förbättringar som uppnåtts genom att använda maskininlärningsmetoder för avvikelsedetektion i kantmoln. Omfattande testning och utvärdering i verkliga miljöer visar hur maskininlärningsdrivna system för avvikelsedetektion förbättrar identifieringen av avvikelser i kantmoln. Resultaten belyser dessa metoders förmåga att uppnå en rimlig avvägning mellan noggrannhet och beräkningsmässig effektivitet. Dessa fynd illustrerar hur maskininlärningsbaserade tillvägagångssätt för avvikelsedetektion bidrar till att bygga robusta och säkra kantbaserade system.

Preface

This thesis starts with a brief overview of edge clouds and explores the challenges associated with detecting anomalies within these environments, focusing on improving the accuracy and efficiency of anomaly detection within edge clouds using machine learning techniques. A concise summary of the contributions made by this thesis is outlined in the six included papers:

Paper I **J. Forough**, M. Bhuyan, and E. Elmroth. Detection of VSI-DDoS Attacks on the Edge: A Sequential Modeling Approach. *In Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES)*, pp. 1-10, 2021.

Paper II **J. Forough**, M. Bhuyan, and E. Elmroth. DELA: A Deep Ensemble Learning Approach for Cross-layer VSI-DDoS Detection on the Edge. *In Proceedings of the 42nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 1155-1165, 2022.

Paper III **J. Forough**, M. Bhuyan, and E. Elmroth. Anomaly Detection and Resolution on the Edge: Solutions and Future Directions. *In Proceedings of the IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pp. 227-238, 2023.

Paper IV **J. Forough**, M. Bhuyan, and E. Elmroth. Unified Identification of Anomalies on the Edge: A Hybrid Sequential PGM Approach. *In Proceedings of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2023.

Paper V **J. Forough**, H. Haddadi, M. Bhuyan, and E. Elmroth. Efficient Anomaly Detection for Edge Clouds: Mitigating Data and Resource Constraints. *Submitted for publication, 2024.*

Paper VI **J. Forough**, M. Bhuyan, and E. Elmroth. Reinforced Model Selection for Resource Efficient Anomaly Detection in Edge Clouds. *Submitted for publication, 2024.*

Acknowledgements

Embarking on the conclusion of my doctoral studies at Umeå University, I wish to express my heartfelt gratitude to those who have been instrumental in shaping this academic journey. As I reflect on this significant chapter of my life, I am sincerely thankful for the invaluable support and guidance I have received.

First and foremost, my deepest gratitude goes to **Erik Elmroth**, my main supervisor, who generously welcomed me into his group. His unwavering support has been a guiding light throughout my PhD, fostering an environment where both research and personal growth could flourish. Erik's dedication to providing a supportive space for exploration has been truly transformative, and I am genuinely thankful for his supervision. His depth of knowledge and keen understanding of the subject matter not only facilitated a more profound exploration of my research questions but also sparked new perspectives that significantly enriched the depth and breadth of my work and made me a better researcher.

I extend my sincere appreciation to **Monowar Bhuyan**, my Co-supervisor, whose remarkable commitment to my academic journey was evident day and night throughout the entire PhD. Monowar's unwavering dedication, showcased his genuine passion for guiding and supporting my progress. His insightful feedback and thoughtful comments not only enhanced the quality of my work but also played a crucial role in refining my research methodologies and encouraging my development as a more skilled and confident researcher.

This study was made possible in part by the **Wallenberg AI, Autonomous Systems and Software Program (WASP)** funded by the Knut and Alice Wallenberg Foundation. A special thanks and mention goes to WASP for their generous financial and educational support. Their belief in my research and their ready assistance played a crucial role in realizing the success of this academic journey. Moreover, the networking opportunities they orchestrated during various events proved invaluable, allowing me to establish meaningful connections with top researchers in the field, fostering collaboration and enriching the overall academic experience. Additionally, I extend my sincere appreciation to **Eddie**, my reference person, for his unwavering support during my doctoral journey. Eddie consistently offered valuable guidance on my academic progress and future as a Ph.D. student. His insights extended beyond research, providing advice on

navigating the challenges of academia and ensuring my overall well-being. I am truly thankful for that.

I express my sincere gratitude to **Paul Townend** for being very supportive and organizing the WASP UK trip, providing us with a unique opportunity to explore some of the world's top universities. His assistance in obtaining the UK visa for this trip was also invaluable. I am also profoundly thankful to **Johan Tordsson** for his invaluable guidance and insightful feedback on my presentations, which has been instrumental in enhancing my understanding of the subject matter.

Moving forward, I want to extend my heartfelt appreciation to **Hamed Haddadi**, my host supervisor at Imperial College London, for warmly welcoming me into his exceptional group. His efforts in ensuring my comfort and seamless integration into the academic environment were instrumental in making my time at Imperial College both enriching and fulfilling. I am genuinely thankful for the valuable experiences and opportunities that have unfolded under his mentorship. Moreover, I want to express my gratitude to **Saeedeh Momtazi**, my Master of Science program supervisor, for her instrumental role in recommending me for this PhD position. Her foresight and unwavering belief in my capabilities have been pivotal in shaping my academic journey.

A very special thanks goes to **Ali**, my office mate and friend with whom I started my doctoral study. Our numerous technical discussions in various areas, such as cloud computing and machine learning, along with his insightful perspectives, have been invaluable to my academic journey. Furthermore, I express my gratitude to **Mohammad Reza**, my colleague and friend, with whom I started my doctoral study. Together, we had several insightful discussions and explored various topics related to Kubernetes.

Special thanks to **Tobias** for his assistance with documents related to the thesis and defense. Additionally, I extend my appreciation to **Oliver** for his support with the Swedish components of the kappa. Moreover, I extend my deepest appreciation to all my former and current colleagues and amazing researchers at ADSLAB specially to **Lili**, **Xuan-Son**, **Chanh**, **Yashwant**, **Sourasekhar**, **Anindya**, **Antonio**, **Lidia**, **Simon**, **Sagar**, **Maarten**, **Tanaz**, **Obaid**, and many others. Your shared pursuit of knowledge has enriched my academic experience immeasurably.

Finally, I owe an immense debt of gratitude to my family. Their unwavering support and encouragement have been the bedrock of my journey, shaping me into the person I am today. Through all the good and challenging days, their patience has been a constant source of strength. You have been there through it all, the highs and the lows, celebrating my successes and helping me navigate challenges, which played a crucial role in shaping not only my academic achievements but also the core of my character.

I want to express my heartfelt thanks to everyone who helped me through this journey. As I put these words of gratitude on paper, it feels like we achieved this together, and I'm truly thankful for that.

Contents

Chapter 1

Introduction

The evolution of computing paradigms has introduced a new era marked by the integration of edge cloud architectures, transforming the landscape of data processing and analysis. This chapter elaborates on the research motivation, objectives, and methodology of this thesis, offering a comprehensive exploration into the core motivations driving our study, the goals we aim to achieve, and the strategies we employ to conduct our research.

1.1 Research Motivation

The evolution and widespread adoption of edge cloud architectures have reshaped data processing, providing industries with real-time decision-making capabilities. Particularly crucial for emerging applications like augmented reality [Sir+21], autonomous transportation [Liu+19], smart healthcare [HHI22], and more, edge clouds have become essential. However, this transformative shift has also posed challenges in maintaining the security, reliability, and operational stability of edge-based systems and applications. Given the critical role of edge clouds in these cutting-edge applications, there is an urgent need to improve their operational resilience. This necessity highlights the need for tailored anomaly detection mechanisms designed explicitly for the dynamic and distributed nature of edge environments.

This research is motivated by the imperative to bridge this gap by designing accurate and efficient anomaly detection methods for edge cloud architectures using machine learning. As mentioned before, anomaly detection holds paramount importance for edge clouds due to their decentralized nature, where data processing occurs closer to the source. This proximity enhances real-time responsiveness but also amplifies vulnerability to anomalies. As edge clouds are the backbone of crucial applications, ensuring their uninterrupted and secure operation is vital. Hence, the development of anomaly detection mechanisms specific to edge environments becomes imperative to protect against potential disruptions,

maintain operational integrity, and ensure the reliability of these advanced applications.

Moreover, recent developments in machine learning methods have exhibited significant promise across various domains [Khe+23]. These advancements demonstrate their adaptability and effectiveness in identify intricate patterns from large datasets, making them particularly suitable for addressing complex tasks like anomaly detection. This capability is crucial in detecting anomalies within the diverse and dynamic landscape of edge environments. Therefore, this research endeavors to explore recent developments in machine learning and examine their applicability in constructing anomaly detection systems customized for edge cloud infrastructures. By utilizing the power of machine learning, this research aims to elevate anomaly detection capabilities, enabling proactive, precise, and efficient identification of anomalies in edge cloud environments.

1.2 Research Objectives

The thesis aim is to improve anomaly detection techniques specific to edge cloud environments using machine learning, aligning with the domains highlighted in Section 1.1. Anomaly detection within edge cloud environments requires innovative solutions due to their inherent challenges. The high-level research objectives are outlined as follows:

RO1: To devise, implement, and evaluate methods leveraging machine learning techniques to enhance the efficiency and accuracy of anomaly detection in edge cloud environments.

RO2: To establish an experimental testbed setup conducive to testing and validating machine learning-driven anomaly detection methods in edge cloud environments, supporting the practical evaluation and validation of the devised anomaly detection methods.

RO3: To conduct a comprehensive review of existing literature and research studies addressing anomaly detection and resolution on the edge, summarizing current practices, identifying gaps, and outlining future directions for research in this domain.

1.3 Methodology

The methodology used in this thesis adheres to a Design Science Research (DSR) [VHM20], as illustrated in Figure 1.1. The initial step involves the identification of prevalent challenges in anomaly detection within edge cloud environments and the precise definition of the addressed research problem. Concurrently, a comprehensive review of relevant literature is conducted to gain a deeper understanding of the domain and existing solutions. The subsequent phase quantitatively defines the objectives of the solution in comparison to existing

methodologies. The next step is the iterative design and development process focused on creating novel anomaly detection techniques tailored explicitly for edge cloud environments. Throughout this research, the proposed solutions are extensively evaluated in testbed edge cloud settings, leveraging a comprehensive experimental setup detailed in each research paper. Final phase encompasses the effective communication of the findings and implications of the research results and findings.

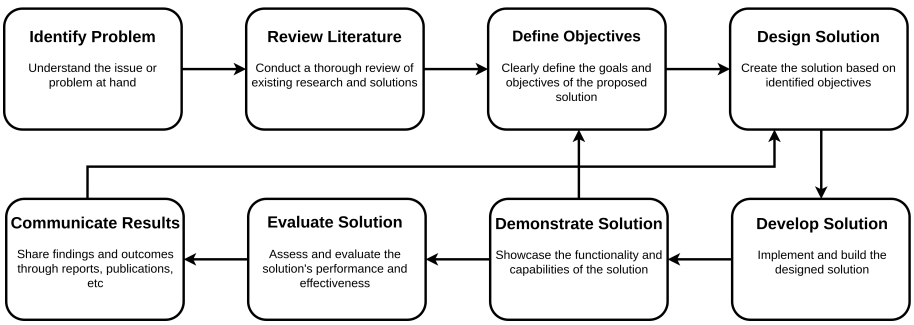

Figure 1.1: Design Research Model (DSR).

1.4 Research Contributions

This thesis focuses on developing, implementing, and evaluating machine learning methods tailored for detecting anomalies within edge cloud environments, aligning with the outlined research objectives (**ROs**) mentioned in Section 1.2. Paper I contributes to **RO1** and **RO2** by introducing a sequential modeling approach for the detection of VSI-DDoS attacks within edge cloud environments. It employs innovative techniques to model sequential anomaly patterns within a testbed edge cloud setting. Paper II continues addressing **RO1** and **RO2** by introducing a deep ensemble learning approach specifically designed to detect cross-layer VSI-DDoS attacks in a testbed edge clouds environment.

In Paper III, solutions and future research directions for addressing anomalies in edge cloud environments are discussed, aligning with **RO3**. The contribution of Paper IV lies in proposing a hybrid sequential probabilistic graphical model for detection of both security and performance anomalies in edge cloud systems, aligned with **RO1** and **RO2**. Paper V aims to address data and resource limitations in supervised anomaly detection within edge cloud environments, aligning with **RO1** and **RO2**. Finally, Paper VI focuses on enhancing resource optimization techniques for anomaly detection in edge cloud systems, making further contributions towards the objectives of **RO1** and **RO2**.

1.5 Thesis Organization

The subsequent chapters in this thesis include Chapter 2 providing an overview of edge cloud environments, general architecture, and related challenges. Chapter 3 covers essential background and fundamental concepts in anomaly detection. This chapter explores various machine learning approaches dedicated to anomaly detection within edge clouds, anomaly resolution strategies for edge clouds, along with in-depth discussions on anomaly detection considerations in such environments. Chapter 4 details the experimental setup used for validating anomaly detection methods in edge clouds, encompassing comprehensive discussions on evaluation metrics, datasets, and the testbed configuration. This includes insights into the container orchestration platform and the suite of microservice benchmarking applications, offering a comprehensive understanding of the experimental framework utilized for our validation purposes. Chapter 5 provides an in-depth presentation and analysis of the detailed contributions from each individual paper, while Chapter 6 outlines potential future research directions, presenting insightful pathways for further investigation within area of anomaly detection in edge cloud environments.

Chapter 2

The Emerging Landscape of Edge Clouds

This chapter provides an overview of edge clouds and their significance in reshaping conventional approaches to data processing and analysis. Edge clouds represent a dynamic paradigm shift by bringing computational resources closer to data sources and end users, overcoming limitations present in centralized cloud infrastructures. The hierarchical architecture of edge computing facilitates efficient data processing and analysis at the edge of the network, offering accelerated processing and reduced latency. Additionally, the decentralized nature of edge clouds enhances scalability and responsiveness, particularly in applications requiring real-time data analysis and decision-making.

2.1 Overview of Edge Cloud Environments

Edge clouds represent a dynamic shift in computational paradigms, fundamentally changing the conventional approach to data processing and analysis [Ren+19]. This architecture positions the computational resources closer to data sources and end users, which effectively circumvents the limitations inherent in centralized cloud structures. By leveraging distributed resources at the edge, edge clouds deliver accelerated data processing, minimizing latency and bandwidth constraints. The hierarchical structure of edge computing, as depicted in Figure 2.1, comprises interconnected layers, facilitating efficient data processing and analysis at the network's periphery [HAA20]. Moreover, the decentralized nature of edge clouds reduces the dependency on centralized data centers, increasing scalability and responsiveness, particularly in applications demanding real-time data analysis and decision-making [Che+17].

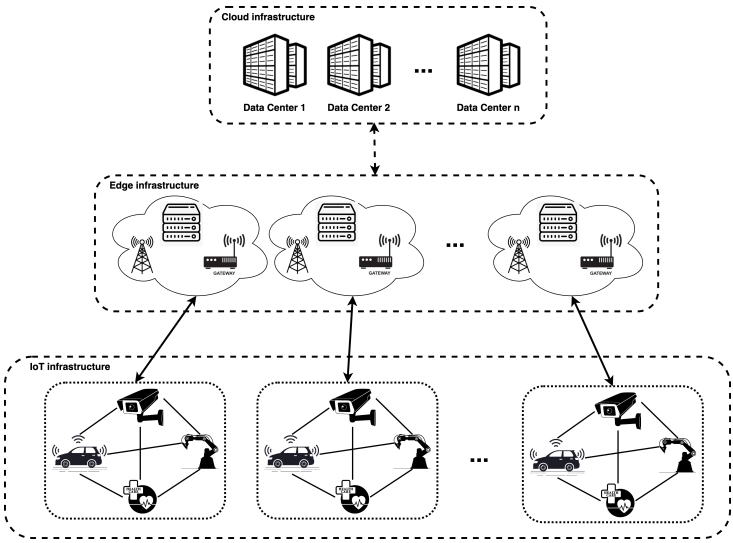

Figure 2.1: Hierarchy of edge clouds [FBE23a].

2.1.1 Edge Clouds Hierarchy

The architectural design of edge clouds embodies a hierarchical framework, encompassing interconnected layers and components that optimize computational tasks at various levels of the network [HAA20]. These layers include but are not limited to the edge devices, edge servers, and central data centers. The edge devices serve as the initial point of contact for data transmission, while edge servers, strategically positioned at the network's edge, facilitate localized data processing and analysis. Central data centers serve as the backbone, supporting extensive data storage and global synchronization [PM17]. This hierarchical structure empowers edge clouds to efficiently distribute computing tasks, ensuring timely responses and scalability.

A fundamental characteristic of edge computing infrastructures lies in their heterogeneity [Car+21], encompassing a diverse array of devices, ranging from IoT sensors, mobile devices, and smart appliances to powerful servers and gateways, each exhibiting varying computational capacities, storage capabilities, and connectivity options. This diverse environment presents unique challenges, demanding adaptive solutions to accommodate varying device capabilities while ensuring seamless interoperability. While adopting edge cloud architectures offers advantages like reduced latency, enhanced reliability, improved data privacy, and decreased network loads as a result of data processing proximity, the decentralized model introduces inherent complexities. Effectively managing the diverse and decentralized nature of edge environments, ensuring seamless resource integration, and enhancing security measures to mitigate potential

vulnerabilities become imperative challenges in this distributed computing paradigm [Shi+16].

The stratified architecture of edge clouds is instrumental in enhancing scalability and responsiveness, pivotal for meeting the crucial needs of modern applications. This architecture achieves decentralized distribution of computing resources and tasks across multiple network layers, alleviating stress on central infrastructures. It enables rapid scalability and responsiveness of computational resources, quickly adapting to varying workloads to optimize network performance and accommodate diverse application requirements. For instance, autonomous vehicles rely on edge computing for real-time decision-making based on sensor and camera data [Liu+19]. Smart cities efficiently manage IoT devices using edge computing, ensuring prompt responses to environmental changes or emergencies [Kha+20]. Healthcare systems leverage edge computing for faster diagnostics and personalized treatment recommendations based on local processing of patient data [HHI22]. Scalability and elasticity are paramount attributes in edge computing architecture, enabling effective resource scalability and smooth integration with varying workloads. Scalability helps edge systems to manage surges in data volumes and user interactions while maintaining optimal performance. Elasticity ensures resource allocation and deallocation based on demand, strategically optimizing resource utilization for cost-efficiency within the edge computing framework.

The distributed nature of edge clouds introduces intricate security challenges, demanding comprehensive measures to maintain data integrity, system functionality, and user trust. Security considerations in edge environments encompass diverse aspects, including authentication protocols, data encryption mechanisms, intrusion detection systems tailored to the distributed nature of edge infrastructures, and compliance with privacy regulations and standards [Xia+19]. With the expanded attack range in distributed edge networks, they become susceptible to various security threats and vulnerabilities. Thus, securing edge devices, communication channels, and data transmissions is critical to protect sensitive information, mitigate potential cyber threats, and ensure the integrity and confidentiality of data processed at the edge. Robust security measures, meticulously integrated across all layers of the edge cloud architecture, are imperative to enhance protection against cyber attacks, maintain user trust, and preserve privacy standards in edge computing systems.

2.2 Edge Cloud Challenges

The evolution of edge cloud environments introduces a set of challenges that must be considered to utilize their full potential. This section explores key challenges faced within the area of edge clouds, mentioning the critical aspects that demand attention.

- **Limited Resources:** Edge nodes often come with limited computational resources. These constraints include restricted processing power, limited

memory capacity, and constrained storage capabilities [Shi+16]. As a result, managing applications and workloads on these nodes is challenging, requiring careful optimization and resource allocation. Developers need to design applications that can efficiently operate with respect to these limitations, ensuring optimal performance and responsiveness. Additionally, the scarcity of resources introduces challenges related to scalability. As the number of edge nodes increases, orchestrating and distributing workloads across these devices while considering their individual resource constraints becomes a critical aspect of edge cloud management. Finding innovative solutions to address these resource limitations is essential for the successful deployment and operation of edge computing applications.

- **Security and Privacy Concerns:** The distributed nature of edge clouds introduces a multitude of security challenges [Xia+19]. Edge nodes, often dispersed across diverse locations, increase the attack surface, making them susceptible to various cyber threats. Securing communication channels between edge nodes and the central infrastructure becomes paramount to prevent unauthorized access, data breaches, and potential disruptions [KKS20]. Moreover, handling sensitive data at the edge raises significant privacy concerns [Zha+18]. Maintaining the right trade-off between providing personalized and context-aware services while ensuring the protection of user privacy requires robust encryption, authentication mechanisms, and adherence to privacy regulations. Addressing security and privacy concerns is fundamental for having trust in edge computing systems and encouraging their widespread adoption.

- **Orchestration and Management:** The orchestration and management of a vast and diverse array of edge nodes pose substantial challenges in edge cloud environments [Vañ+23]. Coordinating tasks, deploying software updates, and ensuring the overall health of the edge infrastructure demand sophisticated management solutions. The heterogeneity of edge nodes, each with its unique capabilities and limitations, adds complexity to these tasks. Efficient orchestration involves dynamically allocating workloads, optimizing resource utilization, and smoothly integrating edge nodes into the broader computing ecosystem. Additionally, managing the entire lifecycle of applications, from deployment to scaling, requires automation and intelligent decision-making. Developing standardized approaches for orchestration and management is essential to simplify operations and facilitate the scalability of edge computing deployments.

- **Interoperability and Standards:** The diverse landscape of edge computing encompasses a wide range of nodes, platforms, and vendors. Achieving seamless interoperability between different components is a paramount challenge [Kor+20]. Without well-defined standards, the integration of diverse edge nodes into a cohesive and interoperable ecosystem becomes difficult. Standardization efforts play a crucial role in establishing com-

mon interfaces, communication protocols, and data formats across the edge computing landscape. These standards enable developers to create applications that can run consistently across various edge environments, encouraging a more collaborative and interoperable ecosystem.

- **Energy Efficiency:** Energy efficiency is a critical challenge in edge cloud environments [Jia+20]. Many edge nodes are constrained by limited power sources. Balancing the need for continuous operation with the necessity to conserve energy is crucial for sustainable and long-term deployment of edge computing solutions. Optimizing algorithms, hardware components, and communication protocols to minimize energy consumption becomes imperative in addressing this challenge [Che+21]. Furthermore, dynamic workload variations and fluctuating demand for edge services require adaptive power management strategies. This involves intelligently scaling the power usage of edge nodes based on the current workload and available resources. Additionally, exploring renewable energy sources and designing energy-efficient hardware architectures [DPP21] are essential steps toward achieving a more sustainable and environmentally friendly edge computing infrastructure.

Chapter 3

Anomaly Detection and Resolution for Edge Clouds

This chapter is focused on explaining how machine learning is applied in anomaly detection. It covers various stages of anomaly detection and explores different machine learning techniques used for this purpose. Additionally, it discusses specific factors to consider when performing anomaly detection in edge clouds, clarifying the unique challenges faced in these environments.

3.1 The Role of Machine Learning in Anomaly Detection

This section explains the crucial role that machine learning plays in anomaly detection. It explores the fundamental components that form the basis of effective anomaly detection and then explains specific machine learning techniques that are applicable for this purpose.

3.1.1 Components of Anomaly Detection

There are several components involved in anomaly detection, as illustrated in Figure 3.1. These components comprise the input module, responsible for data acquisition and preprocessing; The anomaly detection module, which detects anomalies in the data; An evaluation module to assess the anomaly detection process; Visualization and explainability module for meaningful interpretation, and real-time monitoring module for tracking of the metrics and data collection. The detailed description of each component is provided in the following parts.

- *Real-time Monitoring:* Real-time monitoring module is a crucial step for observing the target system's behavior in real-time and collecting data for future training of the anomaly detection module. This stage ensures that

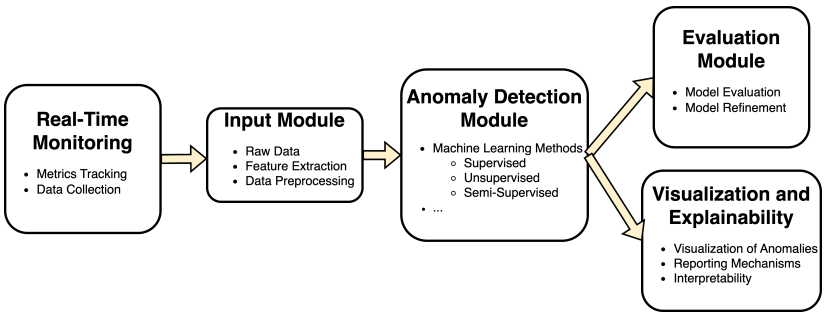

Figure 3.1: Different Components involved in anomaly detection.

the system is continuously monitored for any unusual activity, providing valuable information for improving the anomaly detection process in the future.

- *Input Module:* The input module of an anomaly detection system consists of three fundamental components: raw data, feature extraction, and data preprocessing. raw data encompasses the initial data inputs obtained from various sources such as sensors, logs, or databases. feature extraction [Mut+20] involves identifying and extracting relevant information or attributes from the raw data, enabling a more refined dataset for analysis. data preprocessing [GLH15] focuses on refining, cleaning, and transforming the raw data to address issues like noise, missing values, or inconsistencies, ensuring the data is suitable for further analysis.

- *Anomaly Detection Module:* The anomaly detection module stands as the central component in the anomaly detection process, employing a variety of techniques, with machine learning methods playing a crucial role. These machine learning methods fall into categories like supervised, unsupervised, or semi-supervised, employing algorithms to discern anomalies by recognizing patterns or behaviors learned from the dataset [Al-+21]. Supervised methods learn from labeled data, unsupervised methods detect anomalies without prior labels, and semi-supervised methods try to utilize both labeled and unlabeled data to enhance anomaly detection accuracy.

- *Evaluation Module:* The evaluation module plays a vital role in assessing the performance of the anomaly detection module. Various metrics can be examined, depending on the type of anomaly detection employed. This step is essential for measuring how effectively the system identifies anomalies and ensuring its overall performance.

- *Visualization and Explainability:* Visualization and explainability module is essential for presenting detected anomalies in a human-understandable format. This module involves creating informative visual representations

and reports that facilitate meaningful insights into detected anomalies. Effective visualization aids in understanding complex patterns and anomalies, supporting decision-making processes and explainability.

3.1.2 Machine Learning Techniques for Anomaly Detection

The application of machine learning techniques has significantly transformed anomaly detection, allowing automated model construction based on available training data. This approach is driven by the accessibility and ease of acquiring training data compared to manual model definition, particularly with the increasing complexity and diversity of anomalies. Table 3.1 presents the comparison of machine learning methods for anomaly detection.

Supervised Anomaly Detection

Supervised learning methods [Tiw22] rely on labeled training sets containing both normal and anomalous samples to construct predictive models. These models undergo extensive evaluation, considering metrics like precision, recall, and F1-score to ensure accurate anomaly identification. A substantial challenge lies in acquiring labeled data, especially for rare or novel anomalies, which might require sophisticated data collection or synthesis techniques.

- **Support Vector Machines (SVM):** SVMs [Hos+21; Ma+21] are effective classifiers that aim to find the hyperplane that best separates classes by maximizing the space between them. They can handle both straight and curvy separations using different kernels. Despite their efficacy, SVMs might encounter computational challenges with larger datasets, and tuning their hyperparameters, such as the choice of the kernel function and the regularization parameter, could be crucial for their performance. They are particularly effective when the boundary between normal and anomalous instances is well-defined and separable.

- **Decision Trees:** Decision Trees [Var+21; Dou+23] partition the data by recursively splitting it based on feature attributes. They are highly interpretable, enabling easy visualization of decision rules. However, they tend to overfit when the trees grow too deep, and they might struggle to capture complex relationships in the data.

- **Random Forest:** Random Forest [PT17; BS21] is an ensemble learning method consisting of multiple decision trees. It is resilient to overfitting, works well with noisy data and anomalies, and provides an estimation of feature importance. Random Forests can handle high-dimensional data but may become computationally expensive with a large number of trees in the forest. They stand out in giving strong and reliable classifications by combining decisions from multiple trees.

- **K-Nearest Neighbors (K-NN):** K-NN [Wan+20a; Yin+21] classifies data points based on the majority class among their K-nearest neighbors. It is adaptable to non-standard data types like text or images. However, K-NN's performance heavily depends on the choice of the distance metric and the value of K. Moreover, it can be computationally expensive, particularly with larger datasets, as it requires calculating distances between the query point and all training points.

- **Logistic Regression:** Logistic Regression [Nou+19; Pal19] models the probability of a binary outcome based on predictor variables. It is interpretable, computationally efficient, and provides insights into the influence of input features on the output. However, Logistic Regression assumes a linear relationship between features and outcomes, and it might struggle with non-linear patterns in the data.

Unsupervised Anomaly Detection

Unsupervised techniques operate without training labeled data, relying on core assumptions about statistical differences between normal and abnormal instances. These methods are crucial when labeled data is scarce or when dealing with novel or evolving anomalies.

- **K-Means Clustering:** K-Means [GMC20; Gad+22] partitions data into K clusters based on similarity measures, aiming to minimize intra-cluster distances. It is efficient and works well with large datasets. However, it is sensitive to the initial placement of centroids, struggles with non-spherical clusters, and requires prior knowledge of the number of clusters (K).

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** DBSCAN [Pu+20; Wib+21] identifies clusters based on density in the data space, distinguishing between core points, border points, and noise. It is capable of identifying arbitrarily shaped clusters, and does not require specifying the number of clusters beforehand. However, setting appropriate parameters, such as epsilon and minimum points, can be challenging.

- **Isolation Forest:** Isolation Forest [Les+21; Xu+23] isolates anomalies by randomly partitioning the data space and identifying anomalies in fewer partitions. It is efficient for large datasets and does not assume any underlying data distribution. However, it might struggle with multi-modal data and is not effective in identifying anomalies close to the normal instances.

- **One-Class Support Vector Machines (OCSVM):** OCSVM [QWJ21; LHH23] aims to separate normal instances from anomalies in a hyperspace. It is suitable for anomaly detection when only normal data is available for training. However, determining the appropriate kernel and setting

hyperparameters can be challenging, and OCSVM might struggle with high-dimensional data.

- **Gaussian Mixture Models (GMM):** GMM [Cho+23; JD23] represents the data distribution as a combination of Gaussian distributions. It is flexible in representing complex data distributions but might struggle with high-dimensional data and requires setting the number of components. GMMs are effective when the data exhibits mixed or overlapping clusters.

Deep Learning Models

Deep learning models have emerged as powerful tools that are applicable in anomaly detection area [LJ23], offering both supervised and unsupervised approaches to address the complexities of detecting anomalies in diverse datasets. In supervised scenarios, architectures like Artificial Neural Networks (ANNs) are common. ANNs [Red+21; Alb+22] are versatile models capable of learning complex patterns from data through interconnected layers of nodes. They are highly adaptive and can approximate non-linear functions. However, they require a large amount of data for training and careful tuning of numerous hyperparameters, such as the number of layers, neurons per layer, and learning rates. Overfitting is a common challenge with ANNs, especially in smaller datasets.

Unsupervised deep learning techniques, such as Autoencoders, provide an alternative way for anomaly detection, particularly when labeled data is limited or unavailable. Autoencoders [TMG23; Yun+23] are neural networks designed to reconstruct input data. They learn a compressed representation of the data and are effective in capturing complex patterns. However, training autoencoders requires careful tuning of architecture and regularization techniques to prevent overfitting and ensure effective representation learning.

3.2 Navigating Anomaly Detection in Edge Clouds

Anomaly detection plays a key role in ensuring the smooth operation of edge cloud environments. These distributed systems, characterized by decentralized architecture and close proximity to data sources and end-users, face a range of potential anomalies that could disrupt their seamless operations. These anomalies arise from diverse sources, including security threats, hardware malfunctions, unusual system behaviors, and performance issues [FBE23a]. The timely identification and resolution of anomalies are crucial steps to prevent problems, reduce system downtime, and strengthen the resilience of edge-based systems. This section explains various types of anomalies that may occur in edge cloud environments followed by a detailed explanation of considerations for anomaly detection methods specifically designed for edge clouds. Understanding the diverse anomalies that can arise in these environments is crucial for developing effective detection strategies. Then, specific considerations that

Table 3.1: Comparison of machine learning-based anomaly detection methods.

Algorithm	**Advantages**	**Disadvantages**	**Considerations**
SVM [Hos+21; Ma+21]	Handles non-linear data, effective in high-dimensional spaces	Sensitive to the choice of kernel, memory-intensive	Selecting appropriate kernel, memory constraints
Decision Trees [Var+21; Dou+23]	Easy to interpret, handle missing values	Prone to overfitting, may create complex trees	Pruning techniques, ensemble methods
Random Forest [PT17; BS21]	Robust against overfitting, handles high-dimensional data	Slower training, may not be easily interpretable	Tuning hyperparameters, feature importance
K-NN [Wan+20a; Yin+21]	Simple, adaptable to different data distributions	Computationally expensive with large datasets	Choosing optimal 'K', handling high dimensions
Logistic Regression [Nou+19; Pal19]	Simple, interpretable, works well with small datasets	Sensitive to outliers, assumes linear relationships	Regularization techniques, handling multicollinearity
ANN [Red+21; Alb+22]	Effective for complex data patterns, feature learning	Computationally expensive, requires large datasets	Hyperparameter tuning, model architecture
K-Means [GMC20; Gad+22]	Scales well to large datasets, simple	Sensitive to initial centroids, requires prior knowledge of clusters	Initialization methods, determining optimal clusters
DBSCAN [Pu+20; Wib+21]	Handles arbitrary shaped clusters, robust to outliers	Difficulty in finding appropriate parameters	Parameter tuning, understanding epsilon and minimum points
Isolation Forest [Les+21; Xu+23]	Efficient for large datasets, scales well	Sensitive to hyperparameters, may underperform on small datasets	Tuning contamination factor, ensemble size
OCSVM [QWJ21; LHH23]	Effective for novelty detection, handles high-dimensional data	Sensitive to kernel choice, parameter tuning	Selecting kernel, tuning nu parameter
Autoencoders [TMG23; Yun+23]	Effective in learning non-linear patterns, robust to noise	Complex architecture, challenging to interpret	Hyperparameter tuning, handling architecture
GMM [Cho+23; JD23]	Flexible clustering, works with any distribution	Sensitive to initialization, prone to local optima	Initialization methods, understanding components

come into play when deploying anomaly detection mechanisms in edge clouds are explored.

3.2.1 Importance of Anomaly Detection in Edge Clouds

Anomaly detection constitutes the cornerstone of ensuring the reliability, security, and operational integrity of infrastructure, applications, and the overall ecosystem within edge cloud environments, given their dynamic and distributed nature [FBE23a]. The identification and resolution of anomalies become imperative in such environments, stemming from diverse sources like malicious attacks, unexpected system behaviors, or hardware malfunctions. Timely detection and mitigation of anomalies are vital to decrease potential disruptions, minimize downtime, and enhance the robustness of edge-based systems. Neglecting the significance of anomaly detection within edge clouds could result in substantial operational setbacks and system instabilities, compromising the efficiency, functionality, reliability, and performance of crucial applications operating at the edge, such as those in IoT networks and autonomous systems.

Furthermore, as edge cloud environments evolve and become more essential to industries, the complexities and sophistication of potential anomalies also increase. This necessitates continual improvements in anomaly detection methodologies and tools. Adaptability of detection systems are paramount to effectively address emerging threats and anomalies in real-time [FBE23a]. Additionally, the collaborative and federated nature of edge cloud environments presents unique challenges in anomaly detection, where data sources are dispersed across various nodes and devices. This necessitates the development of decentralized anomaly detection models capable of processing and analyzing data at the network's edge. As such, the ability to identify and respond to anomalies within this distributed framework becomes vital for maintaining the integrity and operational continuity of edge-based systems.

3.2.2 Types of Anomalies in Edge Clouds

There are several types of anomalies in edge cloud environments, and it's essential to recognize that they may be related, with one type potentially resulting in another. Among these anomalies, certain ones hold particular significance due to their potential impact on the operational stability of edge clouds. The most crucial anomalies include:

- **Security Threats:** Among the most prevalent threats to edge cloud environments are security threats, ranging from Distributed Denial of Service (DDoS) attacks to malware injections. A particularly significant threat is the Very Short Intermittent DDoS (VSI-DDoS) attack [FBE21; FBE22]. This type of low-rate DDoS attack exhibits a specific behavior illustrated in Figure 3.2. Unlike conventional DDoS attacks, VSI-DDoS involves several highly-synchronized attacker nodes (bots), denoted as $n_1, n_2, ..., n_k$, sending bursts of requests to the server within a very short

time frame (a few milliseconds), denoted α. Subsequently, the attackers remain idle for a small interval (a few seconds), represented by Δ, and iterate this process to reduce users' Quality of Service (QoS). Consequently, the rate of packet drop related to legitimate users increases. TCP interprets this packet drop as server-side congestion, triggering its congestion control mechanism to retransmit lost packets at a slower rate. This leads to an increased average response time for legitimate users, ultimately resulting in the degradation of QoS, which aligns with the attackers' main goal. Notably, the monitored metrics of the system during a VSI-DDoS attack remain almost similar to normal periods. Such malicious activities pose severe risks to the availability, reliability, and security of edge computing systems. Anomaly detection stands as the frontline defense against such attacks, empowering proactive measures to promptly detect and mitigate these threats.

- **Performance Issues:** Identifying and addressing performance anomalies in edge clouds is essential for ensuring optimal system responsiveness, minimizing disruptions, and strengthen the overall operational stability of these decentralized computing architectures. Furthermore, unusual system behaviors arising from diverse factors such as software bugs or unexpected user interactions, can trigger anomalies in edge cloud systems [RK22]. For instance, software glitches or coding errors may lead to abnormal system behaviors, causing disruptions in data processing or task execution. Unexpected user interactions, such as abnormal input patterns or unauthorized access attempts, can also contribute to anomalous activities within edge cloud environments. Detecting and addressing such anomalies are crucial to maintaining system integrity and ensuring seamless operations.

- **Hardware Malfunctions:** This type of anomalies present a critical challenge in edge cloud environments. Nodes might experience failures, resource constraints, or performance degradation due to hardware issues [Erh+21]. For example nodes within these environments may encounter failures such as sudden power outages, memory module malfunctions, or disk drive failures. Anomaly detection mechanisms are essential in quick identification of these issues, enabling proactive actions such as resource reallocation, to mitigate potential system instabilities. Moreover, performance anomaly issues on edge clouds can significantly impact the efficiency and reliability of these environments. The dynamic nature of edge clouds, with decentralized computing resources distributed across various nodes, introduces challenges in maintaining consistent performance. Fluctuations in network conditions, varying workloads, and resource constraints can lead to anomalies, causing delays, latency, or sub-optimal execution of tasks.

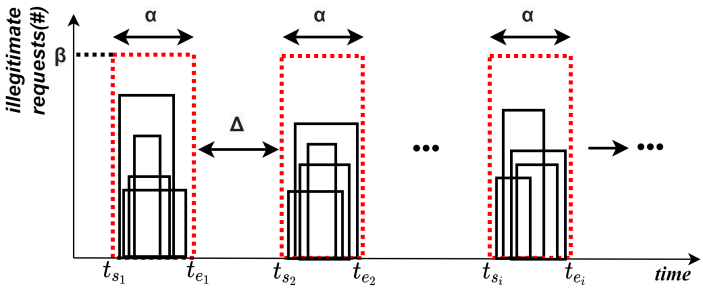

Figure 3.2: Characteristics of a VSI-DDoS attack [FBE22]. The parameter α represents the burst interval, β indicates the number of requests, and Δ stands for an idle interval. Additionally, t_{s_i} and t_{e_i} denote the start and end of the i^{th} burst, respectively.

3.2.3 Anomaly Detection Challenges in Edge Clouds

The dynamic nature of edge cloud environments, marked by their distributed architecture and diverse applications, poses challenges for anomaly detection mechanisms [FBE23a]. These mechanisms need to adapt to the constantly evolving environment, accommodating diverse data sources, heterogeneous devices, and variable workloads. Anomaly detection in edge cloud environments presents a number of challenges, as shown in Table 3.2 rooted in the unique characteristics and constraints inherent in these decentralized systems. One primary challenge revolves around the inherent diversity of edge devices and the heterogeneity of data generated by these devices [Car+21]. Edge networks encompass a wide range of devices with varying computational capabilities, communication protocols, and data formats. This heterogeneity complicates the design and deployment of anomaly detection algorithms, demanding adaptability and the ability to handle diverse data types and processing capabilities.

Moreover, the limited resources of edge devices pose a significant challenge for anomaly detection. These devices often operate with restricted computational power, memory, and energy resources [PM17]. Implementing complex anomaly detection algorithms on such resource-constrained devices becomes a difficult task, requiring lightweight and efficient algorithms that balance accuracy and computational demand. Achieving a balance between detection accuracy and resource utilization is crucial in edge environments where optimizing resource consumption is essential.

Another critical consideration for anomaly detection in edge clouds is the dynamic and evolving nature of edge environments [Shi+16]. These networks undergo frequent changes in device connectivity, mobility, and network topology, introducing inherent instability and network fluctuations. Such dynamic changes pose challenges in adapting anomaly detection mechanisms to account for these dynamic variations and ensuring their robustness against transient

Table 3.2: Challenges for anomaly detection in edge cloud environments.

Challenges	Description	Addressing Strategies
Data Heterogeneity	Diverse devices, varied data formats	Adaptable algorithms, data standardization
Resource Constraints	Limited resources, computational power	Lightweight algorithms
Dynamic Network Behavior	Fluctuations in network topology	Continuous learning models, dynamic thresholding
Data Management	Large volume, efficient processing	Edge-based preprocessing, localized detection

network behaviors. Furthermore, the distributed and decentralized nature of edge environments poses data management challenges. Data generated at the edge needs to be efficiently aggregated, processed, and transmitted for anomaly detection. However, due to the high volume of data generated by edge devices and the distributed nature of edge networks, transmitting all raw data to centralized servers for analysis is impractical and causes significant communication overhead. Implementing efficient data preprocessing, feature extraction, and aggregation techniques at the edge is crucial to reduce data transmission and enable localized anomaly detection without compromising accuracy.

In conclusion, anomaly detection in edge cloud environments presents multifaceted challenges, including data heterogeneity, resource constraints, dynamic network behavior, and efficient data management. Addressing these challenges requires specific solutions that account for the unique characteristics of edge environments while ensuring effective anomaly detection without overwhelming resource limitations. Overcoming these challenges is vital in designing robust and efficient anomaly detection systems capable of enhancing the reliability and security of edge cloud infrastructures.

3.3 Anomaly Detection Methods for Edge Clouds

As it is mentioned before, one of the main aspect of securing edge cloud environments involves the deployment of anomaly detection methods. This section provides an exploration of advanced recent techniques specifically designed for anomaly detection within edge clouds. Each approach is carefully examined, highlighting its unique strengths and applications in addressing the dynamic challenges posed by edge cloud architectures.

- **LSTM with Attention Layer Approach:** This approach employs a sequence modeling approach to address the VSI-DDoS detection problem [FBE21]. The LSTM network is utilized to learn from historical occurrences before each instance, and sliding window features are exploited to capture patterns related to VSI-DDoS attacks. Additionally, a local attention layer is introduced to enhance the model's ability to discern patterns occurring intermittently in very short intervals during VSI-DDoS

attacks. The advantages and disadvantages [FBE21] of this approach are as follows:

Advantages:

- *Effective Sequence Modeling*: The LSTM with attention layer acts effectively in capturing temporal dependencies and patterns in sequential data, making it suitable for VSI-DDoS detection challenges.
- *Improved Detection Accuracy*: This method demonstrates superior detection accuracy compared to state-of-the-art methods, showcasing its effectiveness in identifying VSI-DDoS attacks in edge cloud environments.

Disadvantages:

- *Increased Training and Testing Time*: The addition of the attention layer introduces more parameters to optimize, leading to a slight increase in training and testing time compared to standalone LSTM models. This is due to the added complexity of optimizing the attention mechanism.
- *Parameter Optimization Overhead*: The attention layer increases the complexity of the overall model, requiring careful parameter tuning.

- **Deep Ensemble of Sequential Models Approach:** This approach addresses the challenge of VSI-DDoS attacks targeting different levels of the edge cloud system simultaneously, making the attacks harder to detect [FBE22]. This method utilizes a combined deep and ensemble learning approach along with a novel training algorithm. Additionally, it leverages a novel chunking algorithm that enhances model performance by considering overlapped chunks based on an overlap ratio. The advantages and disadvantages [FBE22] of this approach are as follows:

  Advantages:

  - *Cross-Layer Detection*: This method is able to detect VSI-DDoS attacks across different levels of the edge cloud system, providing a comprehensive detection mechanism against cross-layer VSI-DDoS attacks.
  - *Superior Performance*: This method outperforms state-of-the-art solo and ensemble baseline models, demonstrating its effectiveness in achieving high detection accuracy for cross-layer VSI-DDoS attacks.
  - *Time Efficiency*: Time analysis reveals that DELA is less time-intensive compared to ensemble baseline models, offering an efficient solution for timely VSI-DDoS detection in edge cloud environments.

  Disadvantages:

- *Complexity and Training Algorithm Overhead*: The combined deep and ensemble learning approach, along with the novel training algorithm, introduces additional complexity to this method. While this complexity enhances performance, it requires careful consideration during the training phase, potentially increasing overhead.
- *Dependency on Chunking Algorithm*: The effectiveness of DELA is dependent on the proposed chunking algorithm. If not properly tuned, the model's performance may be sensitive to variations in the overlap ratio, requiring careful parameter optimization.

- **Reinforced Transformer Learning Approach:** Reinforced transformer learning-based approach [Bhu+22] is applicable for detecting VSI-DDoS attacks and degradation of users' Quality of Service and experience in edge clouds. The integration of transformer and deep reinforcement learning enhances the model's effectiveness by using an encoding layer for compact feature representation of raw data. The advantages and disadvantages [Bhu+22] of this approach are as follows:

  Advantages:

  - *Dynamic Attack Behavior*: This approach adopts dynamic attack behavior, allowing it to adapt to evolving attack patterns and improve detection accuracy over time.
  - *Learning Stability*: Leveraging deep reinforcement learning, the model learns stability in decision-making, contributing to consistent and reliable performance.
  - *Multihead Attention for Context Analysis*: The multihead attention mechanism of transformer-based models facilitates contextual information analysis in time-series data, enhancing the model's attack detection capability.

  Disadvantages:

  - *Increased Model Complexity*: The integration of transformer and deep reinforcement learning introduces increased model complexity, which may pose challenges in terms of interpretability.
  - *Dependency on Dynamic Attack Behavior*: While dynamic attack behavior is an advantage, it also introduces a level of dependency, requiring continuous optimization to adapt to evolving attack strategies.
  - *Resource Intensive Training:* The model's training process may be resource-intensive due to the complexity of transformer-based architectures, potentially demanding substantial computational resources.

- **Robust Meta-reinforced Learning Approach:** Robust meta-reinforced learning approach [VMB22] is specifically designed to detect VSI-DDoS

attacks in Edge Clouds. The primary goal of this approach is to address the robustness issue observed in many learning approaches when detecting VSI-DDoS attacks, where these models may exhibit sub-optimal performance when deployed in environments different from their training environment, a challenge known as the covariate shift problem. The advantages and disadvantages [VMB22] of this approach are as follows:

Advantages:

- *Robust Across Different Environments*: This approach demonstrates robustness across diverse settings, such as various scenarios and environments. This ability to stably detect abnormal patterns in different environments contributes to its effectiveness in real-world deployment scenarios.
- *Stable Performance in Online and Offline Evaluations*: This approach exhibits stable performance in both online and offline evaluations, ensuring consistent and reliable detection of VSI-DDoS attacks.
- *Addressing Covariate Shift Problem*: This approach effectively addresses the covariate shift problem, making it suitable for deployment in environments beyond its training domain.

Disadvantages:

- *Model Complexity*: The meta-reinforced learning approach, while providing robustness, may introduce increased model complexity, potentially impacting interpretability and resource requirements.
- *Training Overhead*: The model may cause training overhead to adapt to different environments, requiring sufficient computational resources for effective deployment.
- *Dependency on Historical Data*: This approach's stability and evolution over time depend on the availability of diverse and representative historical data for training.

- **Hybrid Sequential Probabilistic Graphical Model Approach:** Due to the inherent characteristics of edge cloud resources, susceptibility to both performance and security anomalies is prevalent. Identifying the types of anomalies becomes crucial for effective mitigation. This approach [FBE23b] utilizes a hybrid sequential Probabilistic Graphical Model (PGM) incorporating GRU/LSTM layers and Conditional Random Field (CRF) to address unified detection of both security threats and performance issues in edge clouds. The advantages and disadvantages [FBE23a] of this approach are as follows:

  Advantages:

- *Unified Detection of Anomalies*: This approach has superior performance in unified identification of security and performance anomalies, offering a comprehensive approach to anomaly detection in edge cloud environments.
- *Historical Information Utilization*: This approach leverages the historical information in data by utilizing GRU/LSTM as initial layers, enabling better understanding and contextualization of anomalies.
- *Relationship Extraction with CRF*: The use of CRF as the model's final layer allows for the extraction of relationships between former predictions, enhancing the decision-making process for anomaly identification.

Disadvantages:

- *Model Complexity*: The hybrid model introduces added complexity, requiring careful consideration of computational resources for practical deployment.
- *Training Overhead*: Training the model, which is an offline task, may involve overhead due to the sequential and probabilistic components, demanding computational resources and time.

In conclusion, this section provides an in-depth exploration of anomaly detection approaches tailored for the edge cloud environments. The highlighted methods showcase diverse strengths in addressing the challenges posed by edge cloud architectures. The methods discussed here provide good insights into the evolving area of anomaly detection, contributing to the ongoing effort to strengthen the resilience of edge cloud environments.

3.4 Anomaly Resolution Strategies for Edge Clouds

Following the detection of anomalies within edge clouds, implementing effective resolution strategies is crucial to restore normal situation, mitigate the impact of anomalies, and ensure the smooth operation of edge cloud infrastructures. The section outlines several common resolution strategies:

- *Reactive anomaly handling:* This strategy focuses on an immediate response to anomalies by triggering automated actions or alerting system administrators [ERM18]. For instance, when anomalies are detected, affected components or services can be promptly restarted or reconfigured to restore normal operation.
- *Resource reallocation:* Anomalies in edge cloud environments may indicate resource imbalances or bottlenecks. In such cases, resource reallocation strategies [Li+19] can be applied to optimize resource utilization and mitigate the impact of anomalies. This may involve redistributing computing

resources, storage capacity, or network bandwidth based on the detected anomalies.

- *Dynamic workload adjustment:* Anomalies may arise due to unexpected spikes or fluctuations in workload. Dynamic workload adjustment strategies [SKT21] can automatically scale resources up or down to match the demand. This can involve horizontal scaling by adding or removing edge cloud instances or vertical scaling by adjusting the resource allocation of existing instances.

- *Fault tolerance and redundancy:* Anomalies can sometimes result from failures or disruptions in the edge cloud infrastructure. Employing fault tolerance mechanisms [JSW17], such as data replication, load balancing, or backup systems, can help mitigate the impact of anomalies and ensure the high availability of services.

- *Automated recovery and healing:* Anomalies can trigger automated recovery and healing mechanisms [Li+21] to restore the system to a normal state. These mechanisms may include configuration management tools that automatically correct misconfigurations or repair faulty components.

- *Predictive analytics and proactive measures:* To anticipate and prevent future anomalies, predictive analytics techniques [Ban21] can be employed. By analyzing historical data and patterns, proactive measures can be taken, such as predictive resource allocation, anomaly forecasting, or preventive maintenance, to minimize the occurrence and impact of anomalies.

- *Security measures:* Anomalies can also be indicators of security breaches or attacks on the edge cloud infrastructure. In such cases, security measures [Cop+17] such as intrusion detection systems, firewalls, access controls, or encryption mechanisms can be implemented to mitigate the security risks associated with the anomalies.

In conclusion, the selection and combination of resolution strategies will depend on the specific nature of the detected anomalies, the edge cloud environment, and the desired system requirements.

Chapter 4

Evaluation Strategies

This chapter focuses on evaluation of anomaly detection methods in edge cloud environments. Initially, it discusses the evaluation metrics used to measure the performance of proposed anomaly detection systems, utilizing various datasets for a thorough evaluation. Subsequently, it provides detailed insights into the experimental testbed setup, explaining the environment where the proposed methods undergo testbed testing.

4.1 Evaluation Metrics and Datasets

To assess the performance of anomaly detection methods in edge cloud environments, this section discusses commonly used evaluation metrics for edge cloud anomaly detection and highlights relevant datasets for benchmarking and validation purposes.

4.1.1 Evaluation Metrics

Evaluation metrics provide quantitative measures to assess the performance of anomaly detection methods. In the context of edge clouds, these metrics serve as essential tools for evaluating the effectiveness and efficiency of anomaly detection techniques. Commonly used metrics include:

- **Detection Accuracy**[1]**:** Measures an algorithm's ability to correctly identify anomalies. It is computed as the ratio of correctly detected anomalies to the total number of anomalies in the dataset:

$$\text{Accuracy} = \frac{\text{Number of correctly detected anomalies}}{\text{Total number of anomalies in the dataset}}$$

  While accuracy offers objective evaluation and aids in performance benchmarking, it might be impacted by imbalanced datasets, where anomalies are scarce.

[1] https://en.wikipedia.org/wiki/Accuracy_and_precision

- **False Positive Rate**[2]**:** Evaluates an algorithm's tendency to falsely label normal instances as anomalies. It is crucial in scenarios where false alarms can be costly or disruptive. The False Positive Rate (FPR) is calculated as:

$$\text{FPR} = \frac{\text{Number of false positive instances}}{\text{Total number of actual negative instances}}$$

  Lower false positive rates are generally desired in anomaly detection systems.

- **Precision**[3]**:** Measures the accuracy of positive predictions, indicating the proportion of correctly identified anomalies out of all detected anomalies. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

  Precision highlights the system's ability to avoid false alarms by computing the ratio of true positives to the total predicted positives.

- **Recall**[4] **(Sensitivity):** Evaluates the system's ability to detect anomalies correctly, measuring the ratio of true positives to the total number of actual positives. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

  Recall provides insights into the system's sensitivity in identifying all anomalies in the dataset.

- **F1-Score**[5]**:** Represents the harmonic mean of precision and recall, balancing the trade-off between these metrics. It is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

  F1-Score provides a comprehensive understanding of the anomaly detection system's performance by considering both false positives and false negatives.

- **Area Under the Receiver Operating Characteristic Curve**[6] **(AUC-ROC):** In supervised learning-based anomaly detection, various additional metrics contribute to assessing model performance. These include the confusion matrix, AUC-ROC, and precision-recall curve. The AUC-ROC metric evaluates the performance of a binary classification model across

[2]https://en.wikipedia.org/wiki/False_positive_rate
[3]https://en.wikipedia.org/wiki/Precision_and_recall
[4]https://en.wikipedia.org/wiki/Precision_and_recall
[5]https://en.wikipedia.org/wiki/F-score
[6]https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

different discrimination thresholds. It quantifies the model's ability to distinguish between normal and anomalous instances, illustrating the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity). A higher AUC-ROC value (closer to 1) indicates superior discrimination capability and better separation of classes. This metric provides a comprehensive view of the model's classification performance and is particularly useful for evaluating anomaly detection models.

- **Silhouette Score**[7]**:** Measures the cohesion and separation between clusters by computing the mean intra-cluster distance and the mean nearest-cluster distance for each sample. It ranges from -1 to 1; higher values indicate better-defined clusters. A score close to 1 suggests that samples are well-clustered, while values near -1 indicate incorrect clustering. The Silhouette Score formula is given by:

$$\text{Silhouette Score} = \frac{1}{N}\sum_{i=1}^{N}\frac{b_i - a_i}{\max(a_i, b_i)}$$

  Where N is the number of samples, a_i is the mean intra-cluster distance for sample i, b_i is the mean nearest-cluster distance for sample i.

- **Davies-Bouldin Index**[8] **(DBI):** Evaluates the clustering quality by computing the average similarity between each cluster and its most similar cluster, considering both intra-cluster and inter-cluster distances. Lower values of this index (close to zero) indicate better clustering; a smaller index denotes more distinct clusters. The DBI is given by:

$$\text{DBI} = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq i}\left(\frac{\text{avg\_intra\_distance}_i + \text{avg\_intra\_distance}_j}{\text{inter\_distance}_{ij}}\right)$$

- **Adjusted Mutual Information**[9] **(AMI):** Measures the agreement between two clusterings while considering the probability of chance agreement. Higher AMI values denote better agreement between two clusterings, ranging from 0 (no agreement) to 1 (perfect agreement).

  The Adjusted Mutual Information is calculated using the following formula:

$$AMI = \frac{MI - E(MI)}{\max(H(U), H(V)) - E(MI)}$$

  Where, MI is the Mutual Information, E(MI) stands for the Expected Mutual Information under independence, and H(U) and H(V) represent the entropies of the two clustering sets.

[7] https://en.wikipedia.org/wiki/Silhouette_(clustering)
[8] https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index
[9] https://en.wikipedia.org/wiki/Adjusted_mutual_information

- **Detection and Training Time**[10]**:** Detection time refers to the duration taken by an algorithm to identify anomalies in a dataset, while training time represents the time required to train the anomaly detection model. These time-based metrics, specially the time of detection which is an online task, are crucial in real-time applications, where timely detection is essential.

These evaluation metrics provide a multifaceted analysis of anomaly detection systems in edge cloud environments, helping in comparative assessments and algorithm selection.

4.1.2 Datasets

In evaluating anomaly detection methods for edge cloud environments, the presence of datasets reflecting the complexities of these settings is crucial. The existing datasets for evaluation are categorized in two different types as follows:

- **Testbed Datasets:** Although there are scarcity of available datasets specifically collected for edge cloud anomaly detection, some works have initiated the creation of practical datasets for evaluating these techniques [FBE21; FBE22]. For instance, Forough et al. [FBE21] collected an edge clouds dataset focused on the challenges of application-layer Very Short Intermittent DDoS (VSI-DDoS) attacks, which represent a relatively recent type of low-rate DDoS attacks. Additionally, in another work, Forough et al. [FBE22] collected a comprehensive dataset highlighting cross-layer VSI-DDoS attacks specifically targeting edge clouds. These datasets present a comprehensive view, monitoring diverse infrastructure layers including application, virtualization, and physical layers. The datasets encapsulate information like network traffic patterns, the status of deployed applications, and metrics related to the user's Quality of Service (QoS) in testbed edge cloud deployments.

- **Benchmark Datasets:** Beyond datasets tailored explicitly for edge cloud environments, there are also several existing datasets from related domains [MS15; Tav+09; Sha+19; Shi+12; CKK21], such as network intrusion detection or cybersecurity. Although these datasets may not be spesificly collected for edge cloud scenarios, they share commonalities like the presence of anomalies and the imperative for real-time anomaly detection. However, it is crucial to carefully consider potential domain shifts and differences in data distributions when adapting these datasets for anomaly detection in edge clouds.

Table 4.1 provides comprehensive details regarding various datasets commonly employed for evaluating anomaly detection methods in edge cloud envi-

[10]https://www.xilinx.com/applications/ai-inference/difference-between-deep-learning-training-and-inference.html

ronments. This information includes critical dataset aspects such as the number of instances, anomalies, types of anomalies present, and the number of features.

4.2 Testbed Setup

This section details the creation of a testbed setup in general, as illustrated in Figure 4.1, for anomaly detection on edge clouds using various components and technologies.

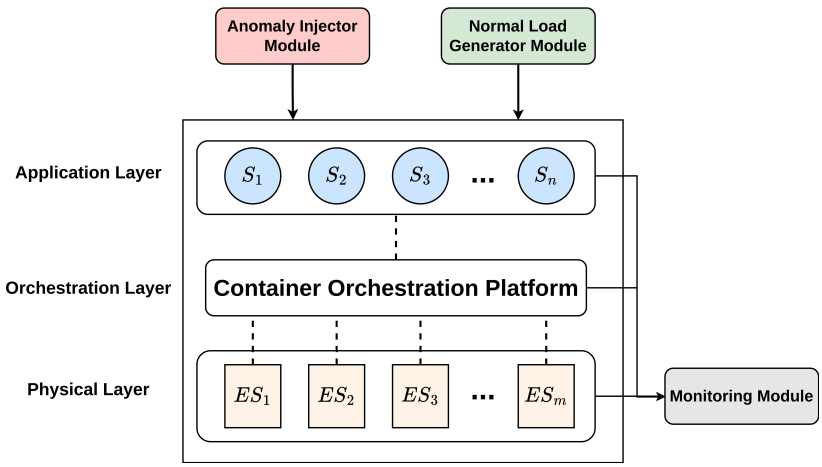

Figure 4.1: A general testbed setup for anomaly detection on edge clouds. S_i stands for i^{th} Service withing the benchmark microservices application, and ES_i stands for i^{th} edge server.

4.2.1 Container Orchestration Platform

Container orchestration platforms are essential in the deployment and management of containerized microservice applications, which is important in development of testbed setup for anomaly detection within edge cloud environments. Among the most widely used platforms, Kubernetes and Docker Swarm stand out as common tools facilitating efficient container management.

Kubernetes[11]**:** Known for its advanced capabilities in automating the deployment, scaling, and management of containerized applications, Kubernetes provides a sophisticated platform for deploying microservices applications in edge clouds. Its architecture allows for seamless allocation and management of containers across different edge nodes. Kubernetes follows a master-worker configuration, where the master node supervises cluster operations, and multiple worker nodes carry out assigned tasks. This distributed structure ensures high

[11] https://kubernetes.io/

Table 4.1: A comparative analysis of edge cloud-specific datasets and benchmarks from related domains.

Datasets	Instances (#*)	Anomalies (#*)	Types of Anomalies	Features (#*)
VSI-DDoS [FBE21] (application layer)	11,452	2,428	VSI-DDoS on layer 7	28
VSI-DDoS (cross layer) [FBE22]	40,000	20,000	VSI-DDoS on layer 4 and 7	60
NSL-KDD [MS15]	148,517	71,463	DoS - Prob - U2R - R2L	41
UNSW-NB15 [Tav+09]	2,540,044	56,215	Fuzzers - Analysis - Backdoors - DoS - Exploits - Generic - Reconnaissance - Shellcode - Worms	49
CIC-DDoS2019 [Sha+19]	16,491,642	85,039	PortMap - NetBIOS - LDAP - MSSQL - UDP - UDP Lag - SYN - WebDDoS - DNS - TFTP	88
ISCX-IDS [Shi+12]	2,454,116	69,424	DoS - DDoS - Probe - U2R - Brute Force	29
AWID [CKK21]	9,192,453	208,368	DoS - Deauthentication - Spoofing - MitM - Eavesdropping	82

*Number of.

availability, fault tolerance, and scalability, aligning well with the dynamic nature of edge environments.

Docker Swarm[12]**:** As an integrated clustering and scheduling tool, Docker Swarm provides a user-friendly interface for deploying containerized applications across edge nodes. Unlike Kubernetes, Docker Swarm follows a simpler setup, leveraging the Docker engine for container orchestration. Its simplicity in configuration and management makes it an interesting option for smaller edge cloud setups.

Role in Testbed Setup: Container orchestration platforms like Kubernetes and Docker Swarm hold significant importance in deploying benchmark web-based microservice applications. These platforms are instrumental not only in creating testbeds for evaluating anomaly detection methods but also in deploying and managing benchmark microservice architectures. By reducing the complexities related to deployment, scalability, and resource management, Kubernetes and Docker Swarm enable the smooth deployment of web-based microservices designed for benchmarking anomaly detection methods.

These platforms offer features like service discovery, load balancing, and self-healing mechanisms that are essential in maintaining the reliability and scalability of benchmarking environments. Especially in the context of anomaly detection, deploying benchmark web-based microservices aids in thorough testing and validation of anomaly detection methodologies. The inherent capabilities of container orchestration platforms facilitate the creation of realistic test environments that accurately reflect edge cloud scenarios, ensuring comprehensive evaluations of anomaly detection methods and enabling efficient comparisons between different detection techniques.

4.2.2 Benchmarking Microservice Applications

Benchmarking microservice applications serves as a critical aspect of evaluating performance, identifying improvement areas, and validating anomaly detection methods. This section focuses on open-source, containerized microservice benchmarks specifically desigend for web-serving use cases, comprising a number of microservices. These benchmark tools were primarily developed to offer hands-on experience with cloud-native platforms, typically showcasing computationally straightforward applications, often linked to e-commerce scenarios.

Examples of such benchmarks include TrainTicket [Zho+18; Zho+22], Sock-Shop [Wea22], OnlineBoutique [Goo23], and DeathStarBenchHotelReservation [Gan+19; Gan22]. Some benchmarks possess basic topologies with a few microservices like Bookinfo, CloudSuite [Fer+12; PSF16; Fer+24], TeaStore [Von+18; Uni24], JPetStore [JA19], PetClinic [Spr24], AcmeAir [24a], Spring-CloudDemo [24d], and BiFrost [24b].

Several benchmarks enable experimentation with various architectures, such as DeathStarBench, µSuite [SW18; 24c], and CloudSuite. Notably, TrainTicket

[12]https://docs.docker.com/engine/swarm/

and DeathStarBench allow for the evaluation of performance impact on a larger scale. However, it is important to note that these benchmarks are built with fixed architectural designs, causing challenges for customization.

Moreover, a recent addition to existing tools is the open-source HydraGen [Sal+23], a benchmark generator specifically designed for microservices. HydraGen facilitates the creation of customizable microservice-based applications tailored for web-serving use cases. It can be used to conduct comprehensive experimental evaluations, assessing factors like application topologies, computational and inter-service complexities that impact on cloud-native resource management mechanisms. This tool enables in-depth investigations into optimizing and enhancing cloud-native systems.

4.2.3 Anomaly Injection and Normal Load Generator

To evaluate the anomaly detection mechanisms within the testbed, an Anomaly Injector and Normal Load Generator modules are introduced. This part outlines the tools that can be employed for both anomaly injection and normal load generation as follows:

Anomaly Injection Tools:

- *Apache Benchmark*[13] (*ab*): Apache Benchmark is a versatile command-line tool designed for benchmarking and stress testing web servers. It can be utilized to inject anomalies by simulating various types of HTTP requests and analyzing the server's response under different load conditions.
- *stress-ng*[14]: stress-ng is a stress-testing tool that systematically exercises different components of a system. It can be employed for anomaly injection by introducing stress on the CPU, memory, I/O, and other system resources, providing a comprehensive tool for generating performance type of anomalies.
- *slowhttptest*[15]: slowhttptest is specifically designed to test the handling of slow HTTP attacks. By simulating slow client connections and slowloris-type attacks, this tool enables the injection of anomalies related to prolonged request-response times. It aids in generating anomalies for assessing how well the system deals with slow and resource-intensive HTTP requests.

Normal Load Generation Tools

- *Locust*[16]: Locust is an open-source load testing tool that allows for the creation of scalable user scenarios. It can be employed as a normal

[13] https://httpd.apache.org/docs/2.4/programs/ab.html
[14] https://wiki.ubuntu.com/Kernel/Reference/stress-ng
[15] https://www.kali.org/tools/slowhttptest/
[16] https://locust.io/

load generator, simulating realistic user interactions with the system. Locust's flexibility and ease of use make it suitable for exploring scenarios that mimic typical user behaviors, aiding in the evaluation of system performance under normal conditions.

- *Other Load Generation Tools*: In addition to Locust, various other load generation tools can be considered based on specific testing requirements. These may include tools like *Apache JMeter*[17], *Gatling*[18], and *Siege*[19], each offering unique features for generating controlled and reproducible loads on the system.

The combination of these anomaly injection and normal load generation tools provides a comprehensive framework for assessing the effectiveness and resilience of the anomaly detection mechanisms within the testbed. Through controlled injection of anomalies and realistic generation of normal user loads, the subsystem facilitates a thorough evaluation of the system's performance under diverse conditions.

4.2.4 Monitoring Module

The Monitoring Module within the testbed setup is incorporated to serve in anomaly detection process within edge cloud environments by integrating various monitoring technologies. These technologies are crucial in providing a comprehensive understanding of the system's behavior, aiding in the identification of anomalies in edge cloud environments.

Prometheus[20], an open-source monitoring and alerting toolkit renowned for its scalability and reliability, can be employed for real-time monitoring, offering the flexibility to extract and analyze an array of metrics related to the deployed container-based microservice application. Through the utilization of customized dashboards and queries, crucial information such as response times, error rates, and resource utilization can be collected, enriching the information needed for anomaly detection within edge cloud environments.

Complementing Prometheus, Grafana[21] can be incorporated for advanced visualization of the monitored metrics. Grafana provides dynamic and interactive dashboards, offering a user-friendly interface to interpret complex data trends. By integrating Grafana, the module enhances the monitoring experience, providing clear insights into the performance of the system in edge cloud environments. Adding an extra layer of insight, Docker monitoring commands[22] can be employed to capture key metrics at the container level. This includes monitoring CPU usage, memory consumption, and network statistics. The

[17] https://jmeter.apache.org/
[18] https://gatling.io/
[19] https://github.com/JoeDog/siege
[20] https://prometheus.io/
[21] https://grafana.com/
[22] https://docs.docker.com/engine/reference/commandline/container_stats/

granularity provided by Docker monitoring allows for enriching the anomaly detection process. Focusing on the orchestration layer, the module employs Kubernetes monitoring mechanisms [23] to track the health and performance of pods. Kubernetes, as a container orchestration platform, provides native tools to monitor the state and resource utilization of pods, the smallest deployable units in the Kubernetes environment. Monitoring at this level ensures a detailed examination of the performance within individual pods. Expanding beyond the orchestration layer, the Monitoring Module extends its capabilities to the physical layer of the edge servers. This involves the collection of metrics related to server health, resource usage, and network latency. By monitoring the physical layer, the module gains a broader perspective.

By integrating these monitoring technologies, the testbed setup establishes a flexible framework that can be effectively utilized for anomaly detection in edge clouds. The diverse set of monitoring tools not only improves anomaly detection capabilities but also encourages a comprehensive understanding of the edge cloud ecosystem.

[23] https://kubernetes.io/docs/tasks/debug/debug-cluster/resource-usage-monitoring/

Chapter 5

Summary of Contributions

5.1 Paper I

J. Forough, M. Bhuyan, and E. Elmroth. Detection of VSI-DDoS Attacks on the Edge: A Sequential Modeling Approach. *In Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES)*, pp. 1-10, 2021.

5.1.1 Paper Contributions

This paper introduces a novel approach to address security concerns in edge cloud environments, particularly focusing on detection of Very Short Intermittent Distributed Denial of Service (VSI-DDoS) attacks. By leveraging Long Short-Term Memory (LSTM) with local attention, the study proposes a sequence modeling technique designed for identify short intermittent bursts of DDoS attacks. This method diverges from traditional approaches by prioritizing crucial patterns in sequence data rather than relying solely on historical information. The key contribution lies in presenting an innovative solution that improves the detection of VSI-DDoS attacks in edge cloud scenarios. This study confirms that the proposed LSTM-based method effectively identifies VSI-DDoS attacks in edge cloud environment. It emphasizes the improved detection capability and how this model could enhance security for edge computing.

5.2 Paper II

J. Forough, M. Bhuyan, and E. Elmroth. DELA: A Deep Ensemble Learning Approach for Cross-layer VSI-DDoS Detection on the Edge. *In Proceedings of the 42nd IEEE International Conference on Distributed Computing Systems*

(ICDCS), pp. 1155-1165, 2022.

5.2.1 Paper Contributions

This paper presents an innovative Deep Ensemble Learning Approach (DELA) tailored specifically for identifying cross-layer Very Short Intermittent DDoS (VSI-DDoS) attacks in edge cloud environments. To address the problem of cross-layer VSI-DDoS attacks on web applications, DELA leverages an ensemble learning strategy coupled with Long Short-Term Memory (LSTM) networks and a unique voting mechanism using Feed-Forward Neural Network (FFNN). Notably, the approach incorporates historical data into the decision-making process and employs a neural network-based aggregator, enhancing adaptability compared to traditional static threshold-based aggregations. Additionally, the proposal introduces an innovative overlapped data chunking algorithm that significantly improves detection performance, offering a robust detection mechanism for such sophisticated attacks. This paper shows DELA's superior performance through comprehensive evaluations on various testbed and benchmark datasets, demonstrating high improvements in detection accuracy compared to existing state-of-the-art methods.

5.3 Paper III

J. Forough, M. Bhuyan, and E. Elmroth. Anomaly Detection and Resolution on the Edge: Solutions and Future Directions. *In Proceedings of the IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pp. 227-238, 2023.

5.3.1 Paper Contributions

This survey paper explores the anomaly detection and resolution strategies designed explicitly for edge cloud environments. Offering a comprehensive overview, it examines the strengths, limitations, and contextual applicability of these strategies across diverse contexts. In evaluating the distinct challenges inherent to edge cloud systems, this paper provides an extensive analysis of related works and tools, providing an insightful exploration into this specialized domain. By investigating the metrics and datasets employed in various studies, it provides precious insights for evaluating the effectiveness and performance of anomaly detection and resolution techniques within edge clouds. This survey concludes by identifying open challenges, mapping future research directions, and providing recommendations.

5.4 Paper IV

J. Forough, M. Bhuyan, and E. Elmroth. Unified Identification of Anomalies on the Edge: A Hybrid Sequential PGM Approach. *In Proceedings of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2023.

5.4.1 Paper Contributions

This paper contributes by filling a crucial gap in edge cloud anomaly detection through introducing a model specifically designed to differentiate between security and performance anomalies. By using sequential modeling and Probabilistic Graphical Models (PGMs), this model explores historical data and connections among previous predictions to accurately classify upcoming anomalies. The main goal of this model is in differentiating between security threats and performance concerns in decentralized edge cloud environments. Through comprehensive evaluations using testbed and benchmark datasets, the proposed model demonstrates superior performance. Furthermore, the model's testing time analysis highlights its efficiency in early anomaly detection, demonstrating its potential in improving edge cloud security and performance.

5.5 Paper V

J. Forough, H. Haddadi, M. Bhuyan, and E. Elmroth. Efficient Anomaly Detection for Edge Clouds: Mitigating Data and Resource Constraints. *Submitted for publication, 2024.*

5.5.1 Paper Contributions

This paper contributes by proposing an innovative approach that addresses challenges related to limited computational resources and lack of labeled data specific for edge clouds. By employing transfer learning, the approach leverages knowledge from pre-existing models, adapting this knowledge to enhance anomaly detection accuracy within edge clouds. This strategy enables the model to take advantage of learned features and patterns from tasks like network intrusion detection, consequently improving its detection capability.

Furthermore, the utilization of knowledge distillation enhances computational efficiency without compromising detection accuracy. This process condenses the knowledge from a high-capacity pre-trained model into a more compact version, significantly reducing the detection time. Evaluations conducted on a testbed setup demonstrate the efficacy of this approach, showcasing remarkable reductions in detection time for both sequential and non-sequential models. These improvements in maintaining high accuracy while substantially

decreasing detection time make this approach particularly advantageous for real-time anomaly detection in edge cloud environments.

5.6 Paper VI

J. Forough, M. Bhuyan, and E. Elmroth. Reinforced Model Selection for Resource Efficient Anomaly Detection in Edge Clouds. *Submitted for publication, 2024.*

5.6.1 Paper Contributions

This paper presents an innovative approach to anomaly detection in edge cloud environments, addressing the significant challenges posed by computational limitations. By leveraging reinforcement learning, particularly Q-learning, the primary aim is to optimize resource usage of anomaly detection without compromising rapid detection times and high accuracy, which are essential requirements in edge cloud environments constrained by limited resources. The approach is extensively evaluated within a testbed setup, demonstrating promising results. It effectively reduces resource usage and inference time while maintaining reasonable accuracy, demonstrating its efficacy in optimizing resource usage for anomaly detection in edge cloud environments.

Chapter 6

Future Research Directions

As edge cloud technology evolves, various open research problems and future directions in anomaly detection emerge. This chapter highlights such problems, as well as provides suggestions for prospective areas of further research and development. The open research problems and future directions lies in several categories that are discussed in the following parts.

6.1 Efficiency and Accuracy

One critical challenge in anomaly detection for edge clouds is the limited resources of edge nodes. Developing resource-efficient anomaly detection techniques that effectively operate within constrained computational resources and minimize energy consumption is pivotal. This pursuit involves exploring lightweight algorithms [Wan+22], compressed representations [Aza+19], and optimized model architectures that balance detection accuracy with resource usage.

To enhance resource efficiency, researchers should focus on specific techniques addressing edge node limitations. For example, algorithms leveraging modern edge nodes' parallel processing capabilities can significantly enhance efficiency. Exploring novel hardware architectures like specialized accelerators or neuromorphic chips [KJS23] may further amplify edge nodes' computational capabilities for anomaly detection while conserving energy.

Considering the accuracy-resource utilization trade-off is crucial in designing resource-efficient anomaly detection techniques. While lightweight algorithms and compressed representations reduce resource demands, they may impact detection accuracy. Maintaining a balance via optimized model architectures, ensemble methods, or leveraging transfer learning can achieve resource efficiency without compromising detection performance.

Real-time anomaly detection is essential in edge cloud environments to promptly respond to anomalies. Research should prioritize the development

of real-time anomaly detection techniques meeting low-latency requirements. This involves exploring stream processing algorithms [KDA19], efficient feature extraction methods [Zeb+20], and parallel computing techniques enabling real-time detection and analysis of edge clouds data.

Efficient feature extraction reduces computational overhead associated with high-dimensional edge clouds data streams. Parallel computing techniques like GPU acceleration or distributed processing can enhance speed and scalability. Maintaining a balance between real-time performance and detection accuracy is essential. Investigating adaptive techniques dynamically adjusting this trade-off based on anomaly urgency and severity is beneficial for real-time anomaly detection in edge clouds.

6.2 Adaptability and Scalability

Edge cloud environments exhibit dynamic fluctuations in network conditions and workload patterns, demanding adaptive anomaly detection methods capable of adjusting to these changes. Future research can explore techniques for continuous learning and updating of anomaly detection models based on evolving data.

One adaptive approach involves employing online learning algorithms [Cui+19] that update anomaly detection models in real-time as new data streams in. These algorithms facilitate quick adaptation to shifting edge cloud conditions, capturing emerging anomalies effectively. Additionally, transfer learning approaches [Dag+19] can leverage knowledge from other domains or richer environments to enhance anomaly detection in edge clouds.

Moreover, reinforcement learning techniques [Moe+23] enable anomaly detection systems to autonomously adapt their behavior based on feedback and rewards. By framing anomaly detection as a sequential decision-making problem, reinforcement learning algorithms learn policies optimizing detection performance in dynamic edge cloud environments.

Research can also focus on utilizing domain-specific anomaly detection techniques to various edge cloud environments. This involves exploring domain-specific features, customizing anomaly detection models, and establishing benchmarks and evaluation methodologies unique to each domain. Understanding the context of edge applications such as smart cities [Kha+20] or healthcare [HHI22] is pivotal for effective anomaly detection. Incorporating domain knowledge and context-specific features through feature engineering can significantly enhance detection performance. The creation of domain-specific benchmark datasets and evaluation methods is essential to accelerated improvements in anomaly detection for edge cloud environments. These standardized resources enable effective comparison and evaluation of different techniques, thereby cultivating improvements in domain-specific anomaly detection for edge cloud applications.

Furthermore, the need for scalable anomaly detection is paramount. One of the main challenges to be investigated in future research is the scalability of anomaly detection methods for edge cloud environments. Future research

may delve into the integration of distributed and federated learning [Zha+21] approaches to enhance the scalability of anomaly detection in edge clouds. These methods enable collaborative learning across multiple edge nodes while preserving data privacy and minimizing communication overhead. Investigating the trade-offs between centralized and decentralized anomaly detection strategies in the context of edge computing will be crucial. Additionally, it is essential to explore the impact of diverse edge cloud architectures and topologies on the scalability of anomaly detection methods. As edge environments vary in terms of device heterogeneity, network connectivity, and workload distribution, understanding how these factors influence the scalability of anomaly detection algorithms will be essential.

6.3 Privacy

Edge cloud environments are often involved with sensitive data, arising significant concerns about privacy and data protection. Future research initiatives can consider the development of cutting-edge privacy-preserving anomaly detection techniques that uphold data confidentiality and align with strict privacy regulations. This investigation includes an exploration of diverse methods such as secure multi-party computation [Lin20], federated learning [Zha+21], and differential privacy [Wan+20b] to enable anomaly detection without compromising the integrity of data privacy.

Secure multi-party computation techniques enable multiple edge nodes to collaboratively perform anomaly detection without disclosing individual data, thereby ensuring robust privacy while obtaining precise and accurate results. Employing federated learning approaches allows the training of anomaly detection models across distributed edge devices without transmitting raw data, thus preserving data privacy. Integrating differential privacy techniques into the anomaly detection process involves introducing noise or perturbations to data, safeguarding individual privacy while still obtaining relevant aggregate information for detection purposes. It is vital for privacy-preserving anomaly detection methods to achieve a delicate balance between ensuring privacy and maintaining detection accuracy.

6.4 Robustness

Edge cloud environments face a range of adversarial attacks aimed at damaging anomaly detection systems. To strengthen these systems, future research can explore strategies to help the resilience of anomaly detection methods against such malicious attacks. This encompasses investigation of adversarial training [AF20], anomaly detection leveraging anomaly injection [Kun+20], and dealing with data poisoning attacks within anomaly detection frameworks [BIA22].

Adversarial training involves exposing anomaly detection models to adversarial examples during their training phase, rendering them more resilient

to potential attacks. By integrating adversarial samples designed to deceive the model into the training process, these models can develop a more robust resistance against malicious manipulations. Similarly, anomaly detection methods that introduce artificial anomalies into training data can help with the learning of more diverse and resilient representations of anomalies, enhancing the model's robustness. Furthermore, the development of anomaly detection techniques capable of detecting and mitigating data poisoning attacks is critical for upholding the integrity of edge cloud environments. These attacks involve malicious entities injecting corrupt data to manipulate the anomaly detection process, potentially compromising its accuracy. Thus, research investigations can concentrate on formulating anomaly detection algorithms adept at identifying and neutralizing data poisoning attacks while preserving high detection performance.

6.5 Trust and Explainability

As anomaly detection in edge clouds progresses, establishing trust in these systems becomes paramount for their widespread adoption and effective operation [YW22]. Trust extends beyond the reliability of anomaly detection results to encompass transparency, interpretability, and accountability throughout the entire detection process [HT21]. Future research in this domain can investigate various aspects to enhance trust in anomaly detection systems for edge clouds.

Building trust begins with ensuring the explainability and interpretability of anomaly detection models [PA21] operating in edge cloud environments. These models should provide clear explanations for their decisions, particularly in critical applications such as healthcare or autonomous systems. Research efforts can focus on developing interpretable anomaly detection techniques that offer clarity into the factors contributing to detected anomalies.

Another critical dimension of trust involves ensuring the trustworthiness of anomaly detection models against various inputs, environmental conditions, and potential adversarial attacks [Yan+20; YW22]. Research can investigate evaluating the resilience of models in real-world edge cloud scenarios, considering factors like data quality, distribution shifts, and uncertainties [Sal+21]. Establishing methods for validating the trustworthiness of anomaly detection models will be instrumental in building confidence among users and decision-makers.

Transparency in the model training process is also essential for improving trust [LZV23]. Future research can explore methodologies to make the model training pipeline more transparent, encompassing aspects such as the selection of training data, hyperparameter tuning, and model updates. Transparent training processes allow stakeholders to understand how models evolve over time and ensure that they align with ethical considerations and regulatory requirements.

6.6 Application in Other Domains

The novel anomaly detection models developed in this thesis for edge cloud environments, leveraging state-of-the-art machine learning techniques such as sequential models (LSTM, GRU), CRF, ensemble learning, knowledge distillation, transfer learning, and reinforced learning, hold the potential for broader applications in various domains. The versatility and adaptability of these models make them valuable candidates for exploration in diverse areas over edge cloud anomaly detection.

Research efforts can extend beyond edge cloud environments to explore how these anomaly detection models generalize to diverse domains characterized by time-series and temporal data. Investigating their effectiveness in tasks ranging from financial fraud detection [PP20] to predictive maintenance [Dal+20] and healthcare monitoring [Šab+21] will contribute to a comprehensive understanding of their applicability and potential impact.

Bibliography

[24a] *Acme Air Sample and Benchmark*. `https://github.com/acmeair/acmeair`. Last checked: 2024-01-12. 2024.

[24b] *Bifrost Microservices Sample Application*. `https://github.com/sealuzh/bifrost-microservices-sample-application`. Last checked: 2024-01-12. 2024.

[24c] *MicroSuite: A Benchmark Suite for Microservices*. `https://github.com/wenischlab/MicroSuite`. Last checked: 2024-01-12. 2024.

[24d] *Spring Cloud Example Project*. `https://github.com/kbastani/spring-cloud-microservice-example`. Last checked: 2024-01-12. 2024.

[AF20] Maksym Andriushchenko and Nicolas Flammarion. "Understanding and Improving Fast Adversarial Training". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16048–16059.

[Al-+21] Redhwan Al-amri, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A Al-Sharafi, and Ammar Ahmed Alkahtani. "A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data". In: *Applied Sciences* 11.12 (2021), p. 5320.

[Alb+22] Jose Edson de Albuquerque Filho, Laislla CP Brandao, Bruno José Torres Fernandes, and Alexandre MA Maciel. "A Review of Neural Networks for Anomaly Detection". In: *IEEE Access* 10 (2022), pp. 112342–112367.

[Aza+19] Joseph Azar, Abdallah Makhoul, Mahmoud Barhamgi, and Raphaël Couturier. "An Energy Efficient IoT Data Compression Approach for Edge Machine Learning". In: *Future Generation Computer Systems* 96 (2019), pp. 168–175.

[Ban21] Vamsikrishna Bandari. "Predictive Analytics in Cloud Computing: An ARIMA Model Study on Performance Metrics". In: *Applied Research in Artificial Intelligence and Cloud Computing* 4.1 (2021), pp. 1–18.

[Bhu+22] Adil Bin Bhutto, Xuan Son Vu, Erik Elmroth, Wee Peng Tay, and Monowar Bhuyan. "Reinforced Transformer Learning for VSI-DDoS Detection in Edge Clouds". In: *IEEE Access* 10 (2022), pp. 94677–94690.

[BIA22] Shameek Bhattacharjee, Mohammad Jaminur Islam, and Sahar Abedzadeh. "Robust Anomaly-based Attack Detection in Smart Grids under Data Poisoning Attacks". In: *Proceedings of the 8th ACM on Cyber-Physical System Security Workshop*. 2022, pp. 3–14.

[BS21] Priyajit Biswas and Tuhina Samanta. "Anomaly Detection using Ensemble Random Forest in Wireless Sensor Network". In: *International Journal of Information Technology* 13.5 (2021), pp. 2043–2052.

[Car+21] Gonçalo Carvalho, Bruno Cabral, Vasco Pereira, and Jorge Bernardino. "Edge Computing: Current Trends, Research Challenges and Future Directions". In: *Computing* 103 (2021), pp. 993–1023.

[Che+17] Zhuo Chen, Wenlu Hu, Junjue Wang, Siyan Zhao, Brandon Amos, Guanhang Wu, Kiryong Ha, Khalid Elgazzar, Padmanabhan Pillai, Roberta Klatzky, et al. "An Empirical Study of Latency in an Emerging Class of Edge Computing Applications for Wearable Cognitive Assistance". In: *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 2017, pp. 1–14.

[Che+21] Xing Chen, Jianshan Zhang, Bing Lin, Zheyi Chen, Katinka Wolter, and Geyong Min. "Energy-Efficient Offloading for DNN-Based Smart IoT Systems in Cloud-Edge Environments". In: *IEEE Transactions on Parallel and Distributed Systems* 33.3 (2021), pp. 683–697.

[Cho+23] Hong-Cheol Choi, Chuhao Deng, Hyunsang Park, and Inseok Hwang. "Gaussian Mixture Model-Based Online Anomaly Detection for Vectored Area Navigation Arrivals". In: *Journal of Aerospace Information Systems* 20.1 (2023), pp. 37–52.

[CKK21] Efstratios Chatzoglou, Georgios Kambourakis, and Constantinos Kolias. "Empirical Evaluation of Attacks Against IEEE 802.11 Enterprise Networks: The AWID3 Dataset". In: *IEEE Access* 9 (2021), pp. 34188–34205.

[Cop+17] Luigi Coppolino, Salvatore D'Antonio, Giovanni Mazzeo, and Luigi Romano. "Cloud Security: Emerging Threats and Current Solutions". In: *Computers & Electrical Engineering* 59 (2017), pp. 126–140.

[Cui+19] Qimei Cui, Zhenzhen Gong, Wei Ni, Yanzhao Hou, Xiang Chen, Xiaofeng Tao, and Ping Zhang. “Stochastic Online Learning for Mobile Edge Computing: Learning from Changes”. In: *IEEE Communications Magazine* 57.3 (2019), pp. 63–69.

[Dag+19] Harshit Daga, Patrick K Nicholson, Ada Gavrilovska, and Diego Lugones. “Cartel: A System for Collaborative Transfer Learning at the Edge”. In: *Proceedings of the ACM Symposium on Cloud Computing*. 2019, pp. 25–37.

[Dal+20] Jovani Dalzochio, Rafael Kunst, Edison Pignaton, Alecio Binotto, Srijnan Sanyal, Jose Favilla, and Jorge Barbosa. “Machine Learning and Reasoning for Predictive Maintenance in Industry 4.0: Current Status and Challenges”. In: *Computers in Industry* 123 (2020), p. 103298.

[Dou+23] Maryam Douiba, Said Benkirane, Azidine Guezzaz, and Mourad Azrour. “Anomaly Detection Model based on Gradient Boosting and Decision Tree for IoT Environments Security”. In: *Journal of Reliable Intelligent Environments* 9.4 (2023), pp. 421–432.

[DPP21] Francesco Daghero, Daniele Jahier Pagliari, and Massimo Poncino. “Energy-Efficient Deep Learning Inference on Edge Devices”. In: *Advances in Computers*. Vol. 122. Elsevier, 2021, pp. 247–301.

[Erh+21] Laura Erhan, M Ndubuaku, Mario Di Mauro, Wei Song, Min Chen, Giancarlo Fortino, Ovidiu Bagdasar, and Antonio Liotta. “Smart Anomaly Detection in Sensor Systems: A Multi-perspective Review”. In: *Information Fusion* 67 (2021), pp. 64–79.

[ERM18] Clément Elbaz, Louis Rilling, and Christine Morin. “Reactive and Adaptive Security Monitoring in Cloud Computing”. In: *IEEE 3rd International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. 2018, pp. 5–7.

[FBE21] Javad Forough, Monowar Bhuyan, and Erik Elmroth. “Detection of VSI-DDoS Attacks on the Edge: A Sequential Modeling Approach”. In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*. 2021, pp. 1–10.

[FBE22] Javad Forough, Monowar Bhuyan, and Erik Elmroth. “DELA: A Deep Ensemble Learning Approach for Cross-layer VSI-DDoS Detection on the Edge”. In: *IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. 2022, pp. 1155–1165.

[FBE23a] Javad Forough, Monowar Bhuyan, and Erik Elmroth. “Anomaly Detection and Resolution on the Edge: Solutions and Future Directions”. In: *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE. 2023, pp. 227–238.

[FBE23b] Javad Forough, Monowar Bhuyan, and Erik Elmroth. “Unified Identification of Anomalies on the Edge: A Hybrid Sequential PGM Approach”. In: *Proceedings of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 2023.

[Fer+12] Michael Ferdman, Almutaz Adileh, Onur Kocberber, Stavros Volos, Mohammad Alisafaee, Djordje Jevdjic, Cansu Kaynak, Adrian Daniel Popescu, Anastasia Ailamaki, and Babak Falsafi. “Clearing the Clouds: A Study of Emerging Scale-out Workloads on Modern Hardware”. In: *SIGPLAN Not.* 47.4 (Mar. 2012), pp. 37–48.

[Fer+24] Michael Ferdman et al. *Cloudsuite: A Benchmark Suite for Cloud Services*. `https://github.com/parsa-epfl/cloudsuite`. Last checked: 2024-01-12. 2024.

[Gad+22] Saad Gadal, Rania Mokhtar, Maha Abdelhaq, Raed Alsaqour, Elmustafa Sayed Ali, and Rashid Saeed. “Machine Learning-Based Anomaly Detection Using K-Mean Array and Sequential Minimal Optimization”. In: *Electronics* 11.14 (2022), p. 2158.

[Gan+19] Yu Gan, Yanqi Zhang, Dailun Cheng, Ankitha Shetty, Priyal Rathi, Nayan Katarki, Ariana Bruno, Justin Hu, Brian Ritchken, Brendon Jackson, et al. “An Open-source Benchmark Suite for Microservices and Their Hardware-software Implications for Cloud & Edge Systems”. In: *ASPLOS ’19*. USA: ACM, 2019, pp. 3–18.

[Gan22] Y. Gan. *Deathstarbench: Open-source Benchmark Suite for Cloud Microservices*. `https://github.com/delimitrou/DeathStarBench`. Last checked: 2023-06-12. 2022.

[GLH15] Salvador García, Julián Luengo, and Francisco Herrera. *Data Preprocessing in Data Mining*. Vol. 72. Springer, 2015.

[GMC20] Reza Ghezelbash, Abbas Maghsoudi, and Emmanuel John M Carranza. “Optimization of Geochemical Anomaly Detection using a Novel Genetic K-Means Clustering (GKMC) Algorithm”. In: *Computers & Geosciences* 134 (2020), p. 104335.

[Goo23] Google Cloud Platform. *Online Boutique: A Cloud-first Microservices Demo Application*. `https://github.com/GoogleCloudPlatform/microservices-demo`. Last checked: 2023-06-12. 2023.

[HAA20] Salam Hamdan, Moussa Ayyash, and Sufyan Almajali. “Edge-Computing Architectures for Internet of Things Applications: A Survey”. In: *Sensors* 20.22 (2020), p. 6441.

[HHI22] Morghan Hartmann, Umair Sajid Hashmi, and Ali Imran. “Edge Computing in Smart Health Care Systems: Review, Challenges, and Research Directions”. In: *Transactions on Emerging Telecommunications Technologies* 33.3 (2022), e3710.

[Hos+21] Mehdi Hosseinzadeh, Amir Masoud Rahmani, Bay Vo, Moazam Bidaki, Mohammad Masdari, and Mehran Zangakani. "Improving Security Using SVM-Based Anomaly Detection: Issues and Challenges". In: *Soft Computing* 25 (2021), pp. 3195–3223.

[HT21] Aleks Huč and Denis Trček. "Anomaly Detection in IoT Networks: From Architectures to Machine Learning Transparency". In: *IEEE Access* 9 (2021), pp. 60607–60616.

[JA19] Reiner Jung and Marc Adolf. "The Jpetstore Suite: A Concise Experiment Setup for Research". In: *Softwaretechnik-Trends* 39.3 (Nov. 2019), pp. 40–42.

[JD23] Vikas Juneja and Shail Kumar Dinkar. "A Predictive Vampire Attack Detection by Social Spider Optimized Gaussian Mixture Model Clustering". In: *Concurrency and Computation: Practice and Experience* 35.2 (2023), e7481.

[Jia+20] Congfeng Jiang, Tiantian Fan, Honghao Gao, Weisong Shi, Liangkai Liu, Christophe Cérin, and Jian Wan. "Energy Aware Edge Computing: A Survey". In: *Computer Communications* 151 (2020), pp. 556–580.

[JSW17] Gauri Joshi, Emina Soljanin, and Gregory Wornell. "Efficient Redundancy Techniques for Latency Reduction in Cloud Systems". In: *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 2.2 (2017), pp. 1–30.

[KDA19] Taiwo Kolajo, Olawande Daramola, and Ayodele Adebiyi. "Big Data Stream Analysis: a Systematic Literature Review". In: *Journal of Big Data* 6.1 (2019), p. 47.

[Kha+20] Latif U Khan, Ibrar Yaqoob, Nguyen H Tran, SM Ahsan Kazmi, Tri Nguyen Dang, and Choong Seon Hong. "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey". In: *IEEE Internet of Things Journal* 7.10 (2020), pp. 10200–10232.

[Khe+23] Vinit Khetani, Yatin Gandhi, Saurabh Bhattacharya, Samir N Ajani, and Suresh Limkar. "Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains". In: *International Journal of Intelligent Systems and Applications in Engineering* 11.7s (2023), pp. 253–262.

[KJS23] Arash Khajooei, Mohammad Jamshidi, and Shahriar B Shokouhi. "A Super-Efficient TinyML Processor for the Edge Metaverse". In: *Information* 14.4 (2023), p. 235.

[KKS20] Manoj Kumar, Harsh Kumar Verma, and Geeta Sikka. "A Secure Data Transmission Protocol for Cloud-Assisted Edge-Internet of Things Environment". In: *Transactions on Emerging Telecommunications Technologies* 31.6 (2020), e3883.

[Kor+20] Ioannis Korontanis, Konstantinos Tserpes, Maria Pateraki, Lorenzo Blasi, John Violos, Ferran Diego, Eduard Marin, Nicolas Kourtellis, Massimo Coppola, Emanuele Carlini, et al. "Inter-Operability and Orchestration in Heterogeneous Cloud/Edge Resources: The Accordion Vision". In: *Proceedings of the 1st Workshop on Flexible Resource and Application Management on the Edge*. 2020, pp. 9–14.

[Kun+20] Arnav Kundu, Abhijeet Sahu, Erchin Serpedin, and Katherine Davis. "A3d: Attention-based Auto-encoder Anomaly Detector for False Data Injection Attacks". In: *Electric Power Systems Research* 189 (2020), p. 106795.

[Les+21] Julien Lesouple, Cédric Baudoin, Marc Spigai, and Jean-Yves Tourneret. "Generalized Isolation Forest for Anomaly Detection". In: *Pattern Recognition Letters* 149 (2021), pp. 109–119.

[LHH23] Cong Lu, Jianbin Huang, and Longji Huang. "Detecting Urban Anomalies using Factor Analysis and One Class Support Vector Machine". In: *The Computer Journal* 66.2 (2023), pp. 373–383.

[Li+19] Chunlin Li, Hezhi Sun, Hengliang Tang, and Youlong Luo. "Adaptive Resource Allocation based on the Billing Granularity in Edge-Cloud Architecture". In: *Computer Communications* 145 (2019), pp. 29–42.

[Li+21] Rui Li, Zhinan Cheng, Patrick PC Lee, Pinghui Wang, Yi Qiang, Lin Lan, Cheng He, Jinlong Lu, Mian Wang, and Xinquan Ding. "Automated Intelligent Healing in Cloud-scale Data Centers". In: *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*. IEEE. 2021, pp. 244–253.

[Lin20] Yehuda Lindell. "Secure Multiparty Computation". In: *Communications of the ACM* 64.1 (2020), pp. 86–96.

[Liu+19] Shaoshan Liu, Liangkai Liu, Jie Tang, Bo Yu, Yifan Wang, and Weisong Shi. "Edge Computing for Autonomous Driving: Opportunities and Challenges". In: *Proceedings of the IEEE* 107.8 (2019), pp. 1697–1716.

[LJ23] Gen Li and Jason J Jung. "Deep Learning for Anomaly Detection in Multivariate Time Series: Approaches, Applications, and Challenges". In: *Information Fusion* 91 (2023), pp. 93–102.

[LZV23] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. "A Survey on Explainable Anomaly Detection". In: *ACM Transactions on Knowledge Discovery from Data* 18.1 (2023), pp. 1–54.

[Ma+21] Qian Ma, Cong Sun, Baojiang Cui, and Xiaohui Jin. "A Novel Model for Anomaly Detection in Network Traffic based on Kernel Support Vector Machine". In: *Computers & Security* 104 (2021), p. 102215.

[Moe+23] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. "Model-based Reinforcement Learning: A Survey". In: *Foundations and Trends® in Machine Learning* 16.1 (2023), pp. 1–118.

[MS15] Nour Moustafa and Jill Slay. "UNSW-NB15: a Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)". In: *military communications and information systems conference (MilCIS)*. 2015, pp. 1–6.

[Mut+20] Wamidh K Mutlag, Shaker K Ali, Zahoor M Aydam, and Bahaa H Taher. "Feature Extraction Methods: A Review". In: *Journal of Physics: Conference Series*. Vol. 1591. 1. IOP Publishing. 2020, p. 012028.

[Nou+19] Subrina Sultana Noureen, Stephen B Bayne, Edward Shaffer, Donald Porschet, and Morris Berman. "Anomaly Setection in Cyber-Physical System using Logistic Regression Analysis". In: *2019 IEEE Texas Power and Energy Conference (TPEC)*. IEEE. 2019, pp. 1–6.

[PA21] Guansong Pang and Charu Aggarwal. "Toward Explainable Deep Anomaly Detection". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 4056–4057.

[Pal19] Francesco Palmieri. "Network Anomaly Detection based on Logistic Regression of Nonlinear Chaotic Invariants". In: *Journal of Network and Computer Applications* 148 (2019), p. 102460.

[PM17] Jianli Pan and James McElhannon. "Future Edge Cloud and Edge Computing for Internet of Things Applications". In: *IEEE Internet of Things Journal* 5.1 (2017), pp. 439–449.

[PP20] C Victoria Priscilla and D Padma Prabha. "Credit Card Fraud Detection: A Systematic Review". In: *Intelligent Computing Paradigm and Cutting-edge Technologies: Proceedings of the First International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2019), Istanbul, Turkey, October 30-31, 2019 1*. Springer. 2020, pp. 290–303.

[PSF16] Tapti Palit, Yongming Shen, and Michael Ferdman. "Demystifying Cloud Benchmarking". In: *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. USA: IEEE, 2016, pp. 122–132.

[PT17] Rifkie Primartha and Bayu Adhi Tama. "Anomaly Detection using Random Forest: A Performance Revisited". In: *2017 International conference on data and software engineering (ICoDSE)*. IEEE. 2017, pp. 1–6.

[Pu+20] Guo Pu, Lijuan Wang, Jun Shen, and Fang Dong. "A Hybrid Unsupervised Clustering-Based Anomaly Detection Method". In: *Tsinghua Science and Technology* 26.2 (2020), pp. 146–153.

[QWJ21] Yan Qiao, Kui Wu, and Peng Jin. "Efficient Anomaly Detection for High-Dimensional Sensing Data with One-Class Support Vector Machine". In: *IEEE Transactions on Knowledge and Data Engineering* 35.1 (2021), pp. 404–417.

[Red+21] Dukka KarunKumar Reddy, Himansu Sekhar Behera, Janmenjoy Nayak, Pandi Vijayakumar, Bighnaraj Naik, and Pradeep Kumar Singh. "Deep Neural Network based Anomaly Detection in Internet of Things Network Traffic Tracking for the Applications of Future Smart Cities". In: *Transactions on Emerging Telecommunications Technologies* 32.7 (2021), e4121.

[Ren+19] Ju Ren, Deyu Zhang, Shiwen He, Yaoxue Zhang, and Tao Li. "A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms: Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet". In: *ACM Computing Surveys (CSUR)* 52.6 (2019), pp. 1–36.

[RK22] Rohit Ranjan and Shashi Shekhar Kumar. "User Behaviour Analysis using Data Analytics and Machine Learning to Predict Malicious User Versus Legitimate User". In: *High-Confidence Computing* 2.1 (2022), p. 100034.

[Šab+21] Edin Šabić, David Keeley, Bailey Henderson, and Sara Nannemann. "Healthcare and Anomaly Detection: Using Machine Learning to Predict Anomalies in Heart Rate Data". In: *AI & SOCIETY* 36.1 (2021), pp. 149–158.

[Sal+21] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. "A Unified Survey on Anomaly, Novelty, Open-Set, and Out-of-Distribution Detection: Solutions and Future Challenges". In: *arXiv preprint arXiv:2110.14051* (2021).

[Sal+23] Mohammad Reza Saleh Sedghpour, Aleksandra Obeso Duque, Xuejun Cai, Björn Skubic, Erik Elmroth, Cristian Klein, and Johan Tordsson. "Hydragen: A Microservice Benchmark Generator". In: *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*. 2023.

[Sha+19] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A Ghorbani. "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy". In: *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE. 2019, pp. 1–8.

[Shi+12] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, and Ali A Ghorbani. "Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection". In: *computers & security* 31.3 (2012), pp. 357–374.

[Shi+16] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. "Edge Computing: Vision and Challenges". In: *IEEE internet of things journal* 3.5 (2016), pp. 637–646.

[Sir+21] Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, and Mika Ylianttila. "A Survey on Mobile Augmented Reality with 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects". In: *IEEE Communications Surveys & Tutorials* 23.2 (2021), pp. 1160–1192.

[SKT21] Mohammad Reza Saleh Sedghpour, Cristian Klein, and Johan Tordsson. "Service Mesh Circuit Breaker: From Panic Button to Performance Management Tool". In: *Proceedings Of The 1st Workshop On High Availability And Observability Of Cloud Systems*. 2021, pp. 4–10.

[Spr24] Spring. *Spring Petclinic: Distributed Version of Spring Petclinic Built with Spring Cloud*. `https://github.com/spring-petclinic/spring-petclinic-microservices`. Last checked: 2024-01-12. 2024.

[SW18] Akshitha Sriraman and Thomas F Wenisch. "μ Suite: A Benchmark Suite for Microservices". In: *2018 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 2018, pp. 1–12.

[Tav+09] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. "A Detailed Analysis of The KDD CUP 99 Data Set". In: *IEEE symposium on computational intelligence for security and defense applications*. 2009, pp. 1–6.

[Tiw22] Ashish Tiwari. "Supervised Learning: From Theory to Applications". In: *Artificial intelligence and machine learning for EDGE computing*. Elsevier, 2022, pp. 23–32.

[TMG23] Hasan Torabi, Seyedeh Leili Mirtaheri, and Sergio Greco. "Practical Autoencoder based Anomaly Detection by using Vector Reconstruction Error". In: *Cybersecurity* 6.1 (2023), p. 1.

[Uni24] University of Wurzburg. *Teastore: A Micro-service Reference Test Application*. `https://github.com/DescartesResearch/TeaStore`. Last checked: 2024-01-12. 2024.

[Vañ+23] Rafael Vaño, Ignacio Lacalle, Piotr Sowiński, Raúl S-Julián, and Carlos E Palau. "Cloud-Native Workload Orchestration at the Edge: A Deployment Review and Future Directions". In: *Sensors* 23.4 (2023), p. 2215.

[Var+21] Shay Vargaftik, Isaac Keslassy, Ariel Orda, and Yaniv Ben-Itzhak. "RADE: Resource-Efficient Supervised Anomaly Detection using Decision Tree-Based Ensemble Methods". In: *Machine Learning* 110.10 (2021), pp. 2835–2866.

[VHM20] Jan Vom Brocke, Alan Hevner, and Alexander Maedche. "Introduction to Design Science Research". In: *Design science research. Cases* (2020), pp. 1–13.

[VMB22] Xuan-Son Vu, Maode Ma, and Monowar Bhuyan. "MetaVSID: A Robust Meta-Reinforced Learning Approach for VSI-DDoS Detection on the Edge". In: *IEEE Transactions on Network and Service Management* (2022).

[Von+18] Joakim Von Kistowski, Simon Eismann, Norbert Schmitt, André Bauer, Johannes Grohmann, and Samuel Kounev. "Teastore: A Micro-service Reference Application for Benchmarking, Modeling and Resource Management Research". In: *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 2018, pp. 223–236.

[Wan+20a] Bingming Wang, Shi Ying, Guoli Cheng, Rui Wang, Zhe Yang, and Bo Dong. "Log-Based Anomaly Detection With the Improved K-Nearest Neighbor". In: *International Journal of Software Engineering and Knowledge Engineering* 30.02 (2020), pp. 239–262.

[Wan+20b] Tian Wang, Yaxin Mei, Weijia Jia, Xi Zheng, Guojun Wang, and Mande Xie. "Edge-based Differential Privacy Computing for Sensor–cloud Systems". In: *Journal of Parallel and Distributed computing* 136 (2020), pp. 75–85.

[Wan+22] Zumin Wang, Jiyu Tian, Hui Fang, Liming Chen, and Jing Qin. "LightLog: A Lightweight Temporal Convolutional Network for Log Anomaly Detection on the Edge". In: *Computer Networks* 203 (2022), p. 108616.

[Wea22] Weaveworks. *Sock Shop: A Microservices Demo Application*. `https://github.com/microservices-demo/microservices-demo`. Last checked: 2023-06-12. 2022.

[Wib+21] S Wibisono, MT Anwar, Aji Supriyanto, and IHA Amin. "Multivariate Weather Anomaly Detection using DBSCAN Clustering Algorithm". In: *Journal of Physics: Conference Series*. Vol. 1869. 1. IOP Publishing. 2021, p. 012077.

[Xia+19] Yinhao Xiao, Yizhen Jia, Chunchi Liu, Xiuzhen Cheng, Jiguo Yu, and Weifeng Lv. "Edge Computing Security: State of the Art and Challenges". In: *Proceedings of the IEEE* 107.8 (2019), pp. 1608–1631.

[Xu+23] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. "Deep Isolation Forest for Anomaly Detection". In: *IEEE Transactions on Knowledge and Data Engineering* (2023).

[Yan+20] Xiaodan Yan, Yang Xu, Xiaofei Xing, Baojiang Cui, Zihao Guo, and Taibiao Guo. "Trustworthy Network Anomaly Detection based on an Adaptive Learning Rate and Momentum in IIoT". In: *IEEE Transactions on Industrial Informatics* 16.9 (2020), pp. 6182–6192.

[Yin+21] Shi Ying, Bingming Wang, Lu Wang, Qingshan Li, Yishi Zhao, Jianga Shang, Hao Huang, Guoli Cheng, Zhe Yang, and Jiangyi Geng. "An Improved KNN-Based Efficient Log Anomaly Detection Method with Automatically Labeled Samples". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15.3 (2021), pp. 1–22.

[Yun+23] Huitaek Yun, Hanjun Kim, Young Hun Jeong, and Martin BG Jun. "Autoencoder-Based Anomaly Detection of Industrial Robot Arm Using Stethoscope Based Internal Sound Sensor". In: *Journal of Intelligent Manufacturing* 34.3 (2023), pp. 1427–1444.

[YW22] Shuhan Yuan and Xintao Wu. "Trustworthy Anomaly Detection: A Survey". In: *arXiv preprint arXiv:2202.07787* (2022).

[Zeb+20] Rizgar Zebari, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction". In: *Journal of Applied Science and Technology Trends* 1.2 (2020), pp. 56–70.

[Zha+18] Jiale Zhang, Bing Chen, Yanchao Zhao, Xiang Cheng, and Feng Hu. "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues". In: *IEEE access* 6 (2018), pp. 18209–18237.

[Zha+21] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. "A Survey on Federated Learning". In: *Knowledge-Based Systems* 216 (2021), p. 106775.

[Zho+18] Xiang Zhou, Xin Peng, Tao Xie, Jun Sun, Chenjie Xu, Chao Ji, and Wenyun Zhao. "Benchmarking Microservice Systems for Software Engineering Research". In: *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*. 2018, pp. 323–324.

[Zho+22] X. Zhou et al. *Train Ticket: A Benchmark Microservice System*. `https://github.com/FudanSELab/train-ticket`. Last checked: 2023-06-12. 2022.