future internet

MDPI

Review

Federated Learning for Intrusion Detection Systems in Internet of Vehicles: A General Taxonomy, Applications, and Future Directions

Jadil Alsamiri and Khalid Alsubhi *

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; jalsamiri@stu.kau.edu.sa

* Correspondence: kalsubhi@kau.edu.sa

Abstract: In recent years, the Internet of Vehicles (IoV) has garnered significant attention from researchers and automotive industry professionals due to its expanding range of applications and services aimed at enhancing road safety and driver/passenger comfort. However, the massive amount of data spread across this network makes securing it challenging. The IoV network generates, collects, and processes vast amounts of valuable and sensitive data that intruders can manipulate. An intrusion detection system (IDS) is the most typical method to protect such networks. An IDS monitors activity on the road to detect any sign of a security threat and generates an alert if a security anomaly is detected. Applying machine learning methods to large datasets helps detect anomalies, which can be utilized to discover potential intrusions. However, traditional centralized learning algorithms require gathering data from end devices and centralizing it for training on a single device. Vehicle makers and owners may not readily share the sensitive data necessary for training the models. Granting a single device access to enormous volumes of personal information raises significant privacy concerns, as any system-related problems could result in massive data leaks. To alleviate these problems, more secure options, such as Federated Learning (FL), must be explored. A decentralized machine learning technique, FL allows model training on client devices while maintaining user data privacy. Although FL for IDS has made significant progress, to our knowledge, there has been no comprehensive survey specifically dedicated to exploring the applications of FL for IDS in the IoV environment, similar to successful systems research in deep learning. To address this gap, we undertake a well-organized literature review on IDSs based on FL in an IoV environment. We introduce a general taxonomy to describe the FL systems to ensure a coherent structure and guide future research. Additionally, we identify the relevant state of the art in FL-based intrusion detection within the IoV domain, covering the years from FL's inception in 2016 through 2023. Finally, we identify challenges and future research directions based on the existing literature.

Keywords: Federated Learning (FL); intrusion detection systems (IDS); Internet of Vehicles (IoV); deep learning; machine learning

Citation: Alsamiri, J.; Alsubhi, K. Federated Learning Based Intrusion Detection Systems in Internet of Vehicles: A Literature Survey. *Future Internet* **2023**, *15*, 403. https://doi.org/10.3390/fi15120403

Academic Editors: Qiang Duan and Zhihui Lu

Received: 30 October 2023
Revised: 9 December 2023
Accepted: 12 December 2023
Published: 14 December 2023

1. Introduction

The rapid expansion of the Internet of Things (IoT) has led to a number of novel applications, such as smart cities, smart grids, and the Internet of Vehicles (IoV). When these smart objects take the form of interconnected vehicles over the internet, the IoT becomes the IoV. Significant interest in IoV technologies has emerged due to substantial advancements in the smart automobile industry. IoV networks are integrated and open network systems that connect vehicles, human intelligence, neighboring environments, and public networks. These networks aim to increase road safety, reduce human error-related accidents, and mitigate congestion. This is accomplished by continuously monitoring traffic congestion. However, despite the numerous benefits offered by the IoV, several issues must be addressed to safeguard the lives of all road users. The IoV is vulnerable to

cyberattacks, which threaten its stability, robustness, and can lead to vehicle unavailability and traffic accidents. Since communication in these networks requires the involvement of multiple components, they are susceptible to a broad array of attacks. Thus, ensuring their security requires advanced intrusion detection systems (IDSs) that can address potential cyberattacks. IDSs excel at identifying anomalies and attacks in the network's data during communications between vehicles and various devices. Given that the IoV is a relatively new network paradigm, new and ever-evolving attacks against it continue to emerge. The IoV network creates a huge amount of data very quickly, especially when there are cyberattacks.The accuracy of machine learning and deep learning approaches makes them a preferred choice in this high-stakes environment [1]. Nevertheless, the need to store and transmit data to a centralized server may compromise privacy and security. In contrast, Federated Learning (FL), a decentralized learning approach that protects privacy, trains models locally before sending only the parameters to the centralized server. Even though FL for IDS development has made significant progress, a comprehensive survey specifically exploring the applications of FL for IDS in the IoV environment has yet to be conducted. To the best of our knowledge, a gap exists in the availability of a study that comprehensively assesses current IDSs based on FL for IoV, similar to the successful systems research conducted in deep learning.

To address this gap, the key contributions of our survey can be summarized as follows:

- We offer a generic taxonomy for describing FL systems (FLSs) to ensure a coherent structure and guide future research.
- We undertake a well-organized literature review on IDSs based on FL in an IoV environment. This review identifies the latest advancements in FL-based intrusion detection within the IoV domain, covering the years from FL's inception in 2016 to 2023.
- Furthermore, we highlight several challenges and potential future directions based on the existing literature.

The remainder of the paper is organized as follows. Section 2 explores the background within this domain, covering IoV, FLSs, and IDSs. Section 3 aims to provide a thorough overview of FL research within the context of IDSs in IoV environments. Finally, we conclude the paper by describing open research challenges and outlining possible future research directions in Section 4. For increased clarity and understanding, abbreviations section summarizes the abbreviations used in this manuscript.

2. Background

2.1. An Overview of Internet of Vehicles

Transportation has become a significant challenge in many countries due to population growth. Often, the transportation system itself is outdated, making upgrading a costly and daunting task. By 2035, the number of vehicles around the globe is estimated to reach two billion. This substantial number will strain existing transportation systems and most likely result in more accidents and traffic jams. Therefore, changes must be made in the transportation system's framework to adjust to emerging prerequisites of new vehicles, travelers, and drivers [2]. Technological advancements have motivated the enhancement of a wide array of gadgets to be used in various fields, including IoT. Additionally, the Internet is helping societies develop much faster, and people in developed societies, in turn, are seeking a better way of life [3]. A few of these technologies have resulted in the further advancement of IoV, a field commonly considered an extension of IoT. IoT is a universal network of interconnected smart devices equipped with embedded hardware and software for environmental sensing and data exchange, with the capability to act on that information. Therefore, including vehicles as devices makes IoV a field with applications in intelligent transportation, crash prevention, and smart cities [4]. IoV networks require software applications to monitor vehicle movements and provide security against malicious attacks. These systems function through interactions with various components,

including vehicle communication with roads, roadside units, and sensors [5]. IoV brings together two cutting-edge dreams—the network and intelligent vehicles—while centering around the objects (e.g., humans, vehicles, systems) to create a perceptive system that relies on information technology and communication features to assist authorities in huge urban territories and entire countries [3]. IoV enables extensive communication between vehicles in various forms, including vehicle-to-vehicle, vehicle-to-road, vehicle-to-human, vehicle-to-infrastructure, and vehicle-to-sensor connections through wireless communication technologies [6]. Additionally, human-to-human interaction occurs in IoV. Generally, though, the human component is gaining importance as the services develop. In their research, Rim et al. [7] view IoV as a worldwide network with three integrated subnets: the intravehicle network, the intervehicle network, and the vehicular mobile internet. By contrast, Garg et al. [3] define IoV from the angle of integration of on-board sensors and communication technologies. These researchers view IoV as intelligent vehicles with advanced devices that utilize modern communication and networking technology to provide vehicles with complex environment sensors, intelligent decision making, and control functions.

2.1.1. Benefits of Internet of Vehicles

IoV has the potential to transform the transportation industry's landscape, making travel safer, more efficient, and friendlier to the environment. The IoV provides several opportunities for improvement and numerous benefits, including the following [3]:

- Lower costs: Improved traffic control results in lower costs, including insurance premiums and operational costs.
- Time efficiency: Traffic is meticulously monitored, examining the time people spend on the road.
- Reduced risk of fatalities: Examining the transportation environment can reduce accidents, such as by helping drivers navigate traffic [8].
- Smart cities development: Smart cities are more organized due to the services they provide, including enhanced navigation and real-time traffic.
- Greenhouse effect reduction: This limits harm to the world.
- Emergency response: IoV can autonomously notify emergency services in the case of an accident, potentially diminishing reaction times and saving human lives.
- Autonomous driving: IoV is an essential part of the development of autonomous and semi-autonomous vehicles, both of which can lower the number of accidents resulting from human mistakes and enhance general road safety.
- Traffic documentation: Filming traffic accidents using services such as pics-on-wheels allows any vehicle on the road to act as a witness to any accident. Among other outcomes, this encourages people to maintain decorum on the road.

In general, IoV offers the potential for safer, more intelligent, and more efficient mobility for individuals and society as a whole.

2.1.2. **Internet of Vehicles' Characteristics and Challenges**

This section elaborates on the characteristics of IoV and discusses various challenges that IoV faces. Compared to other types of networks, IoV networks are distinguished by several qualities. IoV is an evolution of traditional vehicular ad hoc networks (VANETs) and shares many characteristics with VANETs, including dynamic topology, fluctuating network density, high vehicular mobility, and network obstacles [2]. However, IoV networks possess the following additional attributes:

- Scalability: Compared to traditional VANETs, IoV networks have the capacity to incorporate a significantly larger number of interconnected vehicles, ranging from hundreds to thousands. Furthermore, IoV has the potential to significantly augment the number of interlinked gadgets to a magnitude of millions, depending on the utilized application.

- Multiple wireless access methods: The IoV platform supports several types of wireless access methods, including WLANs, WiMAX, cellular wireless, and satellite communications.
- Extended network communication: IoV enables a broader range of communication options than conventional VANETs, characterized by their restricted communication capabilities. IoV facilitates vehicle-to-smart object connection, including devices such as smartphones and tablets.
- Cloud computing: Unlike VANETs, the activities in IoV mostly rely on cloud computing services.
- Predictable mobility: Vehicular networks differ significantly from other ad-hoc network types because vehicles often move quickly and in any direction. Vehicles are predictable in their movement due to the topography, roadway layout, use of signal-received traffic lights, and consideration of other moving vehicles' distance. Therefore, vehicles are predicted to possess integrated GPS systems to ascertain information on their movement.
- Highly dynamic topology: A vehicle network's topology exhibits a high degree of dynamism, characterized by intermittent and rapid changes. Hence, the intricate network topology dynamics must be thoroughly analyzed to advance the IoV environment. IoV encompasses a collection of vehicles that exhibit regular variations in both their velocity and trajectory. As a result, the configuration of the moving vehicles' topology likewise undergoes alteration. Therefore, IoV supports a highly dynamic topology, and the routing protocols are designed to consider this [9].

The IoV encounters a multitude of issues that require thorough investigation to enhance communication dependability, robustness, and steadiness, including the following:

- Fault tolerance: Because the IoV design is built on cloud connections, some vehicles could malfunction; nevertheless, these failures should not influence the functioning of the remainder of the network.
- Latency: The term "latency" refers to the amount of time that passes while a packet is transferred through a network. Latency must be reduced as much as possible in some mission-critical applications, such as accident warnings, to ensure that messages are transmitted quickly.
- Network compatibility: To develop applications and protocols for IoV, researchers must consider the numerous access technologies supported by IoV. This ensures that the networks they create are compatible and allows IoV to function with the various access technologies available today.
- Security: The data shared over the IoV network is sensitive and private, which is especially important given that users can access the internet. As a result, the process of protecting these networks is an essential undertaking and a prerequisite for the implementation of IoV.
- Connectivity: The rapid movement of vehicles can result in frequent fluctuations in network architecture, impacting connectivity. As a result, a significant portion of the rate at which nodes arrive and leave can be influenced. The need to contend with such a restriction depletes an essential amount of communication overhead. Thus, nodes must often choose a trustworthy route to ensure that data is delivered to specific destinations to function correctly. The vehicles must be continuously linked to one another.

2.1.3. IoV Network Requirements and Generic Architecture

The Internet of Vehicles (IoV) is a transformative advancement in the realm of vehicular communications, merging traditional vehicular networks with cutting-edge information and communication technologies. This integration not only expands vehicular capabilities but also introduces intricate challenges and requirements in security, privacy, and functionality. Understanding the architecture and requirements of IoV networks is pivotal for developing sophisticated solutions like Federated Learning (FL)-based Intrusion Detection Systems (IDS). In this subsection we provide a summary analysis of the essential security,

privacy, and functional requirements of IoV networks, alongside a detailed description of a generic IoV network architecture. Figure 1 shows the essential IoV network requirements.

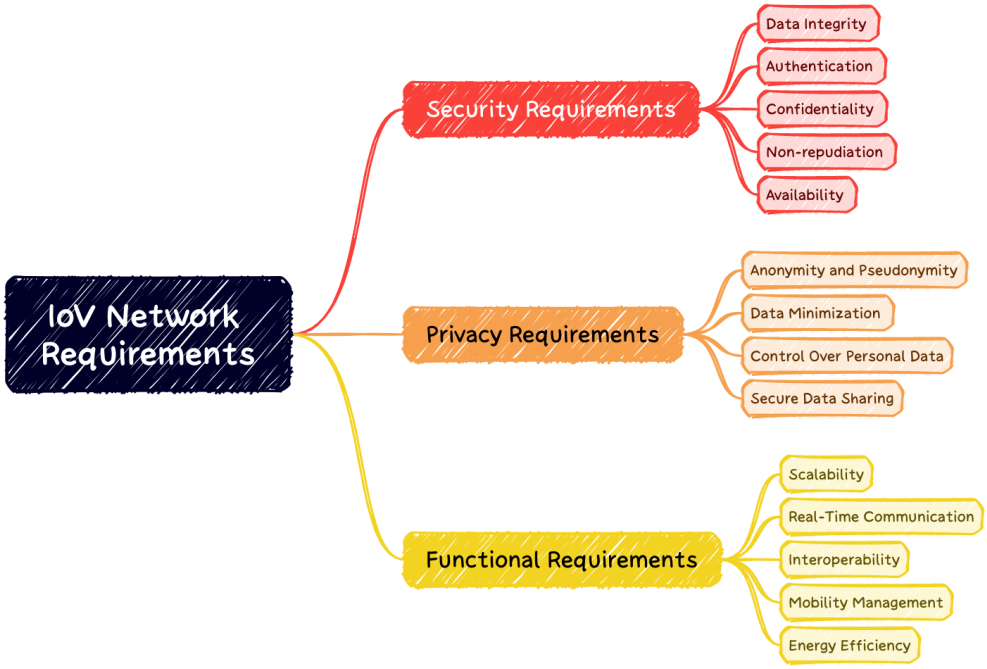

Figure 1. IoV Network Requirements.

Security Requirements in IoV Networks

Security within Internet of Vehicles (IoV) networks is uniquely complex, given the dynamic and mobile nature of vehicular communications [10]. Here, data integrity must go beyond standard concerns—it is critical for safe vehicular operation as vehicles rely on accurate, real-time shared information for essential functions. Any unauthorized data manipulation can lead to immediate safety risks. Authentication in IoV networks is also more challenging than in static networks. It is not just about securing data, but about reliably verifying the rapidly changing participants in the network—vehicles, road infrastructure, and other connected entities—to prevent malicious activities [11]. The confidentiality of data in IoV systems carries additional weight. Protecting user privacy, like location and travel habits, is not only about privacy rights but also about safeguarding against potential threats that could exploit this sensitive data for harmful purposes. Non-repudiation, while important in many digital systems, takes on heightened significance in IoV. Here, it is crucial for legal and liability reasons, ensuring that a vehicle or network component cannot deny its actions, especially in incident analysis and forensic investigations following accidents or security breaches. Lastly, the aspect of continuous availability in IoV networks is paramount. The challenge is to maintain seamless service in a mobile, high-speed environment, where Denial of Service (DoS) attacks or other disruptions not only compromise data but can directly impact physical safety and traffic efficiency.

Privacy Requirements in IoV Networks

Privacy concerns in Internet of Vehicles (IoV) networks are especially pronounced due to the continuous and detailed data generation by vehicles. Protecting user identities and sensitive data here goes beyond typical privacy considerations. Users in an IoV context should have options for anonymity or pseudonymity [3], crucial for preventing the real-time tracking of their vehicles, which could lead to physical tracking in the real world.

The principle of data minimization becomes even more critical in IoV environments. Here, the vast amount of data generated by vehicles, including location, travel routes, and driving patterns, must be carefully managed. Collecting only the necessary data

for intended functionalities not only preserves privacy but also reduces the risk of data breaches with potentially severe real-world consequences. User control over data in IoV networks is vital. Given the diverse sources of data collection and dissemination in IoV—from traffic management systems to third-party service providers—users must have clear and manageable controls over who accesses their data and for what purpose. This aspect is particularly challenging in IoV due to the interconnected nature of vehicular networks and the range of stakeholders involved [12]. Moreover, when data sharing is necessary for the functionality of IoV services, its execution requires robust security measures. It is essential to ensure that sensitive information, such as real-time location or travel behavior, is accessible only to authorized entities [4]. This protection is crucial in preventing the potential misuse of data, which could lead to privacy infringements or even safety hazards.

Functional Requirements in IoV Networks

The functionality of Internet of Vehicles (IoV) networks is not just about enabling vehicular communication; it is about doing so in a way that meets the unique demands of a highly mobile and rapidly evolving vehicular environment. Scalability is more than a feature here; it is a necessity. The IoV network must seamlessly integrate an ever-growing number of vehicles and infrastructure elements, each adding to the complexity and volume of data exchange [13]. Real-time communication in IoV networks is about more than just speed; it is about life-critical decisions. Low latency is indispensable for enabling timely reactions in dynamic driving scenarios, where milliseconds can mean the difference between safety and danger. Interoperability in IoV extends beyond standard tech compatibility. It involves harmonizing a myriad of vehicle models, diverse infrastructural technologies, and varied network protocols to ensure uninterrupted communication, a task that is significantly more complex given the varying standards and technologies in the automotive sector. Effective mobility management in IoV is not just about maintaining network connections; it is about doing so in a context where vehicles are constantly moving at high speeds, often transitioning between different network zones, which requires sophisticated handover mechanisms and robust connectivity management [14]. Furthermore, optimizing energy usage, especially in the realm of electric vehicles, goes beyond conventional energy management concerns. In IoV, this is critical for the sustainable operation of not just individual vehicles, but the entire network, impacting everything from data transmission efficiency to the overall environmental footprint of the vehicular ecosystem.

Generic Architecture of IoV Networks

The Internet of Vehicles (IoV) is an advanced network architecture that integrates vehicular technology with information and communication systems to enhance road safety, traffic efficiency, and driving experiences. The core components of IoV architecture include [15]:

- Vehicles: The primary entities in IoV are the vehicles themselves, equipped with sensors, communication modules, and computing capabilities. These vehicles can collect and share a vast array of data, including speed, location, traffic conditions, and environmental data.
- Roadside Units (RSUs): These are fixed infrastructural components placed alongside roads. RSUs facilitate communication between vehicles and the broader network infrastructure, acting as access points for data transmission and reception [5].
- Central Servers: Central servers provide backend support for data processing, storage, and advanced computational tasks. They play a critical role in managing the overall network, including traffic control, data aggregation, and system updates.
- Communication Network: This includes both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, enabled through technologies like Dedicated Short Range Communications (DSRC) and cellular networks. The network ensures seamless and continuous connectivity within the IoV.

- Traffic Management Center (TMC): The TMC acts as the control hub for traffic management, receiving data from various sources and making decisions to optimize traffic flow, reduce congestion, and enhance road safety [5].
- Cloud and Edge Computing Resources: Cloud computing provides vast storage and processing capabilities, essential for handling the large volumes of data generated in IoV. Edge computing, on the other hand, offers localized processing at the network edge, enabling real-time data processing and decision-making.

This architecture fosters an interconnected environment essential for various applications, including IDS. The architecture's distributed nature, real-time communication capabilities, and integration of advanced computing technologies are vital for implementing effective FL-based IDS.

2.1.4. Security in Internet of Vehicles

IoV technologies are developing rapidly, and a number of industries investing in these technologies are in a race to launch state-of-the-art self-driving vehicles. These rapid advancements in IoV result in security issues that threaten not only the industry but consumers as well [8]. The challenge lies in preventing security breaches and privacy violations in IoV, making it less susceptible to cyberattacks [3].

Cyberattacks in Internet of Vehicles Networks

Vehicular sensor networks comprise a variety of vehicle sensors used to monitor and measure various physical parameters associated with the vehicle and the environment in which it is located. These sensors contribute to a more comfortable driving experience and smoother driving operations. Table 1 lists some of the most frequently employed smart vehicle sensors. Each of these sensors is built using cutting-edge electronic components and communication systems. Due to their limited available resources, implementing sophisticated and reliable security algorithms on these sensors directly is impractical. Consequently, these sensors are susceptible to various cyberattacks [16].

IoVs are susceptible to several types of attacks and threats, including the following:

- The flow of bogus information: Attackers use fake information to make users believe in a false environment.
- Message injection attack: Attackers send seemingly legitimate messages to gain access to one or more entities, which they can also utilize to send out malicious messages [16].
- Replay attack: Attackers iterate messages to gain unlawful access to the network's services and resources [17].
- Cookie theft attack: Resembling the previous attack, attackers use a copy of the cookies they stole to reach the network's resources.
- Sybil attack: Attackers create fabricated vehicles around the vehicle they are targeting and generate a signal jam, compelling the target to use an alternate path. To do this, they use a countless number of fake IDs for a single node to create the appearance of multiple nodes [18].
- Man-in-middle attack: Attackers insert themselves between two communicating entities. In this type of attack, which can be active or passive, the attackers can receive messages from one entity and send them to the other [16].
- Denial-of-service and distributed denial-of-service attacks: Attackers attempt to disrupt the network's efficiency by flooding the target channel with messages that exceed its handling capacity. This is carried out to use the network's limited resources illegally [10].
- Dissimulation of GPS attack: Attackers intercept and modulate GPS signals before the intended receiver receives them. This type of attack can endanger the lives of the people in the target vehicle as they are given the wrong directions.
- Impersonation attack: As the name implies, attackers impersonate the identity of a legitimate user on the network to spoof unsuspecting vehicles on the network with messages that are not only fictitious but dangerous.

- Masquerading attack: Again, as the name implies, attackers masquerade as authorized users. Unlike the previous attack, attackers copy the legitimate ID of one of the network's nodes and can create two different senders using the same identity.
- Wormhole attack: Attacker nodes fake incorrect information about the distance from the target node, aiming to obtain every message sent from the receiver to flow through it. Deadlocks are typically created by these types of attacks [17].
- Eavesdropping attack: Attackers passively listen to the communication on the network. They become a part of the network, aiming to secretly obtain confidential, sensitive data and use it unlawfully.

Table 1. Common sensors in vehicle sensor networks.

Sensor	Use of Sensors in Vehicles
Camera	Identifies traffic signs, enhances night vision, adapts to the light system, determines the likelihood of being involved in a collision, detects lanes, records emergencies, and provides parking aid.
GPS	Tracks location, provides path direction, minimizes fuel costs, lowers operational costs, helps with theft recovery and in an emergency.
Ultrasonic sensors	Include parking assist systems, which monitor the immediate surroundings of the vehicle and measure distance to obstacles.
LiDAR	Ensures safe navigation by detecting objects and estimating distances.
Radar	Detects obstacles or pedestrians, deploys automatic emergency braking, and enables blind-spot monitoring, lane-keeping assistance, and parking assistance in autonomous mode.
Inertial sensors	Provide data concerning the rate of acceleration and the current direction of the vehicle, includes automotive safety systems like airbag and anti-skidding protection.
Tire pressure monitoring system	Monitors tire air pressure and alerts the driver when it falls dangerously low.

2.2. An Overview of Federated Learning

As the risk of a data breach grows increasingly significant, many governments are enacting legislation to protect their citizens' data. Because of a breach that occurred in 2016 involving the personal information of 600,000 drivers, Uber was forced to pay USD 148 million to resolve the investigation [19]. In response to these situations, Google introduced the notion of FL to facilitate on-device learning while ensuring the preservation of data privacy. FL enables collaborative learning among devices without necessitating data sharing with a centralized server. In other words, machine learning and deep learning may be trained across various devices and servers using decentralized data thanks to the capabilities of the technology [20]. This process can be repeated multiple times. This section provides an overview of FL, introducing the concept and highlighting its potential applications and benefits in several domains.

2.2.1. Definition of Federated Learning

FL facilitates the collaborative training of a machine learning model by many parties without the need for the direct exchange of their respective local data. The subject matter encompasses a range of methodologies derived from various fields of research, including distributed systems, machine learning, and privacy. Building on the definitions of FL provided by previous studies [19,21,22] we propose the following definition for FL. In an FL framework, numerous entities work together to train machine learning models without

the need to share their raw data. The result of the process is a machine learning model for each entity involved (which may be identical or distinct). A crucial restriction of a practical FLS is that the performance of the model acquired through FL should surpass that of a model obtained through local training when evaluated using a designated measure, such as test accuracy, using the same model architecture. FLSs include the following aspects:

- Data privacy: An FLS tackles the issue of data privacy by enabling individual entities to maintain their data locally, hence avoiding the need to share it with a centralized server. This is especially crucial when handling private or sensitive data [23].
- Collaborative training: Models are trained collaboratively within the FLS. Based on its local data, each party or device independently computes updates to the model and shares them with other participants or a central server.
- Aggregated model: By combining the model updates from each participant, the central server creates an enhanced global model that gains from everyone's combined expertise. The participants then receive a copy of this combined model.
- Iterative process: The iterative nature of the FLS entails the incorporation of several training rounds. During each iteration, individuals involved in the process update their respective local models and then contribute to the overall global model. The aforementioned iterative procedure persists until the global model reaches a satisfactory performance level.
- Customized models: The FLS enables the customization of models to cater to each participant's specific needs and requirements. Participants may have models customized to their individual needs, depending on the distribution of data and local requirements.

2.2.2. Components of a Federated Learning Framework

In today's data-driven world, the conventional centralized approach to ML—in which data from multiple sources is pooled on a single server for training—is encountering obstacles, particularly regarding privacy and efficiency. This technique collects data from various sources and then stores it on a server. FL has emerged as a potential solution, enabling decentralized training while ensuring that data is kept on its original device, thereby reducing the overhead associated with data transfer [24]. This section discusses the fundamental elements that comprise an FL framework.

- Client devices: These are edge devices, including smartphones, tablets, IoT devices, and even personal computers; they can store and process data locally and oversee local model training.
- Central server: This entity serves as the primary aggregation point in the FL structure. The central server is responsible for communicating with client devices, collecting model updates, and disseminating the global model back to the clients [21].
- Local models: Each client device is equipped with its own version of the ML model, which is trained using the local data available on that device.
- Global model: This model aggregates all the local models stored on the client devices and is hosted on the central server.
- Communication protocol: The primary objective of the communication protocol is to establish reliable and effective communication between the client devices and the central server while ensuring the security of the data sent. It is responsible for overseeing the transmission of updates to the model and the distribution of the global model.
- Aggregation algorithm: The algorithm is implemented on the central server, integrating the model updates received from all client devices to enhance the global model.
- Privacy mechanisms: During model aggregation and communication, additional layers of data security can be added by integrating various techniques, such as differential privacy and Secure Multiparty Computation (SMPC).

By gaining a comprehensive understanding of the fundamental components of FL, one can develop a deeper appreciation for the complexities and possibilities that FL offers

in addressing contemporary challenges within the field of data science. The significance of such decentralized techniques is expected to continue expanding as the digital ecosystem evolves, making FL a cornerstone in the future of ML.

2.2.3. Typical Federated Training Process

The FL process begins with each device developing a localized model using its own dataset. After completing local training, the device transmits model changes—specifically weights and gradients—to a central server, rather than sending raw data [16]. This approach ensures that confidential information remains in its original location, effectively mitigating various privacy risks commonly associated with traditional data centralization [25]. The model updates from all participating devices are consolidated on the central server to create an enhanced global model that incorporates insights derived from all the decentralized data sources. This aggregated model is then distributed to all devices, allowing them to leverage the collective intelligence of the entire network. The iterative process involves local training, model update transmission, aggregation, and global model dissemination, with each iteration progressively improving the accuracy and resilience of the global model [21]. By employing this innovative methodology, FL addresses the challenges related to data privacy and ML efficiency, effectively utilizing a wide range of authentic data sources from the real world while safeguarding the security of individual data [26].

2.2.4. Federated Learning Systems Taxonomy

FLSs facilitate cooperative model training while upholding the principles of data privacy and security. This approach is especially suitable for situations where data is distributed across multiple sources, and the parties involved are hesitant to share their data in a centralized manner. Many new FLSs have emerged since the creation of FL in 2016. There are a general taxonomy describing the difference of FLS is was presented in [19] and also replicated in [27]. Even though their taxonomy was very helpful for many researchers, it had several limitations that need to be addressed. Firstly, the taxonomy primarily focuses on the most prevalent and widely adopted Federated Learning scenarios, and as such, does not encompass all possible scenarios. Secondly, there might be gaps in terms of the different types of data distributions, models, and algorithms presented, indicating that the taxonomy might not be exhaustive. Thirdly, the taxonomy does not delve deeply into the specifics of each category, which could lead to overlooking certain nuances. Lastly, it is worth noting that the taxonomy is a reflection of the state of Federated Learning in 2021 and may require updates.

As the domain progresses, we present a general taxonomy describing the differences between these FLSs in this section. We use the taxonomy to clarify the distinctions between different FLSs, which can be categorized according to their essential features and characteristics. This multidimensional classification considers the most significant components of FLSs, such as data sources, privacy, model aggregation techniques, learning models, scalability, and network topology. Given the prevalent system abstractions and foundational components employed in various FLSs, we can classify these systems based on six key aspects: data distribution, model management, privacy method, communication architecture, FL algorithms, optimization techniques, use cases, and applications. Figure 2 shows this taxonomy of FLSs.

Data Distribution

When discussing the taxonomy of FLSs, the term "data distribution" refers to the process by which various participants or nodes in an FL environment are given different portions of the data. It affects the effectiveness and level of privacy maintained during the learning process, making it an essential component of FL.

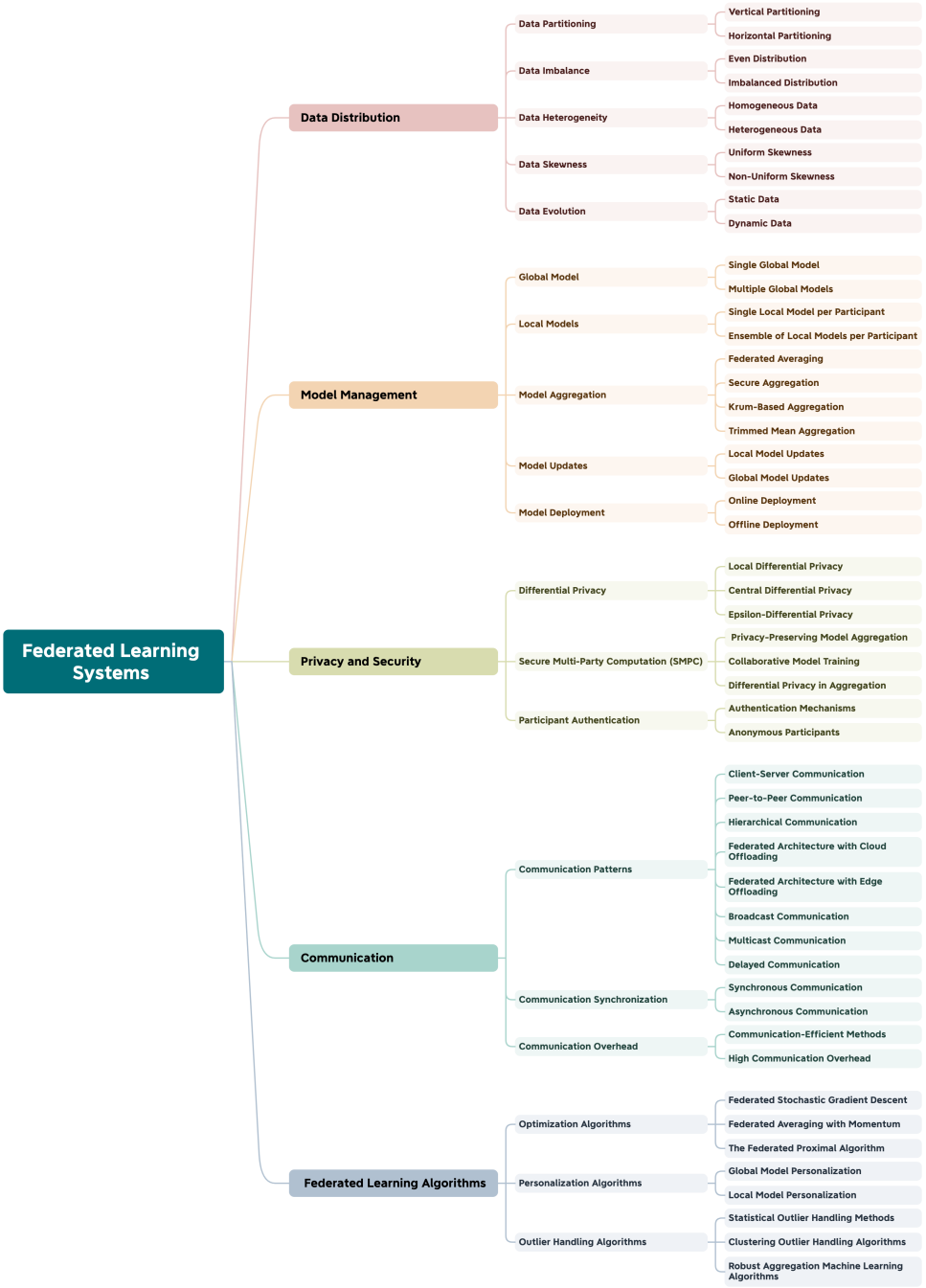

Figure 2. Taxonomy of Federated Learning systems (FLSs).

The following list reviews important factors regarding the distribution of data within this taxonomy:

- Data partitioning: The concept of data partitioning describes how data is allocated or divided among entities. Generally, FLSs can be divided into two categories, vertical and horizontal FLSs, depending on how the data are spread over the sample and feature spaces. The vertical partitioning strategy involves allocating distinct aspects or attributes of the dataset to various participants. For instance, a given participant may possess data pertaining to age and gender, whereas another participant may possess data on income and location. By contrast, in horizontal partitioning, participants have access to distinct sections of the data instances. A slice of the dataset with the same attributes belongs to each participant. For example, one member might have customer

data for a particular location, whereas another participant might have customer data for a different location.

- Data imbalance: The notion of data imbalance holds significant importance within the taxonomy of FLSs as it relates to the uneven allocation of data among the participants or nodes within the system. An imbalanced data distribution can have a substantial impact on the performance, fairness, and effectiveness of FL models [28]. Generally, FLSs can be categorized into systems with an even distribution and those with an imbalanced distribution. In even distribution, data can be distributed among participants to ensure an equitable allocation, thereby resulting in each participant possessing a proportionate share of the data. This methodology is commonly employed in situations with a reasonably equal distribution of data among participants and without substantial disparities in the quantity or significance of the data. In contrast, in imbalanced distribution, participants' data are not dispersed equally, resulting in some participants having noticeably more data than others. Managing data imbalance is a crucial factor to consider, as it might impact the FL process's performance and fairness.
- Data heterogeneity: A vital component of the FLS taxonomy is data heterogeneity, which describes the variation in the kinds, forms, and quality of data among nodes or participants in an FLS [29]. The FL context offers different opportunities and problems when dealing with heterogeneous data. Homogeneous data refers to particular instances of FL where the data possessed by participants exhibits a considerable degree of similarity concerning data type, format, and quality. The utilization of homogeneous data in the FL process facilitates the training of models by enabling a more streamlined approach since the consistency of the data allows for easier training. Homogeneous situations can facilitate model aggregation, sharing updates, and making assumptions about data features. On the other hand, heterogeneous data relates to scenarios when the data obtained from diverse participants exhibit notable variations in terms of data kinds, formats, and quality. Heterogeneity can manifest in myriad ways, such as disparities in feature representations, variations in data preparation techniques, and discrepancies in data-gathering methodologies. Data heterogeneity arises for a variety of reasons, such as the utilization of disparate technologies, the involvement of many companies, and the integration of data from sources that possess separate data schemas. The issue of data heterogeneity is highly significant in the FL context, as it has notable implications for the capacity to develop a valuable global model from varied data sources while ensuring data privacy and model performance. The efficient management of data heterogeneity and adaptation to accommodate the different attributes of individual participants' data are crucial considerations in developing effective FLSs.
- Data skewness: The concept of data skewness holds significant relevance within the FLS taxonomy, as it specifically refers to the uneven distribution of data across the participants or nodes in an FLS. Skewness pertains to the extent of asymmetry or lopsidedness in the distribution of data [30]. The comprehension of data skewness is essential due to its potential impact on the performance, fairness, and convergence of models in the FL context. In certain instances of FL, the distribution of data among participants may exhibit a uniform skew. This implies that each participant's data are subject to a comparable degree of skewness. Uniform skewness is observed when participants show identical patterns of data distribution despite potential variations in the quantity of data. By contrast, non-uniform skewness is observed when the skewness of the data distribution across different participants varies. Certain participants may have heavily skewed data distributions, while others may have more evenly balanced ones. Dealing with non-uniform skewness can pose difficulties as it necessitates accepting diverse levels of skewness in the data distribution. The presence of data skewness In FL gives rise to several challenges, including training imbalance, model bias, concerns over privacy, and increased communication overhead. Weighted

aggregation approaches can be utilized in FLSs to address the issue of data skewness and reduce its impact. These strategies involve allocating varying weights to participants based on the degree of skewness in their data distribution. Participants who possess more highly skewed data may be assigned lower weights in order to prevent their data from exerting an excessive impact on the overall model.

- Data evolution: The evolution of data in FLS taxonomy pertains to the temporal modifications that transpire within the datasets maintained by participants. These modifications can have noteworthy consequences in terms of the efficiency and precision of FL models. In certain FL situations, the data remain static throughout the FL process, resulting in a simplified training procedure. Static data are typically seen in situations where the underlying data exhibits few changes, as in the case of historical datasets or reference databases. Notably, dynamic data have the potential to change over time. Consequently, participants may find it necessary to update their respective local datasets regularly.

Model Management

Another critical component of the FLS taxonomy is model management, which refers to the approaches and techniques used to manage the machine learning models within an FL framework. It includes several aspects of model deployment, customization, aggregation, initialization, and updates in FL environments. The following elements are essential in understanding model management within the FLS taxonomy:

- Global model: In an FL environment, the global model represents the machine learning model trained and updated collectively by all participating devices or nodes. Without consolidating the data, the global model captures the common knowledge derived from the decentralized data sources. Most FL situations have a single global model that all participants work together to enhance. Meanwhile, some global models may be employed in other specialized applications, each tailored to a particular task, set of features, or user group. The central focus of FL is the global model, encapsulating collective intelligence from various data sources while safeguarding data privacy and promoting decentralization. The successful management of the global model is crucial, involving appropriate initialization, secure updates, and precise evaluation.
- Local model: The term "local model" refers to individual machine learning models maintained and updated by each participating device or node in the network. These local models are trained using local data accessible on each individual device, and the raw data is not shared with a centralized server throughout this process. Each participant may have their own unique local model, which they are responsible for maintaining. During the training, participants do not discuss their models, nor do they exchange raw data or model parameters with one another. As an alternative, each participant may maintain their own local ensemble of models, allowing for a variety of perspectives and levels of competence. The ensemble may include models that use various algorithmic approaches, architectural layouts, or hyperparameter settings. More reliable and accurate results can sometimes be achieved by combining the predictions of different models. Local models are essential to the FL process because they enable individuals to contribute to the collective intelligence without compromising the privacy of their personal information. Effective local model management is crucial for the success of FLSs across various domains and applications. This management must include secure training, customization, and evaluation.
- Model aggregation: A key component of model management in the FLS taxonomy is model aggregation, which describes the procedure for combining local model updates from many collaborators to produce a current global model. This procedure is essential to FL because it guarantees the integration of all participants' aggregate knowledge without centralizing their raw data [31]. FLSs use various standard methods for aggregating models. Federated averaging is the most widely used model aggregation technique for FL. After using their own data to train their local model, all participants

transmit the updated model—gradients—to a central server. A new global model is produced by averaging these modifications. Since no raw data is transferred, privacy is guaranteed [32]. This approach's efficiency and simplicity enable quick adjustments to the global model. Another popular FL model aggregation method is secure aggregation. This technique combines model updates while protecting the privacy of individual modifications. It uses cryptographic techniques, such as SMPC, to aggregate data without disclosing the unprocessed changes. It is appropriate for sensitive applications since it offers a high degree of privacy and secrecy. Additionally, it safeguards the integrity of the aggregation process from malevolent attempts. Krum-based aggregation, the third aggregation technique, is designed to stave off Byzantine attacks. This aggregation approach entails sorting the updates from participants according to their impact. The update with the smallest cumulative distance to the k-nearest updates is selected for aggregation [33]. Because it is robust against updates that differ significantly from one another, it can be used in adversarial environments. Trimmed mean aggregation, a popular variation of federated averaging, removes a predetermined proportion of extreme updates before averaging. After sorting the participant updates, the updates with the largest variances from the mean are eliminated. By using this method, the aggregation process becomes more resilient to updates that contain outliers. Participants in the weighted aggregation technique are given varying weights according to the caliber or applicability of their updates. During aggregation, higher weights are assigned to participants who provide more accurate or diverse updates, increasing their contributions' effect on the global model. This allows for the prioritization of more trustworthy or pertinent updates, enhancing the global model's overall quality. In FL, model aggregation is a crucial stage since it establishes the quality and efficacy of the final global model. The best aggregation technique is determined by specific application needs, such as privacy concerns, resilience against adversarial attacks, communication limitations, and required model quality. Effective model aggregation approaches enable FLSs to create precise, reliable, and privacy-preserving global models.

- Model updates: Model updates pertain to modifications made to machine learning models during the FL process. Implementing these updates is paramount in improving the models' overall performance, accuracy, and generalization capabilities. On the one hand, local model updates can be employed in scenarios where players train their own local models using their respective datasets, resulting in model updates derived from their individual training procedures. Local updates are computed via methodologies such as stochastic gradient descent (SGD) or its variations, such as federated averaging. The updates are contingent upon the data on individual participants' devices, enabling models to catch localized patterns. On the other hand, in global model updates, changes are computed by aggregating information from the local models of all participants. Global updates are produced by combining the local model updates contributed by several participants. These updates indicate the cumulative understanding of the FL network as a whole. Model updates play a vital role in the FL context, as they enable the integration of the collective intelligence derived from various data sources into one cohesive and improved model. The success of FLSs in many domains and applications heavily relies on the efficient administration of model updates, encompassing privacy protection, security, and adaptability.
- Model deployment: Model deployment in FLSs includes the steps required to make the trained machine learning model accessible and functional for generating predictions or providing services to end-users or applications. However, model deployment in FL exhibits notable differences from conventional machine learning model deployment, mostly stemming from the decentralized and privacy-preserving characteristics inherent in the FL methodology. The strategic process of deploying models in FLSs involves striking a compromise between real-time adaptation and safeguarding user privacy and data security. FL involves the collaborative training of models on dispersed

devices while preserving the confidentiality of sensitive data within local servers [26]. Following the completion of training, models can be deployed in both online and offline environments. The process of online deployment in FLSs entails the seamless and immediate incorporation of model changes originating from distributed devices. This facilitates the prompt reaction to evolving data patterns and user behaviors in real time. This methodology enables rapid model aggregation, maintaining the pertinence and precision of forecasts in dynamic settings. By employing strategies such as the integration of real-time noise injection to ensure privacy and the implementation of continuous monitoring, the online deployment of the model ensures its ability to promptly adapt to developing trends. Feedback loops facilitate the collection of user interactions in real time, enabling prompt modifications and refinements. Utilizing adaptive learning rates and personalization settings guarantees customized experiences for individual users. Online deployment generally ensures that FL models offer timely, accurate, customized predictions while protecting user privacy. This makes it crucial for applications that require swift and exact answers to real-time data streams. Conversely, offline deployment in FLSs encompasses using pretrained models on novel data without necessitating real-time adaptation. After the FL model completes the training and aggregation process by incorporating updates from devices involved in the process, it can be implemented offline for many applications. Offline deployment is especially advantageous in situations when immediate adjustment is not critical and regular updates are satisfactory. In this particular situation, the model that has undergone training is implemented on servers or edge devices, enabling it to provide predictions or services by leveraging its accumulated knowledge. This deployment strategy demonstrates efficacy when employed in applications characterized by consistent data patterns and when privacy-preserving methodologies have been included during the training phase. Although offline deployment may not possess the immediate responsiveness of online deployment, it offers the advantage of ensuring consistency and accuracy in predictions. This characteristic renders offline deployment well-suited for numerous FL applications. Table 2 provides a comparison of online and offline model deployment in FLSs.

Table 2. Comparison between online and offline model deployment in Federated Learning systems.

Adaptability	Online deployment is well-suited for dynamic environments subject to rapid change, as it enables instant adaptation to new data. In contrast, offline deployment ensures consistency but may not adapt as rapidly to new circumstances.
Privacy	Both deployment strategies prioritize privacy during the training period. However, online deployment guarantees real-time privacy maintenance while updating the model, offering enhanced privacy for continuous interactions.
Resource usage	Online deployment requires consistent and instantaneous information exchange, as well as the availability of computational resources to implement model revisions promptly. In contrast, offline deployment reduces the need for continuous communication, enhancing resource efficiency.
Use cases	Online deployment is highly advantageous in scenarios where real-time adjustments and customized responses are crucial. Offline deployment is a suitable option for applications requiring periodic model updates and consistent forecasts, particularly in situations where continuous communication may not be practical or essential.

The selection of online or offline deployment in FLSs is contingent upon the particular use case, data patterns, privacy stipulations, and the necessity for real-time adaption. Each option presents distinct advantages, enabling organizations and developers to customize their strategy according to the application's specific requirements.

Privacy and Security

Privacy and security are of utmost importance in the taxonomy of FLSs. The preservation of data privacy, secrecy, and security is critical due to the involvement of various sources. The following key elements pertain to privacy and security within FLSs:

- Differential privacy: Differential privacy is a core principle within the field of privacy-preserving data analysis, such as in FL. Differential privacy techniques ensure that the presence or absence of a particular data point does not materially affect the output by adding noise to the computations made on the data [34]. Even when combined with or applied to updates to machine learning models, it safeguards the privacy of individual data pieces. Differential privacy is used in FLSs to protect participant data privacy while enabling group participation in machine learning model training [35]. The following are the fundamental types of differential privacy inside FLSs:
  - Local differential privacy (LDP): When using LDP, noise is applied locally to individual data points on the users' devices before transferring the perturbed data to the central server. This ensures that raw data are never transmitted outside users' devices, offering a higher level of privacy but making it more difficult to aggregate the data [35].
  - Central differential privacy (CDP): In the CDP technique, noise is added to the aggregated statistics or model parameters in a centralized location. This helps to ensure that no participant's data are made public. It is appropriate for situations in which a reliable central server compiles the updates contributed by participants without disclosing their private data [35].
  - Epsilon-differential privacy (ϵ-differential privacy): The level of privacy can be quantified using a parameter known as epsilon. A lower value for epsilon indicates a greater degree of discretion and confidentiality. A balance must be struck between personal privacy and practicality. Lower values result in increased privacy but could also lead to a less accurate global model.

In FLSs, differential privacy is crucial to guaranteeing that users can provide data for model training without risking their privacy. Ensuring the security of sensitive information while maintaining accuracy in models is a critical component of privacy-preserving machine learning in collaborative settings. Table 3 provides a comparison summary between these three standard differential privacy techniques.

- Secure multiparty computation: SMPC is a cryptographic methodology that facilitates collaborative computation of a function by numerous entities while ensuring the privacy of their inputs. Within the realm of FLSs, SMPC assumes a pivotal role in upholding privacy and security. The integration of SMPC inside the FLS taxonomy can be elucidated as follows:
  - Privacy-preserving model aggregation: SMPC guarantees participants' ability to safely submit their model updates or gradients to collectively construct a global model while ensuring that no individual party can access the specific contributions made by others. The integration of collective intelligence from multiple participants while maintaining individual privacy is of utmost importance in the FL context.
  - Collaborative model training: The SMPC technique facilitates the cooperative training of machine learning models, allowing participants to cooperatively compute model parameters without sharing their raw data. Collaborative efforts among participants can be employed to enhance the model's accuracy while maintaining the privacy and confidentiality of their respective datasets.
  - Differential privacy in aggregation: The combination of SMPC and differential privacy approaches allows for the introduction of noise into aggregated results, hence offering a robust privacy assurance. The utilization of the aggregated model guarantees that the determination of the contribution made by any particular

participant remains computationally infeasible, thus upholding the preservation of individual privacy [35].

Table 3. Comparison summary between three common differential privacy techniques.

	Local Differential Privacy	Central Differential Privacy	Epsilon-Differential Privacy
Privacy level	Ensures robust individual privacy by ensuring raw data remains exclusively on the participants' devices.	Provides privacy at an aggregate level, ensuring that individual data is not directly exposed in any circumstance.	Adjusting epsilon allows for custom privacy levels, offering flexibility.
Aggregation complexity	Aggregating locally perturbed data while maintaining privacy is complex.	Since noise addition happens centrally, aggregation is simpler.	The aggregation complexity is determined by the particular implementation and the noise-adding mechanism.
Centralization	Completely decentralized, with no centralized entity participating in the processing of data.	The addition of noise necessitates the use of a reliable centralized server, resulting in centralization.	The centralization level depends on the noise-adding mechanism.
Flexibility	High privacy but increased noise levels may reduce usability.	A balanced approach with group-level privacy.	Flexible, enabling privacy levels to be changed in accordance with application needs.
Challenges	Aggregation complexity. Increased noise.	Central trust. Potential central attack.	Utility trade-off.

The utilization of SMPC-based aggregation plays a crucial role in the FL context, as it enables participants to collectively improve the accuracy of a global model while simultaneously ensuring the protection of their data's privacy. FLSs can utilize safe multiparty computation techniques to use the combined information from various remote sources effectively. This approach ensures that security, privacy, and integrity are maintained during the collaborative learning process.

- Participant authentication: Participant authentication is an essential component within the taxonomy of FLSs, as it verifies the identity and legitimacy of entities involved in the collaborative learning process. Participant authentication can be conducted using authentication mechanisms, or participants can remain anonymous through anonymous participant techniques. Authentication systems are an essential feature of FLSs. They ensure that all participants and entities interacting with the system have their identities checked and are granted the appropriate permissions. Within the FLS taxonomy, the following authentication mechanisms are used:
  - User credentials authentication: In this mechanism, participants must provide their usernames and passwords to verify their identities. This fundamental mechanism is used extensively despite the fact that, if not adequately secured, it is susceptible to password-based attacks.
  - Biometric authentication: The authenticity of the participants is determined by using distinctive biological characteristics, such as fingerprints or facial recognition. Because copying biometric data is so complex, this technique provides a very high level of security.
  - Token-based authentication: Participants authenticate subsequent requests with tokens, which are typically generated once an initial login has been completed successfully. It improves security by minimizing the amount of sensitive credentials that must be transmitted regularly.
  - Certificate-based authentication: Participants show digital certificates signed by a reliable certificate authority to authenticate themselves. This improves security by guaranteeing that a reliable third party confirms participants' identities.

- Multi-factor authentication: In order to gain access, participants are required to furnish a variety of authentication methods, including a password and a verification number transmitted to their mobile device. The implementation of various proofs of identification enhances the level of security.
- oAuth and OpenID connect: The use of secure authentication and authorization protocols is prevalent in web-based FLSs. The system offers standardized and secure authentication techniques, effectively integrating them with a wide range of applications and services.
- Device-based authentication: The authentication of participants' devices is contingent upon the utilization of distinct device identifiers or certificates linked to the hardware. Implementing device authorization in the FLS bolsters security measures by only allowing access to authorized devices.
- Role-based access control: Participants are allocated distinct roles and permissions in accordance with their respective tasks inside the FLS. Implementing access controls guarantees that participants possess suitable levels of access, hence mitigating the risk of unauthorized activities and access to data.
- Continuous authentication: The activities and behaviors of participants are continuously watched to identify any anomalies, ensuring that authenticated users maintain their authentication status. Including this feature enhances security measures by rapidly detecting and addressing any questionable behavior.
- Symmetric encryption: In FLS, symmetric encryption plays a crucial role in maintaining data confidentiality and integrity. This method, utilizing the same key for both encryption and decryption, is particularly efficient for the large volumes of data typical in FLS. It ensures that sensitive information remains secure during transmission, as only model updates or insights are shared across the network, not the raw training data. This encryption method not only protects the data from potential eavesdroppers but also maintains their integrity, making any unauthorized alterations easily detectable. While symmetric encryption is central to preserving data privacy and consistency in FLS, it is typically complemented by other security measures, such as secure key management protocols, to provide a comprehensive security framework. The efficiency and effectiveness of symmetric encryption in these systems highlight its indispensability in the secure and efficient operation of FLS.

The specific requirements of a particular FLS determine the most suitable authentication approach, taking into account aspects such as security, usability, scalability, and management complexity. These techniques can frequently be combined to achieve an efficient balance between security and usability. Table 4 provides a comparison of the main authentication mechanisms within FLSs.

- Anonymous participants: When discussing FLSs, the term anonymous participants refers to the practice of protecting the participants' right to privacy and maintaining the confidentiality of their data and identities. Ensuring that users can participate in FLSs while maintaining anonymity is essential for protecting their data privacy. This objective is accomplished by using a variety of strategies and approaches. In FLSs, the following methods are frequently used by participants who wish to remain anonymous:
  - Participant identity concealment: The concealment of participants' identities is a crucial measure in the FL process, as it guarantees the protection of their personal information from being disclosed. The preservation of user privacy fosters engagement from individuals and businesses who are apprehensive about the potential risks associated with data disclosure.
  - Data anonymization: In the context of FL, personal data undergo anonymization procedures prior to engagement, guaranteeing that even in the event of unauthorized access, the data cannot be directly associated with identifiable individuals.

Methods such as differential privacy, k-anonymity, and data perturbation can be employed to achieve data anonymization
- Pseudonymization: During the FL process, participants are not required to reveal their true identities and instead employ pseudonyms or temporary IDs. The utilization of this technology affords a level of anonymity, making it more challenging to trace particular data contributions to specific individuals.
- Blockchain-based identity management: The utilization of blockchain technology facilitates the management of participants' identities and transactions in a decentralized and tamper-proof manner. The elimination of a central authority and the provision of transparent and safe identity management contribute to the enhancement of security and privacy.

The emphasis on preserving participant anonymity is pivotal for building trust, encouraging engagement, and safeguarding privacy within FLSs. FLSs align with regulatory frameworks like the General Data Protection Regulation, which prioritize principles such as user permission and the anonymization of personal data. By taking these factors into account, FL platforms have the potential to establish a secure, privacy-preserving environment for collaborative machine learning initiatives.

Table 4. Comparison of the main authentication mechanisms within Federated Learning systems.

Authentication Mechanisms	Strengths	Weaknesses
User credentials authentication	Simple, widely understood and used.	Vulnerable to password-based attacks if weak passwords are used.
Biometric authentication	Highly secure, unique to individuals, eliminates the need for passwords.	Hardware requirements (e.g., fingerprint scanners), potential false positives/negatives.
Token-based authentication	Reduces reliance on passwords, enhances security for multiple requests.	Requires secure token storage and transmission mechanisms.
Certificate-based authentication	Strong security, verified by certificate authorities.	Complex certificate management, reliance on a trusted certificate authority.
Multi-factor authentication	Adds an extra layer of security, even if one factor is compromised.	User inconvenience, requires additional verification steps.
OAuth and OpenID Connect	Widely adopted, standardized, secure token-based authentication.	Requires integration and understanding of protocols.
Device-based suthentication	Ensures device authenticity, useful for Internet of Things devices.	Complex device management, potential security vulnerabilities.
Role-based access control	Granular control over user permissions, scalable for large systems.	Initial setup complexity, requires ongoing management.
Continuous authentication	Provides real-time security monitoring, identifies and responds to anomalies.	Requires sophisticated monitoring tools, potential false positive/negative issues.

Communication

Another vital element of FLSs is communication, which involves the exchange of data between the central server and participants, including devices, clients, or edge nodes. Effective and safe communication is necessary for FLSs to function well. The following provides an analysis of the various components related to communication within the taxonomy of FLSs.

- Communication patterns: Communication patterns in FLSs concern how participants, including diverse devices or entities, interact with one another and the central server during the collaborative learning procedure. These patterns play a crucial role in facilitating the effective and secure transmission of data and updates to models. In FLSs, the following communication patterns are considered to be the most common:

- Client–server communication: In this communication pattern, the participants' devices establish direct contact with a central server, through which they transmit their changes and receive aggregated model parameters. It is frequently observed in situations where participants possess restricted computational capabilities and depend on a central server to aggregate models.
- Peer-to-peer communication: In this setting, users directly communicate with one another, facilitating the exchange of model updates or aggregated information without needing a central server to mediate the process. Utilizing decentralized environments is advantageous as it allows players to establish direct connections, minimizing latency and decreasing reliance on a central server.
- Hierarchical communication: In this pattern, participants are systematically grouped into hierarchical structures, wherein updates are initially consolidated at lower levels before being transmitted to higher levels for additional consolidation. This approach exhibits scalability, particularly in the context of massive federated networks, enabling effective aggregation at several hierarchical levels [36].
- Federated architecture with cloud offloading: Participants carry out the preliminary computations at their respective locations and then send the more intensive computations (such as aggregation) to a central server hosted in the cloud. It allows devices with limited resources to take part by offloading complicated activities and distributing computation evenly between on-premise and remote resources.
- Federated architecture with edge offloading: The process resembles cloud offloading, but its computations are offloaded to edge devices situated within the local network. This approach diminishes latency and decreases dependence on a remote cloud server. This technology is well-suited to use cases requiring real-time responses and minimal delay, frequently seen in IoT and edge computing environments.
- Broadcast communication: The central server disseminates model updates to all participants concurrently, maintaining consistency across all devices. The broadcasting of updates, particularly when all participants require identical model parameters, conserves bandwidth and reduces time consumption.
- Multicast communication: Model updates are distributed to distinct groups of participants, enabling selective broadcasting based on the degree to which two sets of data are comparable. When multiple groups of people work on similar activities, this pattern is helpful because it allows for the more efficient use of network resources.
- Delayed communication: Participants gather updates on their local machines and deliver them in batches at regular intervals, thus decreasing the time spent communicating with the centralized server. This reduces the overhead of transmission and the delay, particularly in situations when real-time updates are not essential.

The selection of a communication pattern substantially influences the effectiveness, scalability, and responsiveness of FLSs, rendering it a critical element in their design and execution. Every communication pattern possesses distinct advantages and trade-offs, making them appropriate for specific use scenarios. The selection of a particular pattern is contingent upon various aspects, including but not limited to the configuration of the network, the capabilities of the participants involved, the need for real-time features, and the need to maintain anonymity. A combination of these patterns is frequently utilized to achieve an ideal balance of various components.

- Communication synchronization: In FLSs, "communication synchronization" refers to the process of coordinating and aligning the various communication activities that take place among the participating devices or nodes. It ensures that the processes of aggregation, exchanging data, and updating models happen in a structured and synchronized way. The devices must be synchronized correctly in order to maintain the reliability and precision of the collaborative model being trained across distributed devices. Communication in FLSs can be either synchronous or asynchronous [37].
- Synchronous communication: This type of communication involves individuals sending real-time updates on a predetermined timetable to the central server or other

participants. Everybody synchronizes their communication so that aggregations and model updates happen simultaneously. This synchronous method creates an FLS with a coordinated and organized workflow. Synchronous communication is necessary for applications in autonomous vehicles because it ensures that the vehicle's model can adjust in real time to the constantly shifting conditions of the road and its surroundings. To take full advantage of the benefits of synchronous communication in FLSs, it is vital to properly manage network latency and bandwidth usage.

- Asynchronous communication: This type of communication involves devices or nodes functioning autonomously without the requirement of precise time synchronization. In contrast to synchronous communication, which involves coordinating updates in rounds or at predetermined intervals, asynchronous communication enables participants to individually transmit their updates to the central server or other nodes according to their unique schedules [38]. Asynchronous communication, for instance, makes it easier for research institutes located in several time zones to collaborate, enabling scientists to share their discoveries without being constrained by synchronized communication periods. To fully utilize asynchronous communication's advantages in FLSs, its associated problems must be addressed.

The selection of one of these two approaches is contingent upon the particular demands and limitations of the FLS and the attributes of the involved devices or nodes. Table 5 provides a summary of the comparison between asynchronous and synchronous communication in the context of FLSs.

- Communication overhead: Within the context of FL, the term "communication overhead" refers to the additional data transmission and processing resources necessary for participants to exchange model updates, gradients, and other information while the collaborative learning process is being carried out [39]. The effective management of communication overhead is essential because it directly affects the bandwidth of the network, the latency, and the overall effectiveness of the FLS. In FLSs, a number of different techniques have been established to reduce the amount of communication overhead. Table 6 presents a comprehensive summary of several prominent methodologies.

Table 5. Comparison between asynchronous and synchronous communication in FLSs.

	Synchronous Communication	Asynchronous Communication
Communication Timing	Participants communicate according to a predefined schedule or specific synchronization points.	Participants communicate independently, sending updates whenever they have new data or model improvements to contribute.
Flexibility	Less flexible as participants are bound to fixed communication schedules, potentially causing delays for some participants.	Highly flexible, allowing participants to operate at their own pace, accommodating varying network conditions and device availability.
Dependency on central control	Often requires central control to coordinate communication, ensuring all participants adhere to the predefined schedule.	Reduces dependency on central control, enabling decentralized decision-making and autonomous operation of participants.
Latency	Lower latency as updates are synchronized, allowing rapid model adjustments and real-time responses to changing data patterns.	Potentially higher latency due to the lack of synchronization, especially if updates from critical participants are delayed.

Table 5. *Cont.*

	Synchronous Communication	Asynchronous Communication
Communication overhead	More predictable communication patterns, potentially reducing overall communication overhead.	Can lead to higher communication overhead due to the lack of synchronization, efficient data compression and differential updates are essential to manage this.
Adaptability to dynamic environments	Might struggle to adapt to dynamic environments where network conditions or participant availability fluctuate.	More adaptable to dynamic environments, allowing participants to contribute whenever they can, ensuring continuous collaboration.
Fault tolerance	Susceptible to disruptions if a participant fails to communicate at a scheduled time, potentially affecting the entire synchronization process.	More fault-tolerant as one participant's failure does not disrupt the entire system. Other participants can continue to contribute independently.
Privacy and security	Easier to implement security protocols and encryption as communication occurs at predictable times.	Requires robust encryption and security measures to ensure the safety of data transmitted independently by participants.

Each of the strategies mentioned above are designed to target distinct facets of communication overhead in the context of FL. Frequently, these methodologies are synthesized in practical contexts to attain maximum communication efficacy while concurrently upholding the principles of data confidentiality, model precision, and system promptness. The selection of methodologies is contingent upon the particular application scenario, prevailing network circumstances, and attributes of the involved devices. On the other hand, the presence of significant communication overhead in FLSs can be attributed to several variables. These variables include the utilization of large model sizes, frequent updates, non-selective participant communication, high data dimensionality, non-localized computing, and excessive reliance on encryption or privacy measures. The transmission of machine learning models across distant devices can result in massive data transmission and consume significant network resources, particularly when these models are sizable or updated often. The practice of non-selective communication further exacerbates the issue because all participants send updates without considering their relevance. In addition, many gradients must be transmitted for high-dimensional data, resulting in an additional increase in communication volume. When computations are concentrated in a central location, participants must send unprocessed data, resulting in inefficiencies. Furthermore, an excessive focus on encryption and privacy protocols can increase the quantity of data, thus intensifying the difficulties associated with communication. The presence of inefficient communication protocols can exacerbate these concerns. To address the issue of high communication overhead, techniques such as model compression, intelligent participant selection, dimensionality reduction, localized computation, and the judicious application of encryption methods must be strategically implemented. This ensures a balance between the protection of data and the effectiveness of communication.

Federated Learning Algorithms

FL algorithms are yet another essential component of FLSs. These algorithms make it possible to train collaborative models without transferring raw data between devices and a central server. This helps protect users' privacy while reducing the amount of communication overhead required. The following is a list of essential uses of FL algorithms.

Table 6. An overview of some of key communication-efficient methods.

Methods	Description	Advantages	Considerations
Differential updates	Instead of transmitting the entire model, participants compute and transmit only the changes (gradients) in their local model parameters.	Significantly reduces the amount of data transmitted, especially when only small parts of the model have changed.	Efficient algorithms are needed to calculate and transmit the differentials accurately.
Model compression	Techniques like quantization, where model parameters are represented with fewer bits, and pruning, where insignificant model weights are removed, reduce the model size before transmission.	Reduces the amount of data that must be transmitted, reducing bandwidth and computational overhead.	Balancing compression levels to maintain model accuracy is crucial.
Decentralized optimization	Algorithms like federated averaging allow model updates to be computed locally and averaged among participants, reducing the need to transmit raw data or gradients.	Minimizes communication overhead by performing local computations and transmitting only the aggregated model updates.	Requires careful coordination to ensure accurate aggregation.
Smart sampling and client selection	Algorithms that intelligently select a subset of clients for participation, reducing the total number of updates transmitted.	Reduces the communication overhead by selecting a representative subset of clients, optimizing the use of bandwidth.	Requires algorithms that balance randomness and representation to avoid biased sampling.
Edge computing	Computation and updates are performed locally on edge devices, reducing the need for frequent communication with a central server.	Minimizes communication by allowing edge devices to handle computations and updates, reducing latency and bandwidth usage.	Ensuring that edge devices have sufficient computational resources and storage capacity is essential.
Adaptive communication	Dynamic communication frequency and volume adjustment based on network conditions, participant capabilities, and system requirements.	Optimizes communication overhead in real time, ensuring efficient use of resources.	Requires continuous monitoring and adaptation, potentially introducing computational overhead.
Cryptography and encryption	Secure communication protocols use encryption techniques to protect data during transmission.	Ensures data privacy and security, allowing sensitive information to be transmitted securely.	Introduces computational overhead for encryption and decryption processes.

- Optimization algorithms: Optimization algorithms have a crucial function in FLSs, providing the aggregation of information from various devices and enabling the construction of accurate and efficient machine learning models. These algorithms have been specifically developed to achieve a harmonious equilibrium between the collaborative aspects of FL and the imperative requirements of privacy preservation and computational efficiency [21]. Federated optimization techniques commonly prioritize minimizing a global objective function by integrating local updates obtained from individual devices. Several examples of prominent optimization algorithms employed in FLSs are mentioned below:
  - Federated SGD: This is a pivotal algorithm that has revolutionized the framework of collaborative machine learning in decentralized environments, particularly in the context of Federated Learning Systems (FLSs). This algorithm offers a

nuanced approach to model training, diverging from traditional methods that necessitate the transmission of raw data to a central repository.
At the heart of Federated SGD lies the principle of gradient computation at the local device level. Each participating device in the network utilizes its local data to calculate gradients, which represent the partial derivatives of the loss function with respect to the model parameters. This local computation not only preserves the privacy of user data by avoiding raw data transmission but also significantly reduces the volume of data that needs to be communicated across the network. This aspect of Federated SGD is particularly advantageous in scenarios where network bandwidth is limited.

* Privacy preservation and data integration. The privacy-preserving nature of Federated SGD is one of its standout features. By enabling local gradient computation, the algorithm ensures that sensitive data remains within the confines of the originating device. These locally computed gradients, encapsulating the necessary information for model updates, are then securely transmitted to a central server [40]. On the server, an aggregation process takes place, where these gradients from multiple devices are combined to update the global model. This approach not only safeguards individual data confidentiality but also facilitates the integration of heterogeneous datasets into a unified model. By aggregating diverse local updates, Federated SGD harnesses the collective intelligence embedded in disparate data sources, enhancing the robustness and relevance of the global model.
* Bandwidth optimization and application versatility The reduction in data transmission volume inherent to Federated SGD addresses the challenges posed by restricted bandwidth environments. In traditional centralized learning models, the transmission of large volumes of raw data can be a significant bottleneck, consuming substantial network resources. Federated SGD elegantly circumvents this issue by transmitting only essential gradient information, thereby optimizing bandwidth usage. This optimization is crucial for ensuring the scalability and efficiency of FLSs, particularly when deployed in bandwidth-constrained settings. Furthermore, the versatility of Federated SGD extends its applicability across a broad spectrum of domains. From healthcare to finance, and from mobile computing to Internet of Things (IoT) applications, this methodology proves instrumental in diverse fields by facilitating effective model training across various scenarios while maintaining data privacy and minimizing risk.

– Federated Averaging with Momentum (FedAvgM): represents a significant enhancement over the traditional Federated Averaging (FedAvg) algorithm, primarily used in Federated Learning Systems (FLSs). This advanced algorithm introduces a momentum component to the model updates, enhancing the overall efficiency and accuracy of the learning process. FedAvgM not only leverages the collaborative capabilities inherent in Federated Learning but also introduces the stability and efficiency offered by momentum-based optimization. This results in a more robust and responsive learning algorithm capable of adapting to the nuanced requirements of distributed learning scenarios.
The central innovation in FedAvgM lies in the incorporation of a velocity component, or momentum, into the model updates. This momentum term allows the algorithm to 'remember' and integrate a portion of the previous update into the current one.

  * Enhanced convergence and optimization: By maintaining its previous trajectory through the velocity term, FedAvgM accelerates the convergence process. This momentum-driven approach is particularly beneficial in scenarios with non-IID data distributions or significant data volatility, where traditional FedAvg might struggle with slow or unstable convergence.

* Application in diverse scenarios: FedAvgM demonstrates remarkable effectiveness in a variety of distributed environments. Its ability to facilitate rapid and steady knowledge acquisition across distributed devices makes it an ideal choice for FLSs dealing with complex data landscapes. The algorithm effectively balances the need for accurate and efficient model training while maintaining user privacy and data security. In summary, Federated Averaging with Momentum elevates the traditional Federated Learning approach by introducing a dynamic and adaptive component that significantly enhances model training effectiveness. Its ability to handle complex data distributions and volatile environments, while ensuring rapid convergence and optimization, marks it as a valuable tool in the realm of Federated Learning. The inclusion of momentum in the federated averaging with momentum optimization algorithm enhances the traditional federated averaging approach in FLSs. This modification introduces a velocity component into model updates, enabling the algorithm to maintain its previous trajectory while accelerating, resulting in faster convergence and improved optimization. It facilitates rapid and steady knowledge acquisition across distributed devices, particularly in scenarios involving non-identically distributed (non-IID) data or significant volatility. Federated averaging with momentum demonstrates remarkable effectiveness in achieving accurate and efficient model training while safeguarding user privacy and data security. It combines the collaborative capabilities of FL with the stability offered by momentum-based optimization.

– The Federated Proximal Algorithm: The Federated Proximal Algorithm represents an advanced iteration in the evolution of FL algorithms, tailored to address the challenges posed by non-IID (independently and identically distributed) data across a network of devices. This algorithm is particularly relevant in scenarios where the data distribution varies significantly among the participating nodes, a common occurrence in real-world applications. The Federated Proximal Algorithm is built upon the foundation of the standard Federated Learning framework but introduces a crucial modification in the optimization process. The key innovation lies in the incorporation of a proximal term to the optimization objective. This term essentially acts as a regularizer that encourages the local models to not deviate significantly from the global model. The mathematical formulation of this algorithm involves adding a proximal term to the local loss function, typically represented as a squared Euclidean distance between the local and global model parameters.

  * Addressing non-IID data challenges: In standard Federated Learning setups, the assumption is often that the data across devices is identically distributed. However, in many practical situations, this assumption does not hold, leading to significant challenges in model convergence and performance. The Federated Proximal Algorithm mitigates these issues by ensuring that local model updates remain 'proximal' to the global model. This approach effectively handles the statistical heterogeneity of data, ensuring more stable and consistent model training across diverse data distributions.
  * Optimization process in Federated Proximal Algorithm: During the training process, each participating device computes its local model update by optimizing the modified loss function, which includes the proximal term. Once the local updates are computed, they are sent to a central server where a global aggregation occurs. The server updates the global model by averaging these updates, similar to standard Federated Learning, but with the added nuance provided by the proximal regularization.
  * Advantages and practical applications: The incorporation of the proximal term offers several advantages. Primarily, it enhances model performance

in non-IID data scenarios, which are prevalent in many real-world applications such as healthcare, finance, and mobile services [41]. Additionally, by controlling the extent of deviation of local models from the global model, the Federated Proximal Algorithm promotes more uniform learning across the network, leading to improved overall model accuracy and convergence rates. In summary, the Federated Proximal Algorithm represents a significant advancement in the field of Federated Learning, offering a robust solution to the challenges posed by non-IID data distributions. Its ability to ensure consistent and efficient learning across a decentralized network of devices makes it a valuable tool in the arsenal of Federated Learning algorithms.

The optimization algorithms utilized in FLSs undergo continuous development to effectively address the challenges posed by diverse and privacy-sensitive data. The use of these algorithms ensures the efficient generation of precise global models in FL while protecting user privacy. As a result, these algorithms play a critical role in advancing collaborative and privacy-preserving machine learning methodologies.

- Personalization algorithms: Personalization algorithms within FLSs play a crucial role in customizing user experiences while preserving data privacy. These algorithms facilitate the development of personalized models for users while ensuring the decentralization of their sensitive data. Personalization algorithms utilize data from local interactions and activities on user devices to discern trends and preferences. FL enables the integration of these insights into the global model while upholding user privacy. This practice ensures that recommendations, services, or materials provided to consumers are highly relevant and engaging, aligning with their preferences and needs [42]. FL empowers organizations and service providers to deliver personalized experiences on a large scale, simultaneously enhancing user satisfaction and safeguarding their privacy and data security. Personalization algorithms can be applied to tailor both global and local models within FLSs.
  - Global model personalization in Federated Learning Systems: Global model personalization within Federated Learning Systems (FLSs) is a sophisticated approach that aims to adapt a universally trained model to meet the specific needs and preferences of individual users or user groups. This concept is particularly vital in ensuring that the one-size-fits-all model can be effectively tailored to diverse user contexts while preserving privacy and data security. Global model personalization involves the adaptation of a shared global model, initially trained across multiple devices or data sources, to better align with the unique characteristics, behaviors, or preferences of individual users or specific segments [43]. This adaptation is crucial in FLSs, where a single global model is collaboratively trained but needs to be relevant and effective for each participant in the system. Techniques for global model personalization:
    - Client-side personalization: This involves adjusting the global model on the client's device using local data. Techniques such as model fine-tuning, where the model is slightly adjusted using the user's data, or layer retraining, where specific layers of the model are retrained, are commonly used.
    - User embeddings: Incorporating user embeddings into the model is another effective method. User embeddings are vector representations that capture the unique characteristics of each user. These embeddings can be integrated into the global model to ensure that the model's outputs are personalized for each user.
    - Transfer learning: Leveraging transfer learning, where a model trained on one task is adapted for another related task, can also be employed for personalization. This is particularly useful when the global model is trained on a broad dataset but needs to be adapted for specific user scenarios.

* Meta-learning: Meta-learning, or learning to learn, is a technique where the model is trained to quickly adapt to new tasks or data. In the context of personalization, meta-learning can enable the global model to rapidly adjust to individual user data.

Challenges in global model personalization:

* Data diversity and quality: Ensuring that the global model can effectively personalize across a wide range of diverse user data is a significant challenge.
* Resource limitations: The computational and storage limitations of client devices must be considered, especially when personalization involves additional model training on the device.
* Privacy concerns: Maintaining user privacy during the personalization process, especially when user-specific data are used for model adjustments, is crucial.

Global model personalization in FLSs represents a key strategy in making Federated Learning models more user-centric and effective. By adapting the shared global model to align with individual users' unique tastes and features, FLSs can provide customized and relevant experiences to users, enhancing the overall utility and acceptance of these systems.

– Local model personalization in Federated Learning Systems: Local model personalization in Federated Learning Systems (FLSs) addresses the challenge of customizing machine learning models at an individual level, using data that reside on a user's device. This approach is crucial in FLSs, where maintaining data privacy and catering to specific user needs are paramount.

Local model personalization revolves around adapting a federated model to fit individual user profiles based on their unique data. Unlike global model personalization, which modifies a shared model to suit general user characteristics, local personalization focuses on leveraging data available on each user's device to create a model that reflects their specific preferences, behaviors, and usage patterns.

Techniques for local model personalization:

* On-device training: This involves adjusting the federated model directly on the user's device. The model is fine-tuned with the user's local data, ensuring that the personalized model captures individual preferences and behaviors.
* Data augmentation: Enhancing the local training process with data augmentation techniques can improve the model's ability to learn from a limited amount of user data. This might include generating synthetic data points based on the user's existing data to provide a more comprehensive training dataset.
* Layer customization: In some cases, only specific layers of the neural network are personalized, while others remain shared across all users. This approach can be particularly effective in scenarios where certain aspects of the model need to be user-specific, while others can benefit from broader, global training.
* User feedback integration: Incorporating user feedback directly into the training process allows the model to adapt dynamically to changing user preferences and behaviors. This can be achieved through techniques like reinforcement learning, where the model learns and adapts based on user interactions.

Challenges in local model personalization:

* Resource constraints: Personalizing models on individual devices requires computational and storage resources, which might be limited, especially in mobile or IoT devices.

  - Data quality and diversity: The quality and diversity of local data can significantly impact the effectiveness of personalization. Ensuring that the model can handle a variety of data types and qualities is essential.
  - Privacy preservation: Even though the data do not leave the device, ensuring that the personalization process itself does not compromise user privacy is crucial.

  Advancements in lightweight machine learning models, efficient on-device training algorithms, and privacy-preserving techniques will be key to enhancing local model personalization. Research into optimizing these elements can lead to more effective and user-friendly personalized experiences in FLSs.
  Local model personalization in FLSs represents a critical step towards creating more user-centric and efficient learning models. By leveraging local data to tailor models to individual user needs, FLSs can provide more relevant, accurate, and privacy-preserving services. This personalized approach not only enhances user experience but also drives the effectiveness and adaptability of learning models in diverse real-world scenarios.

- Outlier handling algorithms: Handling outliers is a crucial aspect of data analysis and statistical modeling. Outliers are data points that significantly deviate from the majority [44]. Algorithms within FLSs play a vital role in maintaining the precision and reliability of machine learning models, especially when dealing with noisy or aberrant data points. These methods focus on detecting and managing outliers, which are data examples that deviate substantially from the established norm. The presence of outliers within a dataset can introduce bias during the model training process, potentially compromising the accuracy of subsequent predictions. The management of outliers is of utmost importance in FL, which involves utilizing data from various heterogeneous sources. Once outliers are identified, they can be addressed through data cleaning, imputation, or robust model training techniques. FLSs enhance the performance and utility of models across numerous applications and user scenarios by successfully managing outliers, ensuring data quality and model reliability. Various techniques for detecting outliers, including statistical methods, clustering algorithms, and robust machine learning models, are utilized to find abnormal data points.
  - Statistical outlier handling methods: Statistical techniques are essential tools for addressing outliers within FLSs, offering a quantitative framework for detecting and effectively handling anomalies in data. Methods such as the Z-score, interquartile range, or Tukey's fences are commonly used to identify outliers by quantifying their deviation from the dataset's mean or median. Through the application of statistical metrics, FLSs can pinpoint data points that significantly deviate from the established norm, signifying their potential classification as outliers. Once identified, these outliers can be managed using techniques such as data imputation, transformation, or exclusion to prevent them from unduly affecting the collaborative model training process. Methods for controlling statistical outliers offer a systematic and objective approach to preserving the integrity of data utilized in FL, thereby enhancing the precision and reliability of the resulting machine learning models.
  - Clustering outlier handling algorithms: Clustering algorithms are efficient tools for managing outliers in FLSs, especially when dealing with diverse and heterogeneous data sources. These methods facilitate the clustering of data points that exhibit similarities, allowing the detection and analysis of patterns inherent in the data. Outliers, characterized by significant deviation from the norm, frequently show unusual clustering patterns, making their identification more straightforward. FLSs can effectively detect outlier clusters using clustering algorithms such as k-means, hierarchical clustering, or DBSCAN. Clustering techniques aid in handling outliers within FLSs, providing a data-driven and adaptable approach.

This ensures the robust and accurate collaborative training of models, regardless of the diversity of data sources and patterns.

– Robust aggregation machine learning algorithms: Robust aggregation algorithms play an essential part in FLSs by effectively managing outliers, particularly in scenarios with noisy or inconsistent data originating from multiple sources. These algorithms are designed to minimize the impact of outliers on the aggregation process, ensuring that inaccurate or deceptive data points do not significantly distort the overall model. The use of robust aggregation strategies helps mitigate the influence of outliers during the model aggregation phase. Techniques like the trimmed mean, median-based aggregation, or methods derived from robust statistics are effective in achieving this objective. FLSs can thus maintain the integrity of the shared model, even when confronted with outliers, by reducing the significance of extreme or incorrect updates originating from individual devices. Robust aggregation algorithms are of utmost importance in enhancing the robustness of FL models. These algorithms guarantee that the resulting model accurately captures the collective intelligence of the devices involved, even in scenarios where the data are contaminated with noise or anomalies.

In summary, statistical techniques offer a straightforward and comprehensible approach, albeit potentially lacking in their ability to handle intricate data distributions effectively. Clustering algorithms can uncover subtle patterns within datasets but may be sensitive to parameters and initialization. Robust aggregation methods have been purposefully developed to address the presence of outliers during the process of model aggregation in FLSs, thereby guaranteeing the creation of a more dependable and resilient global model. Table 7 is a comparison table of some common techniques for outlier detection in FLS, including statistical methods, clustering algorithms, and robust machine learning models. The selection of an outlier handling method frequently relies on the data characteristics and the specific requirements of the FLS.

Table 7. A comparison table of some common techniques for outlier detection in FLS.

Technique	Approach to Outlier Detection	Advantages	Disadvantages	Typical Applications
Statistical methods	Use statistical metrics (like Z-score, IQR) to identify data points that deviate significantly from the norm.	Simple to implement; effective for univariate data.	Can be less effective with complex, high-dimensional data.	Data with a well-defined statistical distribution.
Clustering algorithms	Group similar data points together; outliers are points that fall outside clusters.	Effective in identifying groups and anomalies in multi-dimensional space.	May misclassify outliers as a separate cluster, requires determination of the number of clusters.	Multi-dimensional data with distinguishable clusters.
Isolation forest	Isolates anomalies by randomly selecting features and splitting values; outliers are easier to isolate.	Efficient for high-dimensional datasets; low linear time complexity.	Random forest mechanism may lead to inconsistent results.	Large datasets with many features.
Autoencoders (NN)	Neural networks trained to reconstruct input data; outliers are data with high reconstruction error.	Effective in capturing complex, nonlinear relationships in data.	Requires substantial data for training; computationally intensive.	Complex datasets with intricate patterns.

Table 7. *Cont.*

Technique	Approach to Outlier Detection	Advantages	Disadvantages	Typical Applications
Robust ML Models	Models that are less sensitive to outliers, like Random Cut Forest or models with regularization.	Can handle outliers while performing the primary learning task.	May require careful tuning; could ignore subtle but important anomalies.	Scenarios where model robustness is crucial.

This taxonomy offers a structured framework for comprehending and classifying the primary distinctions and factors that must be considered when dealing with FLSs. Depending on the specific use case and context, FLS implementations may vary significantly along these dimensions. Understanding these variations is essential for the proper development and deployment of FL solutions.

2.3. An Overview of Intrusion Detection Systems

The IDS is a vital cybersecurity tool specifically developed to observe and evaluate network traffic to identify any malicious activity or breaches of established policies. The system functions as a diligent protector, continuously monitoring the network environment for atypical patterns or behaviors that could signify a security breach or unauthorized entry [45]. Upon detecting suspicious activity, the IDS provides alerts or notifications, facilitating IT professionals' rapid investigation of and response to security issues. In the realm of network security, IDSs assume a pivotal role by enhancing the overall protection of computer networks. These systems enable enterprises to promptly identify and counteract potential cybersecurity threats, thereby fortifying the security of sensitive data and upholding the integrity of computer systems.

2.3.1. Types of Intrusion Detection Systems

IDSs can be classified based on their focus areas, deployment strategies, and detection techniques. The two primary types of IDS are the host intrusion detection system (HIDS) and the network intrusion detection system (NIDS) [45].

- HIDS: A HIDS is a critical cybersecurity component that focuses on monitoring and protecting specific hosts or devices within a network. It operates directly on endpoints such as servers, workstations, or other devices, analyzing local activities and configurations. HIDS identifies signs of malicious actions by comparing observed activity to predefined security regulations or baselines [46]. These activities may include unauthorized access attempts, file alterations, and unusual processes. HIDS employs methods like log analysis, file integrity verification, and real-time system monitoring to detect potential security issues. If suspicious actions are detected, HIDS generates notifications, alerting administrators to investigate and take appropriate actions to protect individual devices and their stored data. HIDS is particularly useful in environments where safeguarding specific hosts from internal and external threats is paramount.
- NIDS: NIDS is a cybersecurity solution that monitors and analyzes network traffic for indicators of malicious activity or potential security concerns. Unlike host-based systems, NIDS operates at the network level, analyzing data packets as they traverse the network. NIDS is strategically placed at critical points throughout the network, passively observing and analyzing all incoming and outgoing traffic in real time. It generates alerts when it detects suspicious trends, allowing security teams to promptly investigate and respond to potential security incidents. NIDS is especially beneficial for securing large and complex networks

2.3.2. Intrusion Detection Approaches

In the field of cybersecurity, IDSs utilize a range of methodologies to detect and counteract potential security breaches. The primary IDS approaches include the following:

- Signature-based detection: An essential component of an IDS involves comparing known attack patterns, often referred to as signatures, with incoming network traffic or system actions. If there is a match between the observed data and a saved signature, the IDS generates an alert indicating a potential security breach. This method efficiently recognizes well-known attacks that have been documented in the past, including various forms of malware, viruses, and infiltration attempts. However, its most significant limitation is its inability to identify novel or zero-day attacks. These types of security threats exploit vulnerabilities unknown to security professionals. Despite this limitation, signaturebased detection remains a vital part of any comprehensive security strategy. When used as one component of a layered security approach, it can be combined with other detection approaches, such as anomaly-based detection.
- Anomaly-based detection: IDSs use this sophisticated method to identify anomalous patterns or behaviors within the network traffic or system operations. Anomaly-based detection establishes a baseline of normal behavior by examining historical data to create a reference point, rather than relying on pre-defined attack signatures. It identifies any behavior that deviates from this baseline, such as unexpected patterns of network traffic or actions that are atypical for the system, as a potential security threat. ML algorithms are frequently utilized to analyze large datasets, detecting subtle variations that may indicate a security breach. Because it is highly effective at identifying entirely new types of attacks, anomaly-based detection is an essential component of contemporary cybersecurity methods. However, it requires accurate baselines and ongoing tuning to minimize false positives and negatives, maximizing the likelihood of identifying serious threats while reducing interference with legitimate network operations.

2.3.3. Internet of Vehicle Intrusion Detection

Within the dynamic and constantly changing domain of the IoV, IDSs play a vital role as digital protectors, safeguarding the resilience of interconnected vehicular networks against an increasingly diverse range of cyberattacks. Fundamentally, an IDS in an IoV setting entails a multifaceted approach that involves behavioral analysis, signature-based detection, and anomaly-based detection. Behavioral analysis is a fundamental aspect that involves careful observation and a comprehensive understanding of the complex patterns exhibited by vehicle behavior and network connections [47]. By effectively distinguishing between typical and atypical behaviors, the system can immediately detect deviations, therefore flagging possible intrusions or harmful operations. Simultaneously, signature-based detection functions as the initial layer of protection. This approach entails comparing incoming data with an extensive database of identified attack patterns. When a match occurs, it initiates an alert, facilitating prompt remedial action. Anomaly-based detection, a more advanced technique, creates baselines of typical behavior. When anomalies—such as atypical data traffic or unauthorized system access—are identified, alerts are sent, facilitating proactive measures to address potential security risks [48].

Furthermore, within the context of the IoV, ensuring the security of vehicle-to-everything communication is critical. Establishing robust cryptographic protocols is necessary to safeguard the complex communication network between vehicles and outside entities. These protocols play a crucial role in guaranteeing the secrecy, integrity, and validity of the data being communicated. Incorporating physical and cybersecurity measures provides an enhanced level of safeguarding. The detection systems for physical tampering serve to notify the IDS of potential threats, facilitating proactive cybersecurity measures [6]. By harnessing the capabilities of machine learning-based detection, IDSs can dynamically adjust and evolve. Machine learning algorithms, specifically deep learning models, can analyze extensive datasets obtained from car sensors and network interactions. This enables

the detection of subtle patterns that can serve as indicators of cyber risks, including those previously unidentified.

Significantly, the implementation of real-time threat response mechanisms dramatically enhances the effectiveness of IDSs in IoV. Real-time notifications, activated by irregularities or suspected breaches, are received by individuals inside the vehicle, managers overseeing the fleet, and centralized monitoring systems. These notifications prompt swift and targeted actions, including measures such as network segment isolation and emergency protocol activation. These actions effectively contain threats and safeguard the overall integrity of the network. The IDSs employed in the IoV encompass a complex integration of several components, including behavioral analysis, pattern identification, cryptographic techniques, machine learning capabilities, and instantaneous reactions. These technologies ensure secure data transmission inside the IoV and protect the safety, privacy, and trust of all individuals connected to this complex vehicular network. In doing so, they strengthen the fundamental basis upon which the future of transportation technology relies.

3. State of the Art

In this section, we present a well-organized literature review on IDSs based on FL in the IoV environment. This review aims to identify the latest advancements in FL-based intrusion detection within the IoV domain, covering the years from FL's inception in 2016 to 2023.

3.1. Intrusion Detection Systems Based on Federated Learning

The emergence of IDSs that utilize FL represents a significant advancement in cybersecurity. This innovative technique ensures the security of networked environments while upholding data privacy [26]. Unlike conventional IDSs that depend on centralized data analysis, FL-based IDSs operate on a decentralized principle. Within this innovative framework, each device independently generates localized ML models by leveraging their own data inputs. These models are subsequently improved through a collaborative learning process, where devices communicate changes to the models rather than exchanging raw data [49]. Ongoing research efforts continuously enhance this approach, leading to the emergence of FL-based IDSs as a potential future in the pursuit of secure and privacy-conscious network defense mechanisms [46].

Motivation to Adapt Federated Learning in Intrusion Detection Systems

The incorporation of FL into IDSs is driven by the significant demand for heightened security and privacy in our increasingly interconnected society. Despite the notable advancements made by ML and DL in the field of IDSs, various limitations associated with these technologies must be acknowledged, particularly concerning data privacy and communication efficiency. FL addresses these challenges by facilitating localized model training without compromising the privacy of raw data, thereby safeguarding individual privacy while promoting collaborative learning.

FL facilitates decentralized, real-time threat detection in contexts such as the IoT or IoV, where various geographically scattered devices generate data. The IDS's capacity to adapt to local contexts allows it to detect and recognize distinct threats peculiar to individual environments. The motivations for implementing FL in IDSs revolve around several essential elements, including the following [45]:

- Privacy preservation: FL enables collaborative model training while ensuring the privacy of sensitive raw data. Data privacy is of utmost importance in contexts where it holds significant value, such as the healthcare, finance, or government sectors. FL guarantees the protection of individual privacy by maintaining data locally and exchanging model updates. This approach aligns with legal and ethical requirements around privacy.
- Data efficiency: Data efficiency is a significant concern in conventional centralized systems, as transmitting substantial amounts of raw data to a central server may prove

unfeasible. This is particularly true when there are constraints on available bandwidth or communication costs are high. FL addresses this issue by focusing on lowering the volume of data transferred. Specifically, only updates to the model are exchanged, resulting in a substantial reduction in communication overhead.

- Adaptability and customization: The adaptability and customization of FL models allow for their adaptation to specific local settings. In the IDS field, various contexts may encounter distinct and specific threats. FL permits individual devices to customize their intrusion detection models based on their unique threat landscapes, ensuring precision in identifying potential threats.
- Continuous learning: Continuous learning is essential in the security field as threats perpetually evolve. FL permits the ongoing updating of models as new data become accessible. The capacity to adapt in real time ensures that IDSs remain effective in the face of developing threats, providing a significant advantage in dynamic situations.
- Robustness and fault tolerance: The inherent robustness of FL systems is based on their ability to withstand and recover from faults. In the event of a device failure or offline status, the system can maintain operation by utilizing the remaining functional devices [37]. The maintenance of fault tolerance is of the utmost importance in guaranteeing uninterrupted intrusion detection capabilities inside diverse and large-scale networks.
- Decentralization and edge computing: The utilization of FL facilitates decentralized learning, which aligns with edge computing principles, wherein data processing occurs in close proximity to its origin. In scenarios like IoT or IoV, where devices are dispersed geographically, FL enables localized learning, ensuring prompt reactions to potential risks without dependence on a central server.

These elements make FL a compelling and viable approach for enhancing the efficacy and confidentiality aspects of IDSs in diverse settings.

3.2. Related Surveys

A few reviews have focused on the topic of FL-based IDSs. Table 8 succinctly outlines the primary differentiators between our work and the previously conducted surveys. For instance, ref. [45] offers a comprehensive survey of FL-based IDS approaches and discusses the difficulties and challenges of using these methods. This review also outlines potential future directions for FL in IDS. Meanwhile, the authors of [27] focus on the current scientific progress of FL applications in attack detection problems for IoT and explore these applications. The extensive review presented in [50] draws from an analysis of 39 research papers published from 2018 to March 2022, with a specific focus on the IoT. The analysis examined evaluation variables related to IoT, particularly concerning FL, and identified and dis-cussed prospects and unresolved issues pertaining to FL-based IoT. The authors of [25] also provided an overview and comparison of six studies that use FL to enhance IDS effectiveness for IoT. In the absence of specific datasets for assessing FL, the authors emphasized data partitioning modeling among clients. Additionally, they investigated the modeling of bias in the test data to assess its impact on the effectiveness of the ML model. The authors of [51] discussed the implementation of FL-based IDSs in various domains and highlighted distinctions between different architectural configurations. Their structured literature analysis offers a reference architecture that can be used as a set of principles for comparing and designing FL-based IDS. Despite significant progress in FL for IDS development, a comprehensive survey exploring FL for IDS applications within the context of IoV is conspicuously lacking. To the best of our knowledge, no survey has thoroughly evaluated existing IDSs based on FL for IoV. In this direction, we present an organized literature analysis that examines recent developments in IDSs based on FL in an IoV environment. The review covers the years from 2016 (when FL was first introduced) to 2023. We conducted our search using the terms “federated learning”, “intrusion detection”, and “internet of vehicles”.

Table 8. Summary of related surveys on Federated Learning-based IDS.

Survey Title	Year	Main Focus	Key Contributions	IDS	IoV
Survey [45]	2021	FL-based IDS	Discussion on the role of FL in intrusion detection - Comprehensive review of ML/DL/FL in intrusion detection - Highlighting open research challenges	✓	X
Survey [27]	2022	FL in IDS within (IoT) domain	Understanding of federated learning, privacy preservation, and anomaly detection in network systems, with a particular focus on applications in IoT and related domains.	✓	X
Survey [25]	2022	FL-based IDS	- Review of FL system architectures - Review of Evaluation Datasets - Comparative analysis of proposed systems Open challenges and future directions	✓	X
Survey [50]	2022	FL-based IoT	Organizing and reviewing FL-based IoT domains - Creating a taxonomy to organize various aspects of FL-based IoT Providing some research questions about the FL-based IoT area and answering them Reviewing evaluation factors Focusing on open issues and future research challenges	X	X
Survey [51]	2022	FL-based IDS	Review of FL application in attack detection and mitigation Proposal of a reference architecture Establishment of a taxonomy Identification of open issues and research directions	✓	X
Our Survey	2023	FL-based IDS in IoV environment	Offer of a generic taxonomy for describing FL systems A well-organized literature review on IDSs based on FL in an IoV environment. Highlighting challenges and potential future directions based on the existing literature.	✓	✓

Note: In this table, ✓: indicates that the survey discussed the relevant aspect of Federated Learning (FL) or Intrusion Detection Systems (IDS), while X signifies that the aspect was not discussed in the survey.

3.3. Comparative Analysis of Federated Learning-Based Intrusion Detection Systems for Internet of Vehicles

In the rapidly evolving landscape of cybersecurity within IoV, FL has emerged as a transformative paradigm, promising enhanced security and privacy preservation. As the IoV ecosystem expands, robust IDSs become essential to safeguard vehicles, passengers, and the underlying network infrastructure from ever-evolving cyber threats. This section offers an extensive analysis of the relevant literature in the field of IDS based on FL, specifically tailored to the intricacies of IoV. This comparative survey aims to extract significant insights by examining the unique techniques, strategies, and structures of recent studies. These insights are crucial for understanding the current state of IDS solutions based on FL in IoV and provide valuable guidance for future research. We employed a range of criteria to evaluate and differentiate the related works in the domain of FL-based IDSs within the framework of IoV. We formulated the following research questions for our review:

- What kinds of FL designs are used for IDS?
- What ML model architectures are employed in the proposed solutions?
- Which datasets are utilized for evaluating the proposed solutions?
- What types of attacks can be identified by the proposed solutions?
- Which measures do the authors employ to validate their proposed solutions?
- Which communication patterns are utilized in the solutions they offer?
- Do the proposed solutions operate in synchronous or asynchronous mode?
- Which aggregation model do the proposed solutions utilize?
- Which optimization algorithms do the proposed solutions utilize?
- Are the proposed solutions designed to support real-time processing?
- Are the proposed solutions designed to support imbalanced data distribution?
- What is the impact of the implemented solutions on overhead costs?

Based on the formulated questions, we considered the following criteria during our review of the papers to organize the information in a structured manner that allowed for easy comparison and understanding:

- Year of publication;
- Datasets used;
- Attacks detected;
- ML models;
- Communication patterns;
- Communication synchronization;
- Evaluation metrics;
- Model aggregation algorithms;
- Optimization algorithms;
- Real-time considerations;
- Data distribution;
- Communication overhead.

While FL-based IDSs for IoV are the primary focus of this paper, we did not conduct any experiments on the reviewed approaches to evaluate them. The study aimed to highlight open difficulties and research directions by considering the described factors. Table 9 provides a summary of the comparison's results.

Table 9. Comparative Analysis of Federated Learning-based intrusion detection systems for IoV.

Ref.\Year	Dataset	Attacks Detected	ML Model	Communication Patterns	Communication Synchronization	Evaluation Metrics	Model Aggregation	Optimization Algorithms	Real-Time Processing	Data Imbalance	Overhead
[52], 2022	The attack-free dataset of CAN messages published by the HCR Lab of Korea University	-Spoofing -Replay -Drop -Denial-of-Service (DoS)	Convolutional Long Short-Term Memory (ConvLSTM) model	Client–server mode	Synchronous mode	-FPR, TPR -Accuracy -Precision -Recall -F1-score	Secure MultiParty Computation	The Federated Proximal Algorithm	Real-time processing	Imbalanced data distribution.	Reduces the overhead
[53], 2021	VeReMi dataset	-Constant attack -Constant offset attack -Random attack -Random offset attack -Eventual stop attack.	Long Short-Term Memory (LSTM) neural network.	Client–server mode	Synchronous mode	-Precision -Recall -Accuracy	Federated Averaging Algorithm (FedAvg)	Federated Stochastic Gradient Descent (Federated SGD)	Batch processing	Imbalanced data distribution.	Reduces the overhead
[54], 2022	Simulated dataset	Black hole attack	Random Forest 1-dimensional CNN (1-D CNN) 1-dimensional RNN (1-D RNN)	Client–server mode	Synchronous mode	-Precision -Recall -F1-score	Weighted aggregation model	_	Batch processing	_	_
[55], 2023	VeReMi dataset	-Constant attack -Constant offset attack -Random attack -Random offset attack -Eventual stop attack.	Deep neural networks	Federated Arch. with edge offloading	_	-Accuracy -Consensus time -Incentive mechanisms	Federated Averaging Algorithm (FedAvg)	Federated Stochastic Gradient Descent (Federated SGD)	_	_	Reduces the overhead
[56], 2023	The simulated Sybil attack dataset	Sybil attack	_	Client–server mode	Synchronous mode	Accuracy Number of global aggregations	Weighted aggregation model	Fuzzy logic-based technique	Batch processing	_	Reduces the overhead
[16], 2022	Car Hacking dataset	-Flooding -Spoofing -Replay -Fuzzing	Gated Recurrent Unit (GRU) with a Random Forest (RF)-based ensembler unit.	Client–server mode	asynchronous mode	-Accuracy -Precision -Recall -F1 score	Federated Averaging Algorithm (FedAvg)	Adam optimizer	Batch processing	_	_
[57], 2021	CAN-Intrusion dataset (OTIDS)	-DoS attack -Fuzzy attack -Spoofing attack	Random Forest	Client–server mode	_	-Accuracy -Precision -Recall	_	_	Batch processing	_	_
[58], 2022	Car Hacking dataset	-DoS attack -Fuzzy attack -Spoofing attack	Multilayer Perceptron (MLP) model	Client–server mode	_	-Accuracy -Loss -AUC score -Time Cost	Federated Averaging Algorithm (FedAvg)	Stochastic Gradient Descent (SGD) optimizer	Real-time processing	_	_

Table 9. *Cont.*

Ref.\Year	Dataset	Attacks Detected	ML Model	Communication Patterns	Communication Synchronization	Evaluation Metrics	Model Aggregation	Optimization Algorithms	Real-Time Processing	Data Imbalance	Overhead
[59], 2022	Practical dataset	-Spoofing attacks -Replay attacks -Drop attacks -DoS attacks	Long Short-Term Memory (LSTM) neural network.	Client–server mode	_	-The detection accuracy	_	_	Real-time processing	_	_
[60], 2023	Car Hacking dataset	-DoS attack -Fuzzy attack -Spoofing attack	Convolutional Neural Network (CNN)	Client–server mode	_	-Accuracy -Recall -Precision -F1-score	Federated Averaging Algorithm (FedAvg)	Bayesian Optimization (BO)	Real-time processing	Imbalanced data distribution.	Reduces the overhead
[47], 2023	NSL-KDD dataset	-DoS attack -Probe attack -R2L (Remote to Local) -U2R (User to Root)	Memory-Augmented Autoencoder Model	Client–server mode	Synchronous mode	-Accuracy -Precision -Recall -F1 score	Weighted Aggregation Model	Adam optimizer	Batch processing	Imbalanced data distribution.	_
[61], 2023	VeReMi Extension dataset	-Constant attack -Constant offset attack -Random attack -Random offset attack -Eventual stop attack.	Long Short-Term Memory (LSTM) neural network.	_	_	-F1-scores	_	_	_	_	_
[62], 2023	The dataset [RAKGZ20]	-SYN flood attack -UDP flood attack	The deep autoencoder method	Federated Arch. with edge offloading	_	-F1-Score -The false positive rate (FPR)	Federated Averaging Algorithm (FedAvg)	Federated Averaging Algorithm (FedAvg)	_	_	_
[63], 2022	CAN-Intrusion dataset (OTIDS)	-DoS attack -Fuzzy attack -Impersonation attack	Statistical Adversarial Detector		_	-Maximum Mean Discrepancy (MMD) -Energy distance (ED)	_	_	Batch processing	_	_
[64], 2023	The CIC-IDS 2017 dataset	DoS attack, web attacks, port scan, bot, brute force attacks	A Cat Boost model	Client–server mode	_	Precision, recall, Kappa score, accuracy	The Bagging Classifier technique	The grid search method	_	Imbalanced data distribution	_

3.4. Analysis and Discussion

The analysis of the research papers aided us in formulating the following conclusions:

- Dataset: The selection of a dataset is a crucial aspect when evaluating the effectiveness and resilience of proposed solutions in the field of IDS based on FL within the context of IoV. Given the dynamic and complex nature of IoV, it is imperative to use datasets that can accurately depict real-world vehicular communication scenarios, encompassing both normal and malicious activities. These datasets play a fundamental role in training and evaluating IDS models, enabling them to effectively identify threats within the IoV environment. The following describes the datasets utilized in the provided papers to assess the efficacy of various IDS solutions. Three of the papers, namely [52,57,63], employed the CAN-intrusion dataset (OTIDS), which was sourced from the Hacking and Countermeasure Research Lab at Korea University. This dataset provides a comprehensive representation of intrusion scenarios within in-vehicle networks, making it suitable for assessing IDSs specifically designed for vehicular contexts. By contrast, refs. [53,55,61] employed the VeReMi dataset for their experimental analysis. The publicly accessible VeReMi dataset was explicitly developed for analyzing mechanisms to detect misbehavior in VANETs. The authors of [16,58,60] employed the Car-Hacking dataset derived from the "Car Hacking: Attack & Defense Challenge" competition held in 2020. Additionally, some papers used simulated datasets, such as [54], where a simulated dataset was employed to evaluate the effectiveness of their proposed approach in vehicle-to-vehicle and ve-hicle-to-infrastructure scenarios. The authors of [56] employed a simulated attack dataset consisting of simulated Sybil attack flows and normal traffic flows in their experimental analysis. Meanwhile, the simulations in [59] were conducted using the authors' proprietary dataset. Although the NSL-KDD and CIC-IDS 2017 datasets are not dedicated to IoVs and are primarily general intrusion detection datasets, the authors of [47,64] conducted their experiments on these datasets to evaluate the performance of their proposed methods. Finally, ref. [62] utilized the [RAKGZ20] dataset to evaluate the authors' proposed solutions. These datasets collectively offer a comprehensive view of various intrusion detection scenarios, particularly within automotive networks.
- Attacks detected: Within the domain of FL-based IDSs for IoV, numerous research papers have put forth methodologies to identify a diverse range of cyber threats. DoS attacks [47,52,57–60,63] and constant attacks [53,55,61] are the most frequently discussed types of attacks in the literature. In addition, some authors emphasized specific attacks, such as the Sybil assault [56] and the black hole attack [54]. Several papers also explored detecting advanced attacks in in-vehicle networks, including adversarial attacks like fuzzy attacks [16,57,58,60,63], flooding attacks [16,62], and spoofing attacks [16,52,58–60]. These studies highlighted the diverse and persistent nature of cyber threats in the IoV environment, underscoring the critical need for robust IDS solutions. IDSs based on FL in IoV not only demonstrate the adaptability and robustness of FL techniques but also illustrate the essential role these techniques play in protecting the future of connected vehicular systems against a wide array of cyberattacks.
- ML models: Researchers have turned to more powerful ML models to construct resilient FL-based IDSs capable of addressing challenges posed by vehicular networks. These models, tailored to meet the unique requirements of vehicular communication, offer promising ways to detect and mitigate potential attacks. To improve detection capacities and ensure vehicular safety, numerous ML models based on FL in IoV have been implemented in the field of IDS. The following summarizes the ML models utilized in the proposed solutions across the reviewed papers.
  - Long short-term memory (LSTM): This architecture of recurrent neural networks is prominently featured in articles [16,52,53,59,61]. One notable advantage of this

approach is its proficiency in identifying patterns over different time intervals, making it well-suited for analyzing time-series data such as network traffic.
  - Deep convolutional neural network (DCNN): Papers such as [55,60] utilized DCNNs to effectively handle structured grid data, including images or time-series data. These DCNNs possess the capability to automatically and adaptively learn spatial hierarchies.
  - Support vector machine (SVM): ref. [60] utilized SVM, a supervised ML approach applicable to both classification and regression tasks.
  - Statistical adversarial detector: As explicitly stated in [63], this approach employs statistical techniques to identify adversarial examples.
  - Random forest: refs. [54,57] employed the random forest algorithm, an ensemble learning technique. This algorithm constructs numerous decision trees during the training phase and determines the class output by selecting the mode of the classes for classification.

The utilization of a wide array of ML models in the articles highlights the intricate and multifaceted characteristics of intrusion detection in IoV. Researchers have used diverse techniques, such as recurrent networks like LSTM, capable of capturing temporal relationships, and ensemble methods like random forest, which provide robustness. These approaches enhance the security and dependability of vehicular networks.

- Communication patterns: Most of the articles we reviewed provided solutions formulated according to the client–server mode of operation, as exemplified by [47,54,58,60,64], among others. In this mode, clients engage in the process of training their models on a local level without sharing raw data. Subsequently, the model updates are transmitted to the server, the central entity responsible for aggregating them. This procedure guarantees the protection of data privacy and minimizes the necessity of data centralization. Meanwhile, some papers adopted a federated architecture with an edge-offloading technique [55,62]. As mentioned above, this approach diminishes latency and reduces dependence on a remote cloud server. As discussed in the publications mentioned above, the client–server mode of operation emphasizes the shifting paradigm of decentralized data processing in IoV. FL-based IDSs not only protect users' data privacy but also pave the way for more effective and scalable security solutions in rapidly developing vehicular networks. These systems enable vehicles to train models locally, with central servers aggregating the training results.
- Communication synchronization: The communication synchronization mode, whether synchronous or asynchronous, significantly impacts the efficiency and effectiveness of the FL process. Ref. [52] discussed the operational characteristics of synchronous FL, which involves a single launch point and a single aggregate point for the global model. In this model, the beginning of each iteration occurs concurrently for all clients, and the federated aggregation process is performed without establishing a predetermined objective for the learning rounds. In [53], the authors presented a synchronous FL approach, and ref. [54] introduced a conventional synchronous FL protocol. This protocol is considered appropriate for a wide range of FL scenarios, including those involving bottlenecks. On the other hand, ref. [16] preferred an asynchronous mode, which can provide greater flexibility in dynamic settings and effectively handle frequent model changes and bottlenecks. This strategy enables increased adaptability in the learning process, accommodating partial updates from clients that may impact convergence performance. Nevertheless, not all research explicitly addressed this matter. Most of the publications did not specify their operational mode concerning synchronization. The variations mentioned above highlight the varied approaches that researchers have utilized to enhance the effectiveness of IDSs within the rapidly changing environment of IoV. In summary, while synchronous FL was a prevalent technique in the suggested solutions, some studies acknowledged the advantages of asynchronous methods, particularly in environments characterized by frequent updates and potential bottlenecks.

- Evaluation metrics: In most of the papers that were reviewed, the evaluation of the efficacy of FL-based IDS systems relied on ML measures that assess the effectiveness of the analytic model. These metrics include accuracy, precision, recall, and F-measure. A limited number of research publications examined the effects of FL. In particular, ref. [55] discussed the consensus time, which is impacted by the quantity of FL workers and the number of created blocks. The study additionally assessed the effectiveness of the FL-enabled edge node by manipulating the reward and accuracy of the local model. This evaluation considered various elements, including the reward, energy consumption, and processing overhead. Moreover, the researchers did not overlook the significance of accuracy as a fundamental measure for evaluating the efficacy of their proposed solution. The paper also addressed the issues associated with recruiting FL workers, highlighting the possibility of bias and imbalance when selection is primarily predicated on reputation. The authors proposed various strategies to address these difficulties, such as including randomization in the selection procedure. In addition, in [56], the authors considered the "number of global aggregations (NGA)" as an evaluation metric. They presented information regarding the number of global aggregations performed in the proposed system and other state-of-the-art baseline frameworks. Their research demonstrated how many global aggregations are necessary for different numbers of communication rounds (R) to achieve the desired level of accuracy. The FLEMDS framework proposed in the study necessitates a reduced number of global aggregations in comparison to the baseline frameworks to attain a comparable level of accuracy.
- Aggregation model: In the domain of distributed ML, the combination of data or model updates from several nodes holds significant importance in determining the overall performance and efficiency of the system. The aggregation process has been extensively explored in contemporary research, with numerous novel approaches and models offered in recent research papers. These aggregation models aim to successfully harness the collective intelligence of all participating nodes while simultaneously overcoming problems such as data heterogeneity, communication overheads, and adversarial threats.

  The examined literature suggested a range of aggregation models to improve the effectiveness and precision of distributed systems, particularly in the domain of FL. One of the most common aggregation models used in the reviewed papers is the federated averaging method, where local model updates are averaged to produce a global model [16,53,55,58,60,62]. This approach is simple yet impactful, particularly in situations involving non-identically and independently distributed (non-IID) data [32]. An alternative methodology uses weighted federated averaging, as described in several papers [47,54,56]. This technique involves assigning varying weights to local models, considering factors such as the quantity of data samples or the quality of the model. Secure aggregation is another widely employed model aggregation technique in the field of FL, as observed in [52]. In this technique, various cryptographic techniques, including SMPC, are employed to consolidate data while preserving the confidentiality of the unprocessed updates. The authors of [64] used the Bagging Classifier technique as aggregation model in their developed solution. This technique aggregates the predictions of multiple models to produce a single, more accurate model. The resulting supermodel, created by the central server, exhibits better robustness than the individual edge device models.

  Each aggregation method provides specific benefits designed to address the challenges and requirements of dispersed learning settings. The models described above are at the forefront of current research, each tackling distinct issues. As technology advances and increasingly intricate situations arise, these models are expected to continue to develop, facilitating the implementation of more resilient and effective distributed learning systems. The ongoing investigation and advancement of aggregation models

serve as evidence of the dynamic characteristics of ML research and its dedication to optimizing the utilization of distributed nodes' collective intelligence.

- Optimization algorithms: The utilization of FL in IDSs presents a new and innovative method for addressing the issues related to data privacy and effective model training in IoV. Advanced algorithms play a pivotal role in optimizing FL models. For instance, the federated proximal algorithm has been used to fine-tune model parameters, ensuring optimal performance in detecting intrusions [52]. Similarly, some studies have adopted federated stochastic gradient descent (federated SGD) to optimize the parameters of the proposed IDS models [53,55,58]. Furthermore, some papers utilized other optimization techniques, such as the Adam optimizer [16,47], a fuzzy logic-based technique [56], Bayesian optimization (BO) [60], and the federated averaging (FedAvg) algorithm [62]. The authors of [64] used the grid search method for hyperparameter tuning as an optimization algorithm in their solution. This method is employed to optimize the Cat Boost model, a gradient boosting algorithm that utilizes decision trees as the classifier model for edge devices. The grid search technique exhaustively searches over a specified set of hyperparameters to improve the model's accuracy.
  The integration of optimization approaches, combined with the decentralized nature of FL, holds the potential to deliver resilient and effective IDSs for the IoV environment. Decentralizing the learning process and applying complex optimization algorithms not only enhances detection capabilities but also ensures that modern concerns regarding privacy and efficiency within the IoV landscape are effectively addressed. This represents a significant advancement for the industry. The ongoing expansion and development of IoV necessitate the use of innovative strategies to ensure the protection and security of our networked automotive environment.
- Real-time processing: A critical aspect of FL-based IDSs is their ability to process data in real time, ensuring timely detection and response to potential threats. Our review found several papers that proposed IDSs designed for real-time operation [52,58–60]. For instance, refs. [52,53,58] highlighted the significance of real-time processing for IDSs, especially when dealing with vehicular networks. In addition, ref. [60] introduced ImageFed IDS, a system designed for real-time inference. It employs a lightweight image-based feature extraction for CAN packets, making it suitable for real-time applications. On the other hand, some papers supported a batch processing approach rather than real-time processing [16,47,56]. Some papers did not explicitly mention whether their proposed solutions are designed for real-time or batch processing. Nevertheless, all the papers emphasized the importance of real-time processing in IDSs for IoV, with various solutions and methodologies proposed to achieve this objective. The operational significance of IDSs for vehicle networks increases as these networks undergo continuous evolution and encounter a diverse range of cyber threats. The research presented in these papers offers solutions and approaches that contribute to the development of a more secure and responsive IoV environment by emphasizing the significance of real-time processing.
- Data distribution: While imbalanced data distribution is a significant concern in ML and AI research, most of the research papers we reviewed did not address this aspect. We only identified five articles, namely [47,52,53,60,64], that specifically addressed the issues and implications associated with imbalances in data distribution in FL scenarios. They stressed the importance of dealing with this problem to achieve robust and stable model performance. The authors of [52] emphasized that in real FL contexts, the data distributed across many nodes or devices may exhibit non-IID characteristics. These characteristics sometimes arise due to an imbalanced distribution of data, wherein certain data classes may be overrepresented in one node while being underrepresented in another. To overcome this difficulty, the study suggested an IDS that uses FL to help handle imbalanced data distribution. The authors of [53] examined the vulnerability of models to adversarial attacks, particularly when confronted with data imbalance. The presence of an imbalance in vulnerability can be

exploited by adversarial examples, resulting in the misclassification of benign data. The authors presented various techniques for identifying these adversarial examples, indirectly addressing the difficulties associated with data imbalance. From another perspective, the authors of [60] showed that data distribution among vehicles in FL scenarios, particularly in the context of IoV, might exhibit a significant imbalance. This imbalance can potentially impact the overall performance of the global model. The paper introduced various methodologies aimed at alleviating the repercussions of this imbalance, thereby ensuring the robustness of the FL framework. The issues presented by imbalanced data distribution were also addressed in [47]. The authors highlighted the potential emergence of unexpected attack behaviors in the context of IoV development. The absence of comprehensive analysis and systematic gathering of various attack behaviors has resulted in an imbalanced distribution of sample data categories within intrusion detection for IoV. Consequently, this disparity leads to diminished accuracy in detection. The authors proposed an intrusion detection approach integrating FL and a memory-augmented autoencoder (FL-MAAE) to tackle this issue. They have considered the problem posed by imbalanced data distribution in their produced solution, hence ensuring the continued effectiveness of the model. Lastly, the proposed framework in [64] employs the Synthetic Minority Over-sampling Technique (SMOTE) to tackle the issue of class imbalance in the dataset. This approach of oversampling minority classes helps to create a more balanced dataset, which in turn allows for a more accurate and representative evaluation of the classification models. Addressing data imbalance is critical for guaranteeing the resilience and dependability of ML models, particularly in distributed learning scenarios such as FL.

- The overhead: One of the primary issues frequently encountered in the domain of IDSs based on FL is the significant overhead associated with these systems. The effectiveness and responsiveness of IDSs in IoV contexts can be significantly affected by overhead, including computing, communication, and storage expenses. Addressing this overhead is crucial to ensure the seamless operation of these systems without compromising their primary function of identifying and mitigating threats. Several of the reviewed papers examined the issue of overhead, which holds significant importance in the field of distributed systems and FL [52,53,55–57,60]. In [52], the term "overhead" refers to the complexity of the algorithms offered, and the authors stressed how important it is to reduce this complexity as much as possible to ensure efficient operations. In addition, ref. [53] discussed overhead in the context of communication costs, emphasizing the relevance of minimizing overhead to improve system performance. Overhead was explored in relation to the computing expenses of the proposed approaches in [55], which emphasized the necessity of striking a balance between accuracy and computational efficiency in the methods offered. The research presented in [56] investigated the overhead caused by the consensus process in blockchains and suggested that using a lightweight consensus method can reduce overhead and increase scalability. The topic of overhead was discussed in the context of data transmission in [57], which emphasized the significance of effective data-sharing systems to reduce overhead. Lastly, ref. [60] provided a comparative analysis of various solutions. This research suggested that FL approaches often incur less overhead than alternative distributed learning modes. The study also discussed processing overhead in the context of incentive mechanisms for FL. Taken together, these papers highlight the importance of properly managing overhead costs to guarantee the efficiency, scalability, and effectiveness of distributed and federated information systems.

Upon reviewing the collection of work relevant to IDSs based on FL in IoV, it becomes apparent that the realm of security within vehicular networks is experiencing a significant and fundamental change. FL has emerged as a promising solution for effectively addressing the intertwined issues of safeguarding data privacy and enhancing threat detection efficiency. In conclusion, Table 10 presents a comparative analysis of the advantages and drawbacks of each one of these proposed solutions that we discussed.

Table 10. Comparative analysis of FL-based IDS for IoV: advantages and drawbacks.

Ref	Advantages	Drawbacks
[56]	Three-level model aggregation. Fuzzy Logic-Based FL Vehicle Selection (FLBFLVS). Reduced latency.	Complex system architecture.
[52]	Reduced model size and convergence time. High detection accuracy (over 95%)	Complexity of implementation. Scalability concerns.
[53]	Privacy preservation. Reduction in communication overhead. Handling position falsification attacks.	Complexity in implementation. Challenges in federated averaging. Experimental limitations.
[54]	Trust estimator integration. Effective learning with fewer rounds. Improved network performance	Challenges in synchronization and model aggregation. Need for regular updates and maintenance.
[55]	Blockchain integration for trust. Smart contract use. Efficient consensus protocol. High performance.	Resource intensity. Challenges in worker selection and bias. Scalability in real-world deployment.
[16]	High accuracy in cyberattack detection. Reduced communication overhead. Resource efficiency.	Complex implementation.
[57]	Blockchain integration. Decomposition using Fourier transform. High performance.	Complex system architecture. Resource intensiveness. Challenges in blockchain integration.
[58]	High accuracy (up to 98.45%). Low-complexity structure. Adaptability.	Dependency on local data quality.
[59]	High detection accuracy (beyond 90%).	-
[60]	High performance metrics (with an average 99.54% F1-score and 99.87% accuracy, alongside low detection latency). Lightweight feature extraction.	-
[47]	Robust to imbalanced data. Effective in detecting unknown attacks.	The evaluation is conducted used the NSL-KDD dataset, which is not dedicated to IoVs and is primarily an intrusion detection dataset.
[61]	High accuracy in threat detection.	Complexity in tradeoffs between utility and'privacy.
[62]	Zero-day attack detection. High detection rates. Multi-access Edge Computing (MEC) assistance.	Complexity in implementation and management
[63]	Adversarial attack detection. Blockchain integration. High detection accuracy Lightweight feature extraction.	Computational overhead. Limitations in detecting certain adversarial attacks.
[64]	Robust to imbalanced Data. Handling class imbalance.	The evaluation is conducted used the CIC-IDS 2017, dataset which is not dedicated to IoVs and is primarily an intrusion detection dataset.

4. Discussion of Challenges and Future Research Directions

IoV is anticipated to experience significant growth in the coming decade, emerging as a prominent paradigm movement. This projection suggests that IoV will receive substantial attention and witness considerable advancements across several sectors and industries. The integration of FL into IDSs within IoV scenarios presents a significant opportunity to bolster the security of interconnected vehicles in this dynamic environment. The primary objective of incorporating collaborative intelligence concepts and technologies into the domain of

IoV is to facilitate the integration of data and resources from a vast array of vehicles, users, infrastructure, and networks. This integration aims to enhance the reliability, connectivity, and ease of management, control, and operation of IoV systems. Nevertheless, this novel methodology also presents a series of challenges that necessitate meticulous consideration to guarantee the efficiency and security of these systems. These constraints arise from the heterogeneous characteristics of vehicle data, constrained network resources, the persistent risk of adversarial assaults, strict regulatory obligations, and the imperative to uphold the security of FL models. The ability to effectively deal with these intricacies is crucial to fully harness the capabilities of FL-enabled IDSs in IoV scenarios. This will establish a resilient, secure, and privacy-conscious vehicular network. Drawing upon the literature analysis, this section aims to elucidate some of the main challenges we found and possible future research directions for investigating the development of IDSs empowered by FL within the IoV context. Figure 3 summarizes the challenges and future research directions in FL-enabled IDS IoV.

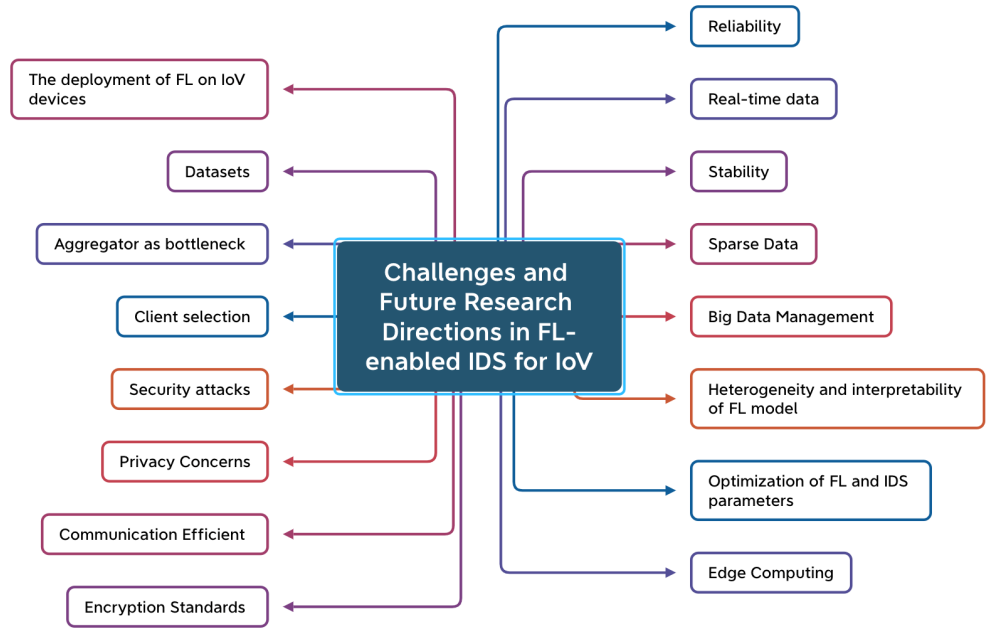

Figure 3. Challenges and future research directions in FL-enabled IDS for IoV.

Here, we cover some of the primary challenges and future research directions associated with developing FL-enabled IDSs in IoV scenarios.

- The deployment of Federated Learning on Internet of Vehicles devices: Deploying an FL-enabled IDS architecture on real IoV devices presents many challenges. One notable obstacle involves the presence of resource constraints since IoV devices frequently have restricted processing capabilities and memory capacities, making the efficient execution of intricate FL algorithms difficult. This challenge can be exacerbated when employing deep learning techniques, as they often require more computational resources than traditional ML [65]. To overcome these restrictions, a prevailing approach involves the implementation of intermediate nodes positioned at the network edge. These nodes serve as clients for FL, receiving data from end devices. Real-time processing poses an additional challenge in the context of IDSs in IoV. These IDSs need to effectively evaluate incoming data and promptly identify any instances of intrusion, requiring the implementation of algorithms that strike a delicate balance between accuracy and processing speed. Consequently, more work is needed to examine the real-world constraints of FL-enabled IDS techniques in IoV contexts to ensure optimal levels of security and efficiency.
- Limitations of existing FL-enabled IDS datasets for IoV: The current datasets available for FL-enabled IDSs in the context of IoV exhibit various limitations. The issue of

data diversity presents a notable obstacle as datasets may lack a comprehensive representation of the wide range of real-world scenarios and driving conditions, resulting in the development of biased models. Data imbalance is a significant issue that warrants attention, as specific categories of security threats may be inadequately represented in the dataset, posing challenges for the FL-enabled IDS to detect these less frequent intrusions accurately and efficiently. Data quality is essential, as any inaccuracies or noise present in the data can significantly impact the learning process, potentially leading to the development of intrusion detection models that are less reliable and potentially misleading. Furthermore, the issue of data privacy poses a significant constraint in the context of IoV. The data generated by IoV systems frequently encompass confidential personal and vehicular details, thereby presenting a formidable obstacle in creating extensive datasets that simultaneously safeguard users' privacy. The concern regarding the scalability of current datasets becomes particularly significant as IoV networks experience rapid expansion. These constraints must be acknowledged and addressed to create resilient IDSs that effectively capture the complicated nature of actual IoV settings while upholding user privacy and data integrity.

- Aggregator as a bottleneck: In the context of IoV scenarios involving FL-enabled IDS, the aggregator frequently becomes a bottleneck despite being a central component. The processing capacity of the aggregator can be overwhelmed by the sheer volume of incoming information if data from multiple vehicles are sent to the aggregator for model training and updating [65]. The influx of data, especially in extensive IoV networks, has the potential to result in delays when it comes to aggregating and updating FL models. Furthermore, given the real-time nature of intrusion detection in vehicle contexts, introducing any delay at the aggregator level can impede prompt responses to security threats. The challenge of balancing the requirement for comprehensive model updates with the practical constraints of aggregators is of utmost importance. This necessitates using innovative approaches in distributed computing, efficient algorithms, and optimized communication protocols. These measures are necessary to address the bottleneck and ensure the smooth operation of FL-enabled IDSs in IoV scenarios.
- Client selection: Identifying suitable clients for FL-enabled IDSs in the IoV context presents a significant challenge. During each training iteration, the coordinator can choose a specific subset of devices to engage as FL clients in the training procedure. The environments in which IoV operates exhibit a high degree of dynamism, characterized by the continuous movement of vehicles within and beyond the network coverage area. The dynamic nature of the environment poses difficulties in maintaining a consistent group of clients who actively participate in the training of FL models. For instance, specific devices may not be accessible during a particular round due to mobility issues or disruptions in connectivity. In addition, the criteria for selection need to consider factors such as the device's current state, its battery life, its computational and networking capabilities, and even the precision of the ML technique. The client selection process can significantly impact the accuracy achieved and, consequently, the detection of potential security breaches within the framework of an IDS approach. Striking a balance in the client selection process, where a diverse, accurate, and current dataset is maintained, necessitates the utilization of advanced algorithms and real-time decision-making to manage the ever-changing pool of participating vehicles effectively. Addressing this challenge is essential to maintain the integrity and accuracy of FL-enabled IDSs in IoV scenarios. Therefore, future strategies for devising an efficient client selection process in IoV systems must consider the dynamic nature of device conditions throughout each training iteration.
- Security attacks: In the context of FL-enabled IDSs in IoV scenarios, security attacks pose a severe threat. Attackers can exploit vulnerabilities inherent in the FL architecture [66]. These exploits can manifest as various types of attacks, including data

poisoning [24], where adversaries inject deceptive data into the training process to manipulate the IDS model [45]. Model inversion attacks can also occur, in which attackers attempt to deduce confidential data from the trained model. In addition, the confidentiality and integrity of data might be compromised by eavesdropping attacks that specifically target the communication channels established between vehicles and the central server. To address these security concerns, robust security measures are essential, including strong encryption, secure communication protocols, anomaly detection techniques, and continuous monitoring. Preserving security in FL-enabled IDSs within IoV scenarios is of utmost importance for protecting against a diverse range of potential cyberattacks and maintaining the efficiency of IDSs in interconnected vehicular networks.

- Privacy concerns: Privacy considerations emerge as a significant challenge in the context of FL-enabled IDSs in IoV scenarios. Although the primary purpose of FL is to address the privacy concerns associated with centralized learning methods, FL may still inadvertently disclose information from the training data of individual clients. FL relies on data provided by individual vehicles for the purpose of training models, raising issues concerning user privacy and data confidentiality. Within IoV, vehicles can generate substantial quantities of sensitive data, including location information, driving behavior, and recordings of communication. The central issue revolves around the need to effectively utilize this data for training IDS models while safeguarding the privacy of both vehicle owners and occupants. As a result, there has been a notable surge of interest has occurred in implementing privacy-preserving methodologies in the field of FL [23]. These methodologies include differential privacy techniques, SMPC, and homomorphic encryption. However, using these advanced approaches often entails a trade-off in terms of precision and effectiveness, potentially compromising the IDS's ability to identify attacks. Deploying these advanced methods is necessary to strike a balance between the need for effective intrusion detection, strict privacy requirements, and meeting user expectations. Further research is required to find the optimal balance between privacy and performance to develop efficient IDS methodologies.
- Communication efficiency: Implementing FL-enabled IDSs within IoV introduces a significant challenge in terms of communication efficiency. In IoV scenarios, where vehicles are in constant motion, transmitting substantial amounts of data to train FL models on a central server can strain network bandwidth and result in significant communication overhead. This challenge is further exacerbated by the need for real-time intrusion detection, where rapid responses are crucial. Optimizing communication protocols and data transmission techniques is essential to alleviate the network's burden while ensuring the timely delivery of relevant data to the central server for model updates. Future research in this field is oriented towards developing sophisticated communication-efficient techniques tailored specifically for IoV scenarios. Approaches such as model quantization, edge computing, and strategic data sampling can be leveraged to minimize the volume of transferred data, thereby enhancing communication efficiency. Balancing the requirement for extensive data exchange with the constraints imposed by network bandwidth is essential for the effective implementation of FL-enabled IDSs in dynamic and bandwidth-limited IoV environments. Research efforts also focus on exploring 5G and beyond-5G technologies, which hold the potential to provide increased bandwidth and reduced latency. These advancements can significantly transform the communication landscape of FL-enabled IDSs in IoV.
- Encryption standards: Encryption standards play a significant and multifaceted role in the context of FL-enabled IDSs within IoV. Ensuring the security and privacy of sensitive vehicular data during the transmission process is of paramount importance. The main challenge lies in adopting encryption standards that combine robustness and efficiency to effectively manage the substantial volumes of data transmitted

between vehicles and central servers. Moreover, within the FL framework, which entails collaborative model training on various devices, selecting encryption methods that can protect data while preserving the integrity of the collaborative learning process is a complex task [67]. Future advancements in this field primarily focus on developing encryption techniques that successfully reconcile the requirements of security, efficiency, and the necessity for collaborative learning. Research efforts aim to establish standardized encryption protocols tailored specifically for IoV settings. These protocols are intended to ensure data security and integrity while facilitating seamless model updates and promoting collaborative learning within a broad spectrum of vehicular networks.

- Edge computing: Incorporating edge computing into FL-enabled IDSs within IoV introduces both challenges and potential solutions. While local data processing on devices has the potential to alleviate network bandwidth demands, it also brings about issues related to resource limitations and data diversity. IoV devices, often constrained in terms of available resources, face difficulties when attempting to execute computationally intensive FL algorithms on the device itself. Furthermore, ensuring consistency and accuracy in updating models across various vehicles with different hardware configurations and data formats presents a significant challenge. Future research in this domain seeks to enhance the effectiveness of edge computing methodologies, facilitating efficient local data processing and collaborative learning while mitigating the variations in device capabilities. Leveraging edge computing, IDSs empowered by FL in IoV can realize benefits such as reduced communication overhead and improved real-time intrusion detection capabilities [68]. This, in turn, contributes to the establishment of more secure and responsive vehicular networks
- Optimization of Federated Learning and intrusion detection system parameters: FL predominantly relies on deep learning models that involve a diverse set of trainable parameters, which the user can configure. Additionally, IDSs are highly sensitive to these parameters. The next research avenue in FL-enabled IDSs for IoV involves optimizing FL and IDS parameters, as this directly impacts performance and training effectiveness [45]. Given the dynamic and diverse nature of IoV environments, it becomes imperative to identify the most suitable parameters for FL algorithms. This includes determining appropriate learning rates, aggregation methods, and local model parameters. In addition, customizing these parameters for specific intrusion detection tasks and diverse vehicular datasets can significantly improve the performance and accuracy of FL-enabled IDSs [51]. Future research should explore these factors in greater depth, utilizing methodologies such as hyperparameter tuning and adaptive learning algorithms [51]. By optimizing these parameters, researchers can finely tailor FL-enabled IDSs to suit specific IoV scenarios. This optimization process ensures effective collaboration, precise intrusion detection, and minimized communication overhead, ultimately paving the way for the development of more robust and responsive vehicular security systems.
- Heterogeneity and interpretability of the Federated Learning model: In the realm of FL-enabled IDSs for IoV, the heterogeneity and interpretability of FL models are of paramount importance. Heterogeneity stems from the distinct characteristics of vehicular data and the varying capabilities of different vehicles and their sensors. Coordinating multiple models for effective collaboration, especially in real-time intrusion detection, introduces a high degree of complexity. Moreover, prioritizing the interpretability of these models is crucial, as it enables a comprehensive understanding of the rationale behind intrusion alerts. This understanding is valuable for both developers and end-users. Future research endeavors are geared towards developing approaches that harmonize these diverse models, ensuring their seamless integration to enhance intrusion detection accuracy Simultaneously, researchers are dedicated to enhancing the interpretability of FL models through methodologies like explainable AI, which provides insights into the decision-making processes of these models. By

effectively addressing these challenges, FL-enabled IDSs in IoV can achieve a state of equilibrium that encompasses various data sources, model interpretability, and efficient intrusion detection. This, in turn, fosters confidence and comprehension among stakeholders in vehicular security.

- Big data management: Effective management of big data poses a significant challenge within the context of FL-enabled IDSs in IoV. The sheer volume, velocity, and diversity of data generated by vehicles require robust storage, processing, and analysis capabilities [69]. The integration of FL-enabled IDSs necessitates the use of extensive data for training and model updates. Efficiently handling this vast amount of data is paramount. The complexity lies in maintaining timely data collection, aggregation, and storage while preserving real-time intrusion detection capabilities, particularly when considering the limited resources of vehicle networks. Future studies will concentrate on creating distributed and scalable storage systems, better data processing algorithms, and advanced data analytics methods. By addressing big data management challenges, FL-enabled IDSs in IoV can harness the wealth of vehicular data efficiently, enhancing the precision and agility of IDSs in dynamic and networked vehicular environments.
- Sparse data: Vehicle data, especially regarding specific types of security threats, can be sparse and unevenly distributed across vehicles. Data sparsity may lead to biased models, as they might not adequately capture certain types of intrusions. Consequently, this limitation can hinder the overall effectiveness of the IDS. Addressing the issue of sparse data requires innovative methodologies, such as data augmentation, imputation approaches, or customized algorithms designed to handle incomplete datasets effectively [70]. Future research efforts aim to develop algorithms that can successfully enable FL models to learn from limited and irregular data. By effectively tackling the issue of sparse data, FL-enabled IDSs in IoV can enhance their precision, ensuring a more comprehensive and nuanced understanding of various intrusion patterns across different vehicular scenarios.
- Stability: Stability is a significant challenge within the context of FL-enabled IDSs in IoV. The inherent instability of the FL process is introduced by the dynamic nature of vehicular networks, characterized by the continuous changes in the composition and positions of vehicles. This variability can potentially disrupt the FL environment, affecting the consistency and accuracy of the IDS models. Maintaining stability requires the implementation of robust systems to address fluctuations in participation rates, network disconnections, and intermittent data availability [69]. Future research aims to develop algorithms that can adapt dynamically to changes in the network, ensuring the stability of FL models, even when confronted with evolving IoV scenarios. By addressing this challenge, FL-enabled IDSs in IoV can consistently perform at a high level, providing reliable capabilities for detecting unauthorized access despite the everchanging characteristics of vehicular networks.
- Reliability: Applications related to intelligent transportation and unmanned aerial vehicle detection demand a high level of reliability due to their safety-critical nature. Failures in meeting reliability standards can lead to severe consequences, including significant loss of life and property. Achieving reliability in the context of intrusion detection within a diverse and dynamic vehicle network presents significant challenges. Maintaining constant and accurate IDS performance is complicated by factors such as network latency, fluctuations in data quality, and the reliability of data transfer from individual vehicles. To ensure reliability, robust FL algorithms are needed to manage data discrepancies, adapt to changing network conditions, and effectively integrate data from diverse vehicles. Moreover, the timely and accurate deployment of intrusion detection solutions depends on the reliability of model updates and communication protocols. Future research aims to enhance the reliability of IDSs in IoV by refining FL algorithms, improving data preprocessing methods, and optimizing communica-

tion protocols. This will ultimately ensure the consistent and reliable operation of FL-enabled IDSs across diverse IoV environments.

- Real-time data: In the context of vehicle environments, responding promptly to security threats is crucial for ensuring passenger safety and network security. Swift and effective intrusion detection relies on processing the substantial volume of real-time data provided by vehicles. The primary challenge lies in developing FL algorithms capable of handling this increased data volume efficiently, with a focus on enabling timely anomaly or intrusion identification. Moreover, optimizing communication protocols to efficiently transmit relevant real-time data to central servers for model updates is of paramount importance. Future research in this area is directed towards creating FL models that combine lightweight characteristics with high-performance capabilities. This involves exploring the use of edge computing for local real-time analysis and improving communication protocols to facilitate seamless and swift sharing of real-time data [68]. By effectively addressing this challenge, the utilization of FL-enabled IDSs in IoV can offer immediate responses to security threats, thereby enhancing the overall safety and security of vehicular networks.

5. Conclusions

When we consider the extensive landscape of IDSs supported by FL in the context of IoV, it becomes abundantly clear that we are on the threshold of a revolutionary era in the field of vehicular network security. This realization is supported by the fact that IoV is the foundation upon which IDSs are constructed. IoV requires a security paradigm that is both resilient and adaptable due to its vast network of interconnected devices and vehicles. With its decentralized approach, Federated Learning has emerged as a beacon, offering a harmonious balance between data privacy and collaborative intelligence. It addresses the growing concerns about data privacy in our hyper-connected world by enabling vehicles to train models locally, ensuring that sensitive data are always retained on the device, thus solving this problem. The aggregation of these local models at a central location produces IDSs that are more accurate and capable of adapting to changing threat land-scapes, while simultaneously tapping into the collective wisdom of the entire network. However, challenges persist, as is expected with any emerging technology. Further research should take into account issues such as scalability, real-time processing demands, and maintaining model correctness across a wide range of vehicle nodes. In this paper, we conducted a well-organized literature review on IDSs based on FL within an IoV environment. We identified the relevant state of the art in FL-based IDSs within the IoV domain, covering the years from FL's inception in 2016 through 2023. Additionally, we introduced a general taxonomy to describe the FL systems, ensuring a coherent structure to guide future research. Finally, drawing upon the literature analysis, we elucidated some of the main challenges and potential directions for future studies in developing IDSs empowered by FL within the IoV context. In conclusion, as IoV continues to rapidly evolve, the interdependence between FL and IDSs will play a crucial role in establishing a vehicular ecosystem that is both secure and resilient, all while also safeguarding privacy.

Author Contributions: J.A.: Conceptualization, Methodology, Analysis, Writing—Original Draft Preparation, Visualization,Validation; K.A.: Supervision, Writing—Review & Editing, Investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by Institutional Fund Projects under grant no. (IFPDP-269-22). Therefore, the authors gratefully acknowledge technical and financial support from Ministry of Education and Deanship of Scientific Research (DSR), King Abdulaziz University (KAU), Jeddah, Saudi Arabia.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoV	Internet of Vehicles
IDS	Intrusion Detection System
FL	Federated Learning
IoT	Internet of Things
SMPC	Secure MultiParty Computation
SGD	Stochastic Gradient Descent
DP	Differential Privacy
LDP	Local Differential Privacy
CDP	Central Differential Privacy
non-IID	non-Independent and Identically Distributed
HIDS	Host Intrusion Detection System
NIDS	Network Intrusion Detection System
ML	Machine Learning
DL	Deep Learning
VANETs	Vehicular Ad-hoc Networks
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
DoS	Denial-of-Service
LSTM	Long Short-Term Memory
DCNN	Deep Convolutional Neural Network
SVM	Support Vector Machine
RF	Random Forest
FedAvg	Federated Averaging Algorithm
BO	Bayesian Optimization
MLP	Multilayer Perceptron
R2L	Remote to Local
U2R	User to Root
FPR	The False Positive Rate
MMD	Maximum Mean Discrepancy
ED	Energy Distance
NGA	Number Of Global Aggregations
R	Numbers Of Communication Rounds
FL-MAAE	Federated Learning Memory-Augmented Autoencoder
SMC	Secure-Multiparty Computation
UAV	Unmanned Aerial Vehicle

References

1. Alladi, T.; Kohli, V.; Chamola, V.; Yu, F.R. Securing the internet of vehicles: A deep learning-based classification framework. *IEEE Netw. Lett.* **2021**, *3*, 94–97. [CrossRef]
2. Ji, B.; Zhang, X.; Mumtaz, S.; Han, C.; Li, C.; Wen, H.; Wang, D. Survey on the internet of vehicles: Network architectures and applications. *IEEE Commun. Stand. Mag.* **2020**, *4*, 34–41. [CrossRef]
3. Garg, T.; Kagalwalla, N.; Churi, P.; Pawar, A.; Deshmukh, S. A survey on security and privacy issues in IoV. *Int. J. Electr. Comput. Eng.* **2020**, *5*, 2088–8708. [CrossRef]
4. Zavvos, E.; Gerding, E.H.; Yazdanpanah, V.; Maple, C.; Stein, S. Privacy and Trust in the Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 10126–10141. [CrossRef]
5. Bevish Jinila, Y.; Merlin Sheeba, G.; Prayla Shyry, S. PPSA: Privacy preserved and secured architecture for internet of vehicles. *Wirel. Pers. Commun.* **2021**, *118*, 3271–3288. [CrossRef]
6. Peng, R.; Li, W.; Yang, T.; Huafeng, K. An internet of vehicles intrusion detection system based on a convolutional neural network. In Proceedings of the 2019 IEEE Intl Conferences on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Xiamen, China, 16–18 December 2019; pp. 1595–1599.
7. Gasmi, R.; Aliouat, M. Vehicular ad hoc networks versus internet of vehicles-a comparative view. In Proceedings of the 2019 International Conference on Networking and Advanced Systems (ICNAS), Annaba, Algeria, 26–27 June 2019; pp. 1–6.
8. Indu, S.K. Internet of Vehicles (IoV): Evolution, Architecture, Security Issues and Trust Aspects. *Int. J. Recent Technol. Eng. (IJRTE)* **2019**, *7*, 260–280.

9. Fu, W.; Xin, X.; Guo, P.; Zhou, Z. A practical intrusion detection system for Internet of vehicles. *China Commun.* **2016**, *13*, 263–275. [CrossRef]
10. Sherazi, H.H.R.; Iqbal, R.; Ahmad, F.; Khan, Z.A.; Chaudary, M.H. DDoS attack detection: A key enabler for sustainable communication in internet of vehicles. *Sustain. Comput. Inform. Syst.* **2019**, *23*, 13–20. [CrossRef]
11. Bagga, P.; Das, A.K.; Wazid, M.; Rodrigues, J.J.; Park, Y. Authentication protocols in internet of vehicles: Taxonomy, analysis, and challenges. *IEEE Access* **2020**, *8*, 54314–54344. [CrossRef]
12. Osibo, B.K.; Zhang, C.; Xia, C.; Zhao, G.; Jin, Z. Security and privacy in 5G internet of vehicles (IoV) environment. *J. Internet Things* **2021**, *3*, 77. [CrossRef]
13. Abbasi, S.; Rahmani, A.M.; Balador, A.; Sahafi, A. Internet of Vehicles: Architecture, services, and applications. *Int. J. Commun. Syst.* **2020**, *34*, e4793. [CrossRef]
14. El Madani, S.; Motahhir, S.; El Ghzizal, A. Internet of vehicles: Concept, process, security aspects and solutions. *Multimed. Tools Appl.* **2022**, *81*, 16563–16587. [CrossRef]
15. Seth, I.; Guleria, K.; Panda, S.N.; Anand, D.; Alsubhi, K.; Aljahdali, H.M.; Singh, A. A taxonomy and analysis on Internet of Vehicles: Architectures, protocols, and challenges. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9232784. [CrossRef]
16. Driss, M.; Almomani, I.; e Huma, Z.; Ahmad, J. A federated learning framework for cyberattack detection in vehicular sensor networks. *Complex Intell. Syst.* **2022**, *8*, 4221–4235. [CrossRef]
17. Sharma, N.; Chauhan, N.; Chand, N. Security challenges in Internet of Vehicles (IoV) environment. In Proceedings of the 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018; pp. 203–207.
18. Hu, Q.; Fan, X.; Shan, A.; Wang, Z. Sybil attack detection method based on timestamp-chain in Internet of vehicles. In Proceedings of the 2021 IEEE International Conference on Smart Internet of Things (SmartIoT), Jeju, Republic of Korea, 13–15 August 2021; pp. 174–178.
19. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3347–3366. [CrossRef]
20. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
21. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
22. Qiang, Y.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19.
23. Ruzafa-Alcázar, P.; Fernández-Saura, P.; Mármol-Campos, E.; González-Vidal, A.; Hernández-Ramos, J.L.; Bernal-Bernabe, J.; Skarmeta, A.F. Intrusion detection based on privacy-preserving federated learning for the industrial IoT. *IEEE Trans. Ind. Inform.* **2021**, *19*, 1145–1154. [CrossRef]
24. Shejwalkar, V.V. Quantifying and Enhancing the Security of Federated Learning. 2023. Available online: https://www.cics.umass.edu/event/20230426/quantifying-and-strengthening-security-federated-learning (accessed on 6 September 2023).
25. Fedorchenko, E.; Novikova, E.; Shulepov, A. Comparative review of the intrusion detection systems based on federated learning: Advantages and open challenges. *Algorithms* **2022**, *15*, 247. [CrossRef]
26. Alazab, M.; RM, S.P.; Parimala, M.; Maddikunta, P.K.R.; Gadekallu, T.R.; Pham, Q.V. Federated learning for cybersecurity: Concepts, challenges, and future directions. *IEEE Trans. Ind. Inform.* **2021**, *18*, 3501–3509. [CrossRef]
27. Belenguer, A.; Navaridas, J.; Pascual, J.A. A review of federated learning in intrusion detection systems for iot. *arXiv* **2022**, arXiv:2204.12443.
28. Sattler, F.; Wiedemann, S.; Müller, K.L.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef] [PubMed]
29. Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access* **2020**, *8*, 140699–140725. [CrossRef] [PubMed]
30. Sittijuk, P.; Tamee, K. Performance measurement of federated learning on imbalanced data. In Proceedings of the 2021 18th International Joint Conference on Computer Science and Software Engineering (JCSSE), Virtual, 30 June–3 July 2021; pp. 1–6.
31. Moshawrab, M.; Adda, M.; Bouzouane, A.; Ibrahim, H.; Raad, A. Reviewing Federated Learning Aggregation Algorithms; Strategies, Contributions, Limitations and Future Perspectives. *Electronics* **2023**, *12*, 2287. [CrossRef]
32. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
33. Prakash, S.; Avestimehr, A.S. Mitigating byzantine attacks in federated learning. *arXiv* **2020**, arXiv:2010.07541.
34. Rodríguez-Barroso, N.; Jiménez-López, D.; Luzón, M.V.; Herrera, F.; Martínez-Cámara, E. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Inf. Fusion* **2023**, *90*, 148–173. [CrossRef]
35. El Ouadrhiri, A.; Abdelhadi, A. Differential privacy for deep and federated learning: A survey. *IEEE Access* **2022**, *10*, 22359–22380. [CrossRef]
36. Liu, J.; Huang, J.; Zhou, Y.; Li, X.; Ji, S.; Xiong, H.; Dou, D. From distributed machine learning to federated learning: A survey. *Knowl. Inf. Syst.* **2022**, *64*, 885–917. [CrossRef]

37. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
38. Mammen, P.M. Federated learning: Opportunities and challenges. *arXiv* **2021**, arXiv:2101.05428.
39. Huang, C.; Huang, J.; Liu, X. Cross-silo federated learning: Challenges and opportunities. *arXiv* **2022**, arXiv:2206.12949.
40. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1205–1221. [CrossRef]
41. Kholod, I.; Yanaki, E.; Fomichev, D.; Shalugin, E.; Novikova, E.; Filippov, E.; Nordlund, M. Open-source federated learning frameworks for IoT: A comparative review and analysis. *Sensors* **2020**, *21*, 167. [CrossRef] [PubMed]
42. Jamali-Rad, H.; Abdizadeh, M.; Singh, A. Federated learning with taskonomy for non-IID data. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 8719–8730. [CrossRef] [PubMed]
43. Tan, A.Z.; Yu, H.; Cui, L.; Yang, Q. Towards personalized federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 9587–9603. [CrossRef] [PubMed]
44. Huong, T.T.; Bac, T.P.; Ha, K.N.; Hoang, N.V.; Hoang, N.X.; Hung, N.T.; Tran, K.P. Federated learning-based explainable anomaly detection for industrial control systems. *IEEE Access* **2022**, *10*, 53854–53872. [CrossRef]
45. Agrawal, S.; Sarkar, S.; Aouedi, O.; Yenduri, G.; Piamrat, K.; Alazab, M.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Comput. Commun.* **2022**, *195*, 346–361. [CrossRef]
46. Rashid, M.M.; Khan, S.U.; Eusufzai, F.; Redwan, M.A.; Sabuj, S.R.; Elsharief, M. A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks. *Network* **2023**, *3*, 158–179. [CrossRef]
47. Xing, L.; Wang, K.; Wu, H.; Ma, H.; Zhang, X. FL-MAAE: An Intrusion Detection Method for the Internet of Vehicles Based on Federated Learning and Memory-Augmented Autoencoder. *Electronics* **2023**, *12*, 2284. [CrossRef]
48. Amanullah, M.A.; Loke, S.W.; Chhetri, M.B.; Doss, R. A Taxonomy and Analysis of Misbehaviour Detection in Cooperative Intelligent Transport Systems: A Systematic Review. *ACM Comput. Surv.* **2023**, *56*, 1–38. [CrossRef]
49. Rani, P.; Sharma, C.; Ramesh, J.V.N.; Verma, S.; Sharma, R.; Alkhayyat, A.; Kumar, S. Federated Learning-Based Misbehaviour Detection for the 5G-Enabled Internet of Vehicles. *IEEE Trans. Consum. Electron.* **2023**. [CrossRef]
50. Hosseinzadeh, M.; Hemmati, A.; Rahmani, A.M. Federated learning-based IoT: A systematic literature review. *Int. J. Commun. Syst.* **2022**, *35*, e5185. [CrossRef]
51. Lavaur, L.; Pahl, M.-O.; Busnel, Y.; Autrel, F. The evolution of federated learning-based intrusion detection and mitigation: A survey. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 2309–2332. [CrossRef]
52. Yang, J.; Hu, J.; Yu, T. Federated AI-enabled in-vehicle network intrusion detection for internet of vehicles. *Electronics* **2022**, *11*, 3658. [CrossRef]
53. Uprety, A.; Rawat, D.B.; Li, J. Privacy preserving misbehavior detection in IoV using federated machine learning. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6.
54. Hbaieb, A.; Ayed, S.; Chaari, L. Federated learning based IDS approach for the IoV. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–6.
55. Boualouache, A.; Brik, B.; Senouci, S.-M.; Engel, T. On-Demand Security Framework for 5GB Vehicular Networks. *IEEE Internet Things Mag.* **2023**, *6*, 26–31. [CrossRef]
56. Vinita, L.J.; Vetriselvi, V. Federated Learning-based Misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles. *Hoc Netw.* **2023**, *144*, 103153. [CrossRef]
57. Aliyu, I.; Feliciano, M.C.; Van Engelenburg, S.; Kim, D.O.; Lim, C.G. A Blockchain-Based Federated Forest for SDN-Enabled In-Vehicle Network Intrusion Detection System. *IEEE Access* **2021**, *9*, 102593–102608. [CrossRef]
58. Zainudin, A.; Akter, R.; Kim, D.-S.; Lee, J.-M. FedIoV: A Federated Learning-Assisted Intrusion Messages Detection in Internet of Vehicles. 2022; pp. 305–306. Available online: https://www.dbpia.co.kr/Journal/articleDetail?nodeId=NODE11197063 (accessed on 6 October 2023).
59. Yu, T.; Hua, G.; Wang, H.; Yang, J.; Hu, J. Federated-lstm based network intrusion detection method for intelligent connected vehicles. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 4324–4329.
60. Taslimasa, H.; Dadkhah, S.; Neto, E.C.P.; Xiong, P.; Iqbal, S.; Ray, S.; Ghorbani, A.A. ImageFed: Practical Privacy Preserving Intrusion Detection System for In-Vehicle CAN Bus Protocol. In Proceedings of the 2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), New York, NY, USA, 6–8 May 2023; pp. 122–129.
61. Xu, Q.; Zhang, L.; Ou, D.; Yu, W. Secure Intrusion Detection by Differentially Private Federated Learning for Inter-Vehicle Networks. *Transp. Res. Rec.* **2023**, *2677*, 421–437. [CrossRef]
62. Korba, A.A.; Boualouache, A.; Brik, B.; Rahal, R.; Ghamri-Doudane, Y.; Senouci, S.M. Federated Learning for Zero-Day Attack Detection in 5G and Beyond V2X Networks. In AlgoTel 2023-25èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications. 2023. Available online: https://hal.science/hal-04087452/ (accessed on 22 September 2023).
63. Aliyu, I.; Engelenburg, S.V.; Mu'Azu, M.B.; Kim, J.; Lim, C.G. Statistical Detection of Adversarial Examples in Blockchain-Based Federated Forest In-Vehicle Network Intrusion Detection Systems. *IEEE Access* **2022**, *10*, 109366–109384. [CrossRef]

64. Sebastian, A. Enhancing Intrusion Detection in Internet of Vehicles Through Federated Learning. *arXiv* **2023**, arXiv:2311.13800.
65. Campos, E.M.; Saura, P.F.; González-Vidal, A.; Hernández-Ramos, J.L.; Bernabé, J.B.; Baldini, G.; Skarmeta, A. Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges. *Comput. Netw.* **2022**, *203*, 108661. [CrossRef]
66. Billah, M.; Mehedi, S.T.; Anwar, A.; Rahman, Z.; Islam, R. A systematic literature review on blockchain enabled federated learning framework for internet of vehicles. *arXiv* **2022**, arXiv:2203.05192.
67. Duy, P.T.; Hao, H.N.; Chu, H.M.; Pham, V.H. A Secure and Privacy Preserving Federated Learning Approach for IoT Intrusion Detection System. In Proceedings of the Network and System Security: 15th International Conference, NSS 2021, Tianjin, China, 23 October 2021; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 353–368.
68. Duan, Q.; Huang, J.; Hu, S.; Deng, R.; Lu, Z.; Yu, S. Combining Federated Learning and Edge Computing Toward Ubiquitous Intelligence in 6G Network: Challenges, Recent Advances, and Future Directions. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2892–2950. [CrossRef]
69. Danba, S.; Bao, J.; Han, G.; Guleng, S.; Wu, C. Toward collaborative intelligence in IoV systems: Recent advances and open issues. *Sensors* **2022**, *22*, 6995. [CrossRef]
70. Thonglek, K.; Takahashi, K.; Ichikawa, K.; Nakasan, C.; Leelaprute, P.; Iida, H. Sparse communication for federated learning. In Proceedings of the 2022 IEEE 6th International Conference on Fog and Edge Computing (ICFEC), Messina, Italy, 16–19 May 2022; pp. 1–8.