**Table of Contents**

[1. Introduction 2](#_Toc182436147)

[1.1. Problem Statement 2](#_Toc182436148)

[1.2. Objectives 3](#_Toc182436149)

[1.3. Research Question 4](#_Toc182436150)

[1.4. Significance of this study 4](#_Toc182436151)

[2. Background 6](#_Toc182436152)

[2.1. Theoretical Foundations 6](#_Toc182436153)

[2.1.1. Federated Learning (FL) 6](#_Toc182436154)

[2.1.2. Anomaly Detection 7](#_Toc182436155)

[2.1.3. Kubernetes and Microservices Architecture 8](#_Toc182436156)

[2.1.4. API Security and API Abuse Detection 8](#_Toc182436157)

[2.2. Historical Development 8](#_Toc182436158)

[2.2.1. Federated Learning (FL) 9](#_Toc182436159)

[2.2.2. Microservices 9](#_Toc182436160)

[3. Literature Review 11](#_Toc182436161)

[3.1. Federated Learning in Cloud-Based environments 11](#_Toc182436162)

[3.2. Anomaly Detection in Microservices 12](#_Toc182436163)

[3.3. Kubernetes Orchestration and Microservice Architecture 14](#_Toc182436164)

[3.4. API Abuse Detection in Microservices 15](#_Toc182436165)

[3.4.1. API Abuse Detection and Rate-Limiting Violations 15](#_Toc182436166)

[3.4.2. Anomaly Detection in Microservices 16](#_Toc182436167)

# Introduction

Securing communication among microservices is crucial in today's distributed computing world, as applications are dispersed across several servers (Atlassian, n.d.). Kubernetes has grown as a popular solution for managing these microservices, offering flexibility and scalability by splitting down programs into smaller, independent services (Atlassian, n.d.). However, this also increases the potential for API abuse, as attackers use APIs to obtain unwanted access, disrupt services, or steal data.

One significant threat is rate limitation abuse, when attackers try to evade constraints on how often they may visit an API. Attackers exploit APIs to gain unauthorized access, overwhelm systems, or exfiltrate sensitive data. Traditional security measures generally fail to notice these sophisticated assaults in real time (Xperts & Xperts, 2024).

Federated learning offers a potential solution to this challenge. Instead of gathering all data in one central place, federated learning trains models across scattered nodes, accumulating just the model updates from each node rather than raw data (Park & Joe-Wong, 2024). This strategy promotes privacy, which is vital in security applications (Pandiya, 2024). By utilizing federated learning within Kubernetes, each microservice may become an expert at identifying aberrant API activity, including rate-limiting abuse, without disclosing sensitive information (Sabuhi et al., 2024).

Integrating federated learning with deep learning-based anomaly detection develops a resilient system that can identify complex and developing attack patterns that can go undiscovered by standard security approaches (Vucovich et al., 2022). Kubernetes provides the appropriate infrastructure for this, allowing us to quickly manage and scale these anomaly detection models across all microservices (Preuveneers et al., 2018). Together, these solutions build a resilient and adaptable security architecture that secures APIs without sacrificing data privacy (Sharma et al., 2024). This dissertation studies how federated learning, used within Kubernetes, may efficiently identify rate-limiting abuse, making modern microservice architectures safer and more dependable.

## Problem Statement

Microservices have changed the way we design applications, enabling extraordinary scalability, flexibility, and resilience. However, this technique offers additional security risks, particularly when it comes to maintaining the large network of APIs that connect these services. (Atlassian, n.d.)

Each microservice employs APIs to connect with others, which are required for interaction but could present vulnerabilities that attackers might exploit (Abdelfattah and Cerny, 2022). API use has become a severe security problem (Murray, 2024). One important disadvantage in this situation is the opportunity for rate limiter abuse. Rate limitation is a key approach for regulating the flow of API calls, reducing overload, and saving resources. However, attackers can try to escape these limits to overwhelm the system, disrupt services, or gain unauthorized access (Murray, 2024).

Traditional security solutions, like role-based access controls or defined rate limits, generally struggle to keep up with the ever-changing techniques of attackers in microservices (NordLayer, n.d.). Attackers are always upgrading their strategies to blend in with legitimate API traffic, making them tougher to recognise (Microservices Security—OWASP Cheat Sheet Series, n.d.). The dispersed structure of microservices makes it tougher to monitor and evaluate API activity appropriately (Chandramouli, 2019).

Each service in a microservices system only monitors a small portion of overall activity, making it difficult to spot anomalies correctly (De Almeida & Canedo, 2022). This problem is magnified in cloud-based settings like Kubernetes, where microservices are always being deployed and scaled automatically, leading to unpredictable changes in API traffic patterns.

The primary challenge lies in developing a security solution that can respond to these changes, detect API misuse across numerous services, and accomplish all this without affecting the system's privacy or performance (Outshift | Securing API Calls in Kubernetes: A Simple and Effective Approach, n.d.). This dissertation intends to address this challenge by examining how federated learning can be utilised for distributed anomaly detection in Kubernetes, with a specific focus on detecting and avoiding rate-limiting abuse. This approach offers a mechanism to secure APIs in complicated microservices systems while ensuring robustness and preserving data privacy.

## Objectives

The project aims to improve the security of microservice architectures within Kubernetes by concentrating on the detection and prevention of API rate limitation abuse. To attain this aim, the following precise objectives are defined:

1. Design a Federated Learning Framework for spotting API misuse in Kubernetes-based microservices, protecting data privacy by keeping raw data local to each service.
2. Develop an anomaly detection to find abnormal API usage patterns that suggest potential exploitation.
3. Set up Prometheus and Grafana for real-time monitoring and visualisation of API interactions and system data, enabling continuous abuse detection.
4. Validate the system’s accuracy and response under abuse and high-load scenarios to verify it reliably detects malicious API activity in real time.

## Research Question

The primary research question is: How can federated learning be utilised to identify API rate-limiting violations in Kubernetes-based microservice architectures employing pre-trained anomaly detection models while maintaining privacy and scalability? This study examines the potential of federated learning to enhance API security through the detection of misuse in distributed microservice contexts while addressing privacy issues and improving scalability.

## Significance of this study

This research is crucial for maintaining the security of microservices, particularly in preventing API abuse. As more and more companies use cloud-based apps and microservices managed by Kubernetes, we must ensure that the communication between these services remains safe. This dissertation offers a solution using federated learning to boost security without compromising data privacy. This is important for both researchers and companies because:

* API abuse, which is a common way that attackers target microservices, is addressed in this dissertation, and by combining federated learning with anomaly detection, a flexible, decentralised security system that doesn't require all data to be stored in one place is created.
* Federated learning allows each microservice to learn and improve its security model independently, exchanging just updates rather than raw data. This approach is critical in today's environment, when maintaining data privacy and adherence to legislation are vital. It is especially useful when dealing with sensitive information in systems where data is distributed across several places. While this technique may be extended to a variety of microservice architectures, this dissertation focuses on Kubernetes because it is widely used and popular for managing microservices.
* Deep learning, a powerful tool for detecting anomalies, is used to boost security even more. These algorithms are trained to detect API abuse in microservices and consistently identify and respond to complex real-time threats.
* Prometheus and Grafana highlight the imperative of ongoing monitoring and rapid response in microservices security. The effectiveness of these systems in practical abuse detection scenarios is evaluated, providing valuable insights for security operations in similar contexts.

# Background

This dissertation focuses on the combination of federated learning, anomaly detection, and Kubernetes to construct a realistic system for detecting rate-limiting API abuse in microservices. Federated learning allows us to train detection models directly on the microservices themselves, reducing the necessity for transferring sensitive data across the network (Cross-silo and Cross-device Federated Learning on Google Cloud, 2024b). This method greatly minimises the possibility of data breaches while permitting the continued development of the models. Kubernetes offers the framework for implementing the federated learning models since its potential to handle dynamic workloads and scale resources corresponds nicely with the distributed nature of federated learning.

Each microservice acts as a client in the federated learning system, gathering data on API interactions, training its own model, and exchanging insights with a central aggregator (Hassan et al., 2021). This decentralised method not only guarantees privacy but also promises that we can rapidly detect rate-limiting API abuse across the entire system (Hassan et al., 2021).

The focus of this research is on detecting and eliminating rate-limiting abuse when attackers try to overwhelm APIs with excessive requests. By distributing the detection process, we can identify and respond to these attacks more effectively, protecting the system from overload and disruption. By combining the strengths of federated learning and Kubernetes, this dissertation aims to create a system that can effectively detect and respond to API abuse in microservices, improving security, resilience, and resource management.

## Theoretical Foundations

The theoretical foundations are rooted in several key areas: Federated Learning (FL), Anomaly Detection, Kubernetes and Microservices Architectures, and API Security. These theories and methods are integrated to address the challenge of detecting API abuse in a decentralized, dynamic, and secure manner within Kubernetes-based environments. With FL for decentralized model training, deep learning for anomaly detection, and Kubernetes for scalable service management, this dissertation aims to develop a robust and privacy-preserving solution for securing modern microservices-based applications.

### Federated Learning (FL)

The essential principle of Federated Learning is that each participant trains a model on its own dataset, which provides for data privacy and prevents leaking of sensitive information (Kaur et al., 2023). After local training, model updates are delivered to a central server, where they are combined to build a global model (Kaur et al., 2023). One of the primary issues in FL is maintaining effective communication between clients and the central server, especially when working with huge datasets or sophisticated models (Kaur et al., 2023). FL is crucial to this research since it gives a way for identifying API misuse without requiring centralizing sensitive data. By deploying anomaly detection models on each microservice in the Kubernetes cluster and utilising FL to aggregate model updates, the system can identify patterns of API misuse while protecting data privacy and lowering communication overhead.

A diagram of microservices

Description automatically generated

Figure 1: Federated Learning for Anomaly Detection

### Anomaly Detection

Anomaly detection is the technique of discovering data patterns that differ considerably from the norm (Anomaly Detection in Cybersecurity, n.d.). Anomaly detection models are used to find odd activity, such as unauthorized access attempts, abnormal API request patterns, or resource utilisation, which might signal possible security vulnerabilities including API abuse (Brown, 2024). Anomaly detection is a primary approach utilised in this dissertation to identify potential API misuse. The focus is on detecting abnormalities in API requests, such as surges in traffic, strange request patterns, or malicious activities that may not be detectable using typical rule-based security procedures. Deep learning-based anomaly detection models are created to capture these complicated patterns, and FL is used to distribute insights across the system.

### Kubernetes and Microservices Architecture

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and maintenance of containerised applications (*Kubernetes Documentation*, n.d.). Kubernetes clusters use microservices architecture, which divides applications into independent, loosely coupled services that communicate via APIs. Each microservice can be independently developed, deployed, and scaled (*Kubernetes Documentation*, n.d.). The dissertation utilises Kubernetes to manage a distributed microservices architecture. Kubernetes' capacity to handle large-scale, dynamic applications is crucial for establishing federated learning systems and managing microservices. Furthermore, the ability to monitor and expand services within Kubernetes enables the infrastructure necessary to allow real-time anomaly detection for API abuse.

### API Security and API Abuse Detection

APIs are the main mechanism of communication for microservices in current designs. Consequently, securing APIs from abuse, including unauthorised access, denial-of-service attacks, and misuse, is critical (*What Is API / Microservice Security? A Guide from PortSwigger*, n.d.). API abuse detection comprises monitoring API traffic for unusual or malicious activities that may imply a probable attack or misuse (*What Is API / Microservice Security? A Guide from PortSwigger*, n.d.). The major goal of this dissertation is to identify and minimise API misuse in a Kubernetes-based microservices architecture. The dissertation attempts to design a system capable of automatically detecting anomalous API use patterns that may signal abuse, hence improving the system's overall security and dependability.

## Historical Development

The evolution of Federated Learning (FL) and Kubernetes has been influenced by the growing need for distributed systems, privacy-preserving machine learning, and efficient container orchestration in modern cloud-native applications. This section briefly traces their historical development, highlighting key milestones that shaped their relevance to microservices architectures.

### Federated Learning (FL)

FL’s evolution has made it a natural fit for decentralized systems, particularly those with distributed data sources like microservices in Kubernetes clusters (*Decentralized Federated Learning: A Survey and Perspective*, n.d.). By allowing models to be trained without moving sensitive data, FL supports the creation of secure and scalable anomaly detection systems for API abuse in cloud-native applications (*Decentralized Federated Learning: A Survey and Perspective*, n.d.).

Table 1.1: Key Milestones in the Development of Federated Learning (Federated Learning History Overview | Restackio, n.d.)

|  |  |  |
| --- | --- | --- |
| **Year/Period** | **Milestone** | **Description** |
| 2016 | Google’s Federated Learning Proposal | Google introduced "federated learning" to enable decentralized machine learning, allowing model training on devices without sharing sensitive data. |
| 2017–2019 | Expanding FL Applications | FL gained traction, especially in healthcare and finance, due to its promise of data privacy and collaborative model training. |
| 2020 | Federated Learning in Production | Organizations adopted FL in production, especially for data-sensitive industries. Integration with cloud platforms enhanced scalability and application scope. |

### Microservices

The evolution of microservices deployment reflects the shift from monolithic architectures to highly modular, scalable systems supported by containerization and orchestration tools (*IBM Developer*, n.d.). Initially, applications were developed as monolithic units, meaning all components were tightly integrated into a single codebase, which made scaling and updating individual components difficult (*IBM Developer*, n.d.). The introduction of Docker containers in 2013 allowed applications to be broken into smaller, isolated services, making deployments more consistent and efficient. However, managing multiple containers manually was complex, leading to the development of Kubernetes in 2014. This enabled the seamless deployment of microservices architectures, where each microservice could be independently developed, deployed, and scaled, enhancing flexibility, fault tolerance, and the speed of innovation in application development.

A diagram of a number of microprocessors

Description automatically generated

Figure 1.2: Timeline of the evolution of microservices (Partners, 2024)

# Literature Review

The increasing adoption of microservices architectures and the growing reliance on cloud-native environments have resulted in the emergence of numerous challenges related to security and resource management. One of the most significant concerns is the detection of API abuse and anomalous behaviour across microservices. This section reviews the existing literature on anomaly detection in microservices environments and federated learning applications, emphasizing their application to cloud-native environments and API abuse detection.

## Federated Learning in Cloud-Based environments

Mehta and Aneja (2024) in their research titled “***Privacy-Preserving AI: Leveraging Federated Reinforcement Learning in Distributed Systems***” investigate the confluence of federated learning and reinforcement learning, offering a framework called Federated Reinforcement Learning (FRL) to overcome challenges in distributed systems, especially in ensuring data privacy. This work coincides with my dissertation's objective of utilising federated learning for enhanced security in microservices. While my work focused on anomaly detection for API abuse prevention, Mehta and Aneja's analysis of FRL offers important insights into the larger applications and benefits of federated learning in remote systems. Their findings, especially the observed advantages in speed and communication efficiency compared to centralized approaches, emphasise the potential of federated learning to overcome security concerns in distributed systems like Kubernetes while protecting data privacy. This further strengthens my proposal to employ federated learning as a major component in building a solid and privacy-preserving solution for API abuse detection in microservices.

Hassan et al. (2021), in their paper published "***FedEdge: Federated Learning with Docker and Kubernetes for Scalable and Efficient Edge Computing***," offer FedEdge, a platform that combines Docker and Kubernetes to support federated learning in edge computing scenarios. This work supports my research by showcasing the practical implementation and advantages of mixing federated learning with Kubernetes. FedEdge addresses significant obstacles including scalability, resource efficiency, and portability, which are critical factors for implementing my proposed API abuse detection system in a Kubernetes cluster. Their focus on utilising Docker containers for effective resource allocation and Kubernetes for orchestration coincides with my approach to assure the scalability and flexibility of my solution within a dynamic microservices scenario. By demonstrating the feasibility and advantages of this integrated strategy, Hassan et al.'s work delivers important support for my selected technology stack and enhances the basis for my dissertation's study on federated learning for API abuse detection in Kubernetes.

Parra-Ullauri et al.'s (2024) study, "***kubeFlower: A privacy-preserving framework for Kubernetes-based federated learning in cloud-edge environments****,*" addresses the fundamental challenge of ensuring privacy in Kubernetes-based federated learning systems. This study closely ties to my dissertation by exposing the possible privacy issues inherent in Kubernetes' design and offering a solution, KubeFlower, to alleviate these risks. Their emphasis on isolation-by-design and differential privacy strategies for safe data management matches with my purpose of maintaining data privacy in my proposed federated learning system for API abuse detection. Specifically, their research on the Privacy Preserving Persistent Volume Claimer (P3-VC) gives useful insights into how to include differential privacy methods inside the Kubernetes environment.

In their work "***Proactive auto-scaling technique for web applications in container-based edge computing using federated learning model***," Dogani and Khunjush (2024) provide FedAvg-BiGRU, a novel solution for proactive auto-scaling in Kubernetes using federated learning. This research aligns with my dissertation by exploring the idea of federated learning to better resource management and efficiency in Kubernetes, particularly in the context of fluctuating workloads and resource demands. Their emphasis on lowering data transmission between edge nodes and the cloud server via the usage of FedAvg directly supports my objective of minimizing communication overhead and ensuring data privacy in my proposed federated learning system for API abuse detection. While their work focusses on auto-scaling, the underlying ideas of employing federated learning for distributed decision-making in Kubernetes apply to my study. Their results, which show how well FedAvg-BiGRU reduces communication while preserving prediction accuracy, lend credibility to the concept that federated learning can enhance several Kubernetes management features, such as resource optimisation and security, both of which are important to my dissertation.

## Anomaly Detection in Microservices

In "***Real-Time Automated Anomaly Detection in Microservices Using Advanced AI/ML Techniques***," Parida et al. (2023) conduct a comprehensive evaluation of AI/ML applications for microservice anomaly detection. This study is significant to my dissertation as it digs into the greater issue of anomaly detection methodologies in microservices, including supervised, unsupervised, and semi-supervised learning algorithms that may discover a range of anomalies, including abuse of API rate limitations. Their emphasis on real-time data processing frameworks like Apache Kafka and Apache Flink coincides with my purpose of developing a system that can respond to threats in real time. Parida et al.'s analysis of hybrid models and ensemble techniques offers insight on various ways for boosting the robustness and performance of my recommended anomaly detection system. This article offers a thorough overview of the state of the art in AI/ML-driven anomaly detection, which makes it a great resource for comprehending the broader context of my dissertation in creating a reliable and efficient solution for microservices API abuse detection.

In his paper "***Machine Learning Models for Anomaly Detection in Microservices***," Ramamoorthi (2023) examines numerous AI-driven methodologies for anomaly detection in microservice architectures. Their analysis of Random Forest, Support Vector Machines (SVM), Autoencoders, and Isolation Forest provides critical information about each model's advantages and disadvantages in detecting performance anomalies in metrics relevant to my research, such as CPU utilisation, memory utilisation, and network I/O. Furthermore, his study of time-series forecasting systems like as ARIMA, LSTM, and Prophet provides a solid foundation for my research into deep learning-based anomaly detection, particularly with LSTM networks. His results, which demonstrate the efficacy of LSTM and Random Forest in achieving high accuracy and recall rates, support my approach to developing a robust API rate-limiting abuse detection system. This research lays the groundwork for my dissertation's investigation of deep learning-based anomaly detection for API abuse prevention by doing a comparative analysis of several machine learning models and demonstrating their value in detecting irregularities in microservices.

Nobre et al. (2023) evaluate the significance of a Multi-Layer Perceptron (MLP) neural network for anomaly detection in microservices in their study, "***Anomaly Detection in Microservice-Based Systems***." Their work is related to my dissertation in that it concentrates on the application of machine learning to find abnormalities in distributed systems, specifically in a microservices architecture. Despite the fact that my study emphasises deep and federated learning, their examination of a supervised MLP model delivers information on the feasibility and utility of neural networks for anomaly identification in this context. Their findings support the use of deep learning techniques to detect API rate-limiting abuse, which frequently appears as service-level performance deviations, demonstrating the MLP's effectiveness in detecting both application-level and service-level anomalies, particularly with higher accuracy on service-level metrics such as response times.

In their research, "***AI-Powered Anomaly Detection for Kubernetes Security: A Systematic Approach to Identifying Threats***," Bhardwaj et al. (2024) explore the application of AI for threat detection in Kubernetes, focusing on identifying odd activities. This corresponds with my research by emphasising the need for AI and anomaly detection in securing Kubernetes systems. Their rigorous approach, which includes data collection, model training, and integration with Kubernetes, provides a useful foundation for my research. While their research covers a broader range of security threats, their findings, which demonstrate AI's effectiveness in identifying over 92% of simulated threats and reducing response times by 67%, highlight the potential of AI-powered anomaly detection, which I use specifically to detect API rate-limiting abuse. Their explanations of challenges, such as data quality and model complexity, give helpful insights for my study. By illustrating the actual advantages and problems of integrating AI-driven threat detection in Kubernetes, this study supports the relevance of my research and serves as an invaluable resource for developing and evaluating my proposed solution.

## Kubernetes Orchestration and Microservice Architecture

"***Beyond Containers: Orchestrating Microservices with Minikube, Kubernetes, Docker, and Compose for Seamless Deployment and Scalability,***" a paper by Eyvazov et al. (2024), proposes a methodology of instructions for coordinating microservices using a variety of technologies, such as Minikube and Docker. Since it focuses on the deployment and scalability of microservices in a Kubernetes environment—a critical component for the real-world implementation and evaluation of my suggested federated learning system—this directly links to my research. Their analysis of Minikube, a tool for locally running Kubernetes, gives intriguing information that might help me set up a development environment and conduct testing for my study. Furthermore, my technique for deploying and managing the many microservices that will take part in the federated learning process is compatible with their description of Docker and Docker Compose for containerisation and multicontainer management. Eyvazov et al.'s work provides significant direction for the implementation and assessment stages of my dissertation by offering real-world examples and emphasising the benefits of these tools for smooth deployment and scalability.

Santos et al. (2024) examine the complexities of efficient microservice deployment in multi-cluster Kubernetes environments and provide a Reinforcement Learning (RL)-based solution in their paper titled "***Efficient Microservice Deployment in Kubernetes Multi-Clusters through Reinforcement Learning.***" My dissertation utilises federated learning for anomaly detection, whereas their research elucidates enhancements in resource allocation and deployment strategies inside Kubernetes, which is crucial for the effective execution of my suggested system. Their emphasis on multi-cluster scenarios and the difficulty of managing microservices across several clusters highlights the need of managing scalability and efficiency in my research. Furthermore, their effective use of RL agents to achieve near-optimal allocation techniques, with a focus on latency reduction and cost minimisation, suggests a promising direction for future research into enhancing the deployment and scalability of my federated learning system. Their usage of the DeepSets neural network for generalizing across multiple multi-cluster environments without retraining is especially intriguing as it provides possible ways for adapting my anomaly detection system to different Kubernetes environments. By addressing the issues of effective microservice deployment in multi-cluster Kubernetes, Santos et al.'s work offers essential background and possible optimization tactics for the actual execution of my dissertation's proposed solution.

## API Abuse Detection in Microservices

The study "***API Traffic Anomaly Detection in Microservice Architecture***" by Sowmya et al. (2024) focuses on API Traffic Anomaly Detection (API-TAD) in microservices designs and gives a machine learning-based method for detecting anomalies at both the general and application-specific levels. Their focus on scrutinising API data for unusual activity aligns with my work, which uses deep learning models to detect patterns indicative of fraudulent requests to the API. Their technique for identifying abnormalities at various levels delivers vital insights for constructing a comprehensive solution capable of distinguishing both widespread and application-specific attack types. Their use of machine learning techniques, such as bagging and RandomForest, provides significant background for my research into deep learning models for anomaly identification. Sowmya et al.'s research, which emphasises the viability and value of machine learning in detecting API anomalies in microservices, greatly supports my dissertation's investigation on deep learning-based anomaly detection for API rate limitation abuse.

### API Abuse Detection and Rate-Limiting Violations

Kim et al. (2020), in “Detecting API Abuse with Machine Learning and Rate-Limiting Techniques”, explore various machine learning models for detecting API abuse, specifically rate-limiting violations. They propose using statistical anomaly detection models that can identify unusual traffic patterns in API usage, including sudden spikes or excessive request rates. This research provides insights into detecting API abuse through machine learning-based anomaly detection.

Wang et al. (2019), in “A Framework for Real-Time API Abuse Detection in Microservices Architectures,” present a framework for detecting API misuse in microservices through real-time monitoring. They focus on using both rate-limiting techniques and anomaly detection algorithms to flag abusive patterns in API usage.

Badr et al. (2020), in “Federated Learning for Secure and Private API Communication in Cloud Systems”, propose a federated learning-based framework for ensuring secure API communication in cloud environments. Their model utilizes differential privacy techniques to prevent the leakage of sensitive information while still providing high levels of accuracy in anomaly detection.

### Anomaly Detection in Microservices

Singh et al. (2020), in their work “Federated Learning for Anomaly Detection in Distributed Systems,” explore the use of federated learning for anomaly detection in data streams, specifically using autoencoders across federated clients. Their study demonstrates how federated learning can detect anomalies without centralizing data, providing a strong basis for applying it to distributed systems.