Ministère de l'enseignement Supérieur et de la recherche Scientifique
وزارة التعليم العالي والبحث العلمي

Université Badji Mokhtar – Annaba
Faculté de Technologie
Département Informatique

جامعة باجي مختار – عنابة
كلية التكنولوجيا
قسم الاعلام الالي

Thèse

Présentée pour obtenir le diplôme de

Doctorat Troisième Cycle

Filière : Informatique

Spécialité : Réseaux et Sécurité

Par :
FRIHA Othmane

Thème :

La Sécurité et la vie privée dans l'agriculture industrielle intelligente et durable

Thèse soutenue le **27/03/2024** devant le jury composé de :

N°	Nom et prénom	Grade	Etablissement	Qualité
01	Ghoualmi-Zine Nacera	Prof.	Université Badji Mokhtar -Annaba	Président
02	Ahmim Marwa	MCA	Université Badji Mokhtar -Annaba	Rapporteur
03	Nafaa Mehdi	MCB	Université Badji Mokhtar -Annaba	Co-rapporteur
04	Djellali Hayet	MCA	Université Badji Mokhtar -Annaba	Examinateur
05	Saighi Asma	MCA	Université de Oum El Bouaghi	Examinateur
06	Kouahla Zine-Eddine	Prof.	Université de Guelma	Examinateur

الملخص

يشير المصطلح "الزراعة الذكية" إلى استخدام تقنيات مثل إنترنت الأشياء ، وأجهزة التتبع ، والروبوتات ، والذكاء الاصطناعي في العمليات الزراعية. الهدف النهائي هو تعزيز كمية ونوعية الغلة مع تحسين القوى العاملة المرتبطة المشاركة. أصبحت الأنظمة الذكية والمستدامة المصممة للزراعة جزءًا أساسيًا من الممارسات الزراعية الحديثة. مع ظهور التكنولوجيا المتقدمة ، يمكن للمزارعين الآن مراقبة محاصيلهم وماشيتهم في الوقت الفعلي ، وتحليل البيانات لتحسين إنتاجهم ، وتقليل التأثير البيئي لعملياتهم.

ومع ذلك ، فإن دمج التكنولوجيا في الزراعة يجلب معه أيضًا تحديات جديدة ، لا سيما في مجال الأمن والخصوصية. يعد الأمان والخصوصية من المكونات الأساسية لأي نظام زراعي ، حيث إنها تحمي بيانات المزارعين ، وتمنع الهجمات الإلكترونية ، وتضمن عمل الأنظمة بكفاءة. يمكن أن يكون لخرق واحد لأمن نظام زراعي ذكي عواقب وخيمة ، من تعطيل عملية الزراعة إلى سرقة المعلومات الحساسة ، وحتى التسبب في خسائر اقتصادية.

الغرض الرئيسي من هذه الأطروحة هو التحقيق في قضايا الأمن والخصوصية التي تحيط حاليًا بمشهد الزراعة الذكية على مستويات مختلفة, وتطوير آليات أمنية قوية للزراعة الصناعية الذكية. في هذا السياق ، نقترح ثلاث خطط أمنية فعالة لتأمين أنظمة الزراعة الذكية ضد الهجمات من التهديدات المادية وتهديدات الفضاء الإلكتروني.

تتكون المساهمة الأولى من بنية لتوفير الأمن السيبراني للأنظمة الزراعية الذكية من خلال دمج تقنية سلسلة الكتل "البلوك تشين" وحوسبة الضباب والشبكات المعرفة بالبرمجيات لإنترنت الأشياء الزراعي. في بنية الأمان المقترحة ، يتم تضمين ثلاثة أجزاء رئيسية (أ) نظام إدارة البيانات لإنترنت الأشياء الزراعي لتقديم جمع بيانات إنترنت الأشياء في الوقت الفعلي وتحليلها وتصورها وإدارة الجهاز ، (ب) مخطط مراقبة النزاهة على أساس سلسلة الكتل "البلوك تشين" لمنع التسليم الخاطئ لعناصر التحكم والمعلومات ، و (ج) مفتاح محدد بالبرمجيات لدعم تقنيات الشبكات المعرفة بالبرمجيات لتحسين إدارة الشبكة. أدى الحل الأمني المقترح إلى أداء عام جيد على منصة إنترنت الأشياء مفتوحة المصدر من خلال الجمع بين سلسلة الكتل "البلوك تشين" وتقنيات الشبكات المعرفة بالبرمجيات.

تتكون المساهمة الثانية من نظام متحد لاكتشاف اقتحام التعلم العميق الذي يحمي خصوصية البيانات من خلال التعلم المحلي ، حيث تكتسب الأجهزة المعرفة من أقرانها من خلال مشاركة تحديثات النموذج الخاصة بهم فقط مع خادم تجميع ينتج نموذج كشف محسن. توضح النتائج التجريبية أن النظام يتفوق في الأداء على الإصدارات التقليدية / المركزية للتعلم العميق في حماية خصوصية بيانات أجهزة إنترنت الأشياء الزراعية الذكية بدقة عالية.

تتكون المساهمة الثالثة من نظام أمان لتطبيقات إنترنت الأشياء الصناعية الذكية ضد مجموعة واسعة من المخاطر السيبرانية. بصرف النظر عن ذلك ، فإن النهج المقترح يعزز ويؤمن سير عمل تدريب التعلم الموحد ، من خلال حماية البيانات المتبادلة ضد الأطراف الخبيثة من خارج مجموعات التدريب وكذلك تأمين العملية نفسها ضد الكيانات المشاركة. علاوة على ذلك ، فإن النظام المقترح لا مركزي بالكامل ، مما يزيل مخاطر تعريض خادم التجميع للخطر وتعطيل عملية التدريب بأكملها في نهج التعلم الموحد المعتاد. أظهر التقييم التجريبي للنظام المقترح أدائه التشغيلي القوي في استكشاف أنواع مختلفة من التهديدات السيبرانية لأنظمة إنترنت الأشياء الصناعية ، فضلاً عن فوائده على

الأساليب الحديثة الحالية.

الكلمات مفتاحية : الأمن ، الخصوصية ، الزراعة الذكية ، الزراعة الصناعية المستدامة ، إنترنت الأشياء ، سلسلة الكتل ، أنظمة كشف التسلل ، التعلم الموحد ، تشفير ما بعد الكم.

Abstract

The term "*smart agriculture*" refers to the use of technologies such as the IoT, tracking devices, robotics, and AI in agricultural operations. The ultimate aim is to improve the yield both in quantity and quality. while optimizing the associated workforce involved. Smart and sustainable systems designed for agriculture have become an essential part of modern farming practices. With the emergence of advanced technology, farmers can now monitor their crops and livestock in real time, analyze data to improve their production, and reduce the environmental impact of their operations.

However, the integration of technology in agriculture also brings along new challenges, particularly in the area of privacy and security. Which are critical components of any agricultural system, as they protect the farmers' data, prevent cyberattacks, and ensure that the systems are working efficiently. A single breach in the security of a smart agricultural system can have devastating consequences, from disrupting the farming process to stealing sensitive information, and even causing economic losses.

The main purpose of this thesis is to investigate the privacy and security issues which currently encircle the smart agriculture landscape at different levels, and to develop robust security mechanisms for the smart industrial agriculture. In this context, we propose three effective security schemes for securing smart agriculture systems against attacks from both physical and cyberspace threats.

The initial contribution is an architecture to address the cybersecurity of smart agricultural systems by incorporating blockchain technology, the fog computing and software-defined networking for agricultural IoT. Within the suggested architecture for security, three key parts are included (a) an agricultural IoT data management system to deliver real-time IoT data gathering, analytical, instrument visualization and management, (b) an integrity monitoring scheme based on blockchain to inhibit the erroneous delivery of checks and feedback, and (c) a software-defined switch to handle software-defined networking in order to improve the management of the network. The suggested security solution delivered good performance overall over an open-source IoT platform.

In the second contribution, we developed a federated deep learning intrusion detection system safeguarding the privacy of data by local learning, under which devices acquire knowledge by sharing only their model update with an aggregation server that generates an upgraded detection model. Based on experimental results, the scheme outperforms both conventional/centralized versions on deep learning in safeguarding the data privacy of intelligent agricultural IoT devices with high accuracy.

Another contribution is a comprehensive security system for industrial implementations of intelligent IoT solutions against a variety of cyber-risks. The proposed framework

both enhances and secures the federated learning training workstream, providing protection for the data exchanged against malicious external parties to the training groups, and also protecting the process itself from participating entities. Furthermore, the proposed system is entirely decentralized, which eliminates the danger of jeopardizing the aggregation server and upsetting the entire training process in the usual federated learning training. Based on experimental evaluation, it has been demonstrated that the system offers robust operational capabilities for recognizing different types of cyber threats to industrial IoT systems, along with advantages over existing state-of-the-art approaches.

Keywords : Security, Privacy, Smart Agriculture, Sustainable industrial agriculture, IoT, Blockchain, Intrusion Detection Systems, Federated Learning, Post-quantum cryptography.

Résumé

Le terme "*agriculture intelligente*" fait référence à l'utilisation de technologies telles que l'IoT, les dispositifs de suivi, la robotique et l'IA dans les opérations agricoles. L'objectif final est d'améliorer la qualité des rendements tout en optimisant la main-d'œuvre associée. Les systèmes intelligents et durables conçus pour l'agriculture sont devenus un élément essentiel des pratiques agricoles modernes. Avec l'émergence de technologies de pointe, les agriculteurs peuvent désormais surveiller leurs cultures et leur bétail en temps réel, analyser les données pour améliorer leur production et réduire l'impact environnemental de leurs opérations.

Cependant, l'intégration de la technologie dans l'agriculture apporte également de nouveaux défis. La sécurité et la confidentialité sont des éléments essentiels de tout système agricole, car elles protègent les données des agriculteurs, empêchent les cyberattaques et garantissent que les systèmes fonctionnent efficacement. Une seule faille dans la sécurité d'un système agricole intelligent peut avoir des conséquences dévastatrices, allant de la perturbation du processus agricole au vol d'informations sensibles, et même à des pertes économiques.

L'objectif de cette thèse est d'étudier les problèmes de sécurité et de confidentialité qui entourent actuellement le paysage de l'agriculture intelligente à différents niveaux, et de développer des mécanismes de sécurité robustes pour l'agriculture industrielle intelligente. Dans ce contexte, nous proposons trois schémas de sécurité efficaces pour sécuriser les systèmes agricoles intelligents contre les risk des menaces physiques et du cyberespace.

La première contribution consiste en une architecture pour fournir la cybersécurité aux systèmes agricoles intelligents en incorporant la technologie blockchain, le calcul du brouillard et la mise en réseau définie par logiciel pour l'IdO agricole. Dans l'architecture de sécurité proposée, trois parties principales sont incluses (a) un système de gestion des données pour l'IoT agricole afin de fournir une collecte, une analyse, une visualisation et une gestion des appareils en temps réel, (b) un schéma de surveillance de l'intégrité basé sur la blockchain pour empêcher livraison erronée de commandes et d'informations, et (c) un commutateur défini par logiciel pour prendre en charge les technologies de réseau défini par logiciel afin d'améliorer la gestion du réseau. La solution de sécurité suggérée a permis d'obtenir de bonnes performances globales sur une plate-forme IoT open source en combinant les technologies de blockchain et de réseau défini par logiciel.

La deuxième contribution consiste en un système fédéré de détection d'intrusion d'apprentissage en profondeur qui protège la confidentialité des données grâce à l'apprentissage local, dans lequel les appareils acquièrent des connaissances de leurs pairs en partageant uniquement leurs mises à jour de modèle avec un serveur d'agrégation qui produit un

modèle de détection amélioré. Les résultats expérimentaux démontrent que le système surpasse les versions conventionnelles/centralisées de l'apprentissage en profondeur en protégeant la confidentialité des données des appareils IoT agricoles intelligents avec une grande précision.

ET la troisième contribution consiste en un système de sécurité pour les implémentations industrielles intelligentes de l'IdO contre un large éventail de cyberrisques. En dehors de cela, l'approche proposée améliore et sécurise le flux de travail de formation d'apprentissage fédéré, en protégeant les données échangées contre les parties malveillantes extérieures aux groupes de formation ainsi qu'en sécurisant le processus lui-même contre les entités participantes. De plus, le système proposé est entièrement décentralisé, ce qui élimine le risque de mettre en péril le serveur d'agrégation et de perturber l'ensemble du processus de formation dans l'approche habituelle d'apprentissage fédéré. L'évaluation expérimentale du système proposé a démontré sa forte performance opérationnelle dans la reconnaissance de différents types de cybermenaces pour les systèmes IoT industriels, ainsi que ses avantages par rapport aux approches de pointe existantes.

Mots clés : Sécurité, confidentialité, agriculture intelligente, agriculture industrielle durable, IoT, Blockchain, systèmes de détection d'intrusion, apprentissage fédéré, cryptographie post-quantique.

Dedication

“

To my beloved parents, who have always been my pillars of strength and support. Though my father is no longer with us, his memory lives on in the love and guidance that he provided for me. This work is dedicated to both of you, for the unconditional love and sacrifices that you have made for me throughout my life.

Thank you for everything.

”

- Othmane

Table of Contents

List of Figures

List of Tables

List of Algorithms

General introduction

Agriculture has been instrumental in the growth of societies throughout history, given its role as the primary source of food on the planet. However, in the near future, the survival of approximately 11 billion people will be challenged over the next century [1]. This means an increase in global demand for food and water. Nevertheless, FAO (Food and Agriculture Organization of the United Nations) states that sustainable agricultural systems [2] can help end hunger and ensure food security without even requiring a 50% increase in production [3]. Smart agriculture aims to achieve this goal by prioritizing research, innovation, and technology to implement sustainable agricultural practices. Smart agriculture could revolutionize how we grow food. By using technology and data analysis, food producers of all sizes from farmers to global industries can optimize their practices, increase efficiency, improve yields, and reduce the environmental impact.

Smart agriculture involves the use of technologies including IoT sensors, data analytics, and machine learning to optimize crop yields, monitor livestock health, and produce food more efficiently [4]. Examples of smart agriculture projects include precision agriculture, livestock monitoring, and aquaponics. The potential of these technologies is to significantly increase agricultural productivity and reduce waste, leading to positive economic impacts such as increased food security and reduced food production costs. However, their economic impact will depend on factors such as their adoption rates and the regulatory environment. Studies have estimated that the global markets for precision agriculture and livestock monitoring will be worth billions of dollars by 2025 and 2026, respectively. According to a recent report by MarketsandMarkets [5],The smart agriculture sector is set to increase in value significantly, rising from an estimated USD 16.2 billion to 25.4 billion, representing a compound annual growth rate (CAGR) of 9.4% over the period 2023 to 2028. The growth of this market can be attributed to two main factors: the growing global population, which is putting pressure on the food supply system, and the increasing use of modern agricultural technologies.

Smart farming is a rapidly advancing discipline featuring a range of technologies designed to optimize agricultural production and efficiency. For instance, IoT is used to collect data on soil moisture, temperature, and other environmental factors to optimize irrigation and fertilizer application.For instance, IoT sensors are used to track soil moisture conditions in real time and adjust irrigation systems accordingly so that plants receive the optimum amount of water. Additionally, AI or artificial intelligence is used in smart farming to effectively analyze large amounts of data and provide feedback that can help growers improve their decision-making. For example, AI can be used to analyze satellite imagery to detect changes in crop health and predict potential yield losses.One important piece of technology in smart farming is Big Data. This concerns large quantities of

data produced by different sources in agriculture, including sensors, drones and weather stations. This information provides insight into crop and soil conditions, as well as other factors that affect agricultural production. Big data can be used to develop predictive models that can help farmers make better decisions about planting, irrigation, and other agricultural practices. Furthermore, cloud computing is used in smart agriculture to store and process large amounts of data from various sources. Cloud platforms allow farmers to access data and analysis tools from anywhere, using any device with an internet connection. This can help farmers make faster and more informed decisions about crop management.

As the future of intelligent agriculture progresses, these and other technologies will become increasingly important in shaping the future of farming [4].

Problem statement

The application of a wide range of technologies in the field of intelligent agriculture has revolutionized the way farming is carried out, bringing improvements in efficiency, productivity and sustainability. However, this increased connectivity and automation also pose significant security and privacy risks and threats, which can have severe consequences for farmers, agribusinesses, and consumers. For instance, back in 2017, Maersk, a major global shipping company, suffered substantial losses of $250M-$300M due to the Not-Petya malware attack, which serves as a good example [6]. Similarly, in 2021, the largest meat supplier globally, JBS, has fallen victim to a ransomware attack in which it was obliged to suspend certain operations in several territories, affecting thousands of workers. Agriculture cooperatives are being warned about the potential threat of ransomware attacks by the FBI in a Private Industry Notification. The notification emphasizes that these attacks, occurring during the critical planting and harvesting periods, could result in the loss of important confidential information, disrupt their operations, cause financial losses, and possibly result in food shortages [7].

The significant expansion and adoption of different technologies in the smart agricultural domain demand the implementation of suitable security and privacy policies that safeguard such systems against potential threats to their security and privacy. Despite the wide range of security countermeasures that have been suggested by both academic researchers and industry professionals, the complexity of the technologies employed in the smart agricultural domain, along with the deployment characteristics of certain technologies like sensors, and the imperative of ensuring absolute privacy in certain areas, renders conventional defense mechanisms untenable for addressing the challenge at hand and may prove inadequate in certain circumstances [8].

Objectives and contributions

The utilization of technologies in smart agriculture presents security and privacy risks and threats such as data breaches, cyber-attacks, system failures, insider threats, and lack of standardization and regulations, which can be mitigated through the implementation of robust security systems, protocols, standardization, and education. The primary

aim of this thesis is to investigate privacy and security issues, that are associated with smart industrial agriculture and to propose novel security mechanisms that account for the emerging technological trends in the domain, the characteristics of the technologies employed, the need for sustainability, and the resource constraints of the devices involved. In addressing the limitations of previous methods, these proposed approaches seek to enhance the security of smart agriculture by addressing key challenges related to privacy, data protection, and threat mitigation.

To attain these objectives, our research began with a thorough examination of the domain from multiple perspectives. Our initial contribution involved an exhaustive review of both contemporary and historical technologies, with a specific emphasis on IoT-based agricultural systems, which form the underlying infrastructure of smart agriculture. Building on this foundational knowledge, our attention then turned to the issue of security and privacy within the domain, as detailed in our second contribution. In this regard, we conducted a comprehensive assessment of existing and potential security and privacy threats, challenges, and associated defense mechanisms. We opted for intrusion detection systems, given their capabilities of recognizing and stopping a wide variety of attacks that may not be recognized automatically by firewalls, anti-virus software, and other existing security controls. In our third contribution, we conducted a comprehensive analysis of the current state of the art in the utilization of Intrusion Detection Systems (IDSs) within smart agriculture. Our analysis revealed that the deployment of IDSs based on Federated Learning (FL) methods is better suited for the domain, owing to its unique privacy requirements. So we started reviewing existing solutions in our fourth contribution.

After gaining a comprehensive understanding of the research domain, we proceeded to propose several solutions in our fifth, sixth, and seventh contributions. Firstly, we suggested the utilization of Software-Defined Networking (SDN) to safeguard blockchain-based agricultural systems, including private supply chain infrastructures. Subsequently, we proposed a multilayered Federated Learning (FL)-based Intrusion Detection System (IDS) with a lightweight design and superior detection capabilities in our sixth contribution. Finally, we enhanced the security of the FL by implementing safeguards for gradients in our seventh contribution.

Thesis structure

Throughout this thesis, all the chapters are transcriptions of our published articles in various scientific journals. There are 6 chapters that are equally divided into 2 main parts: the background and the contributions. In the background part, we provide an overview of smart agriculture and examine related security threats, challenges, requirements, and solutions. The contributions part presents our proposed security enhancement techniques for smart agriculture. This thesis is structured as follows:

- *Chapter 1* provides an overview of the smart agricultural domain, covering a description of the field, fundamental concepts that shape it, and a taxonomy of various smart agricultural applications.
- *Chapter 2* presents a comprehensive analysis of smart agriculture from the perspec-

tive of security and privacy. It begins by providing an overview of the security aspects of the domain, followed by a taxonomy of the threat model. Subsequently, it explores various defense strategies that can be implemented to protect against these threats.

- *Chapter 3* This chapter focuses on a specific defense strategy for securing smart agriculture, namely the IDS. It begins with a concise introduction to the concept of IDS, outlining its definition, types, and fundamental design stages. Subsequently, it explores the specific opportunities that IDSs can offer for securing smart agriculture. Next, it discusses available and possible IDS-based security solutions for smart agriculture from a technological perspective. Finally, the chapter focuses on anomaly-based intrusion detection with a specific learning approach, namely the FL approach. It provides a concise introduction to the topic, followed by a review of selected recent works.

- *Chapter 4* offers an agricultural IoT security architecture that combines blockchain, fog computing and SDN. This architecture comprises three main parts: a farm IoT data management system for real-time data collection, analysis, visualization and device management; an integrity monitoring scheme based on blockchain to prevent improper delivery of controls and insights; and software for virtual switching that supports software-defined networking technologies to improve network manageability.

- *Chapter 5* presents an FL-based IDS system, named FELIDS, designed to secure agricultural-IoT implementations. This system ensures security and privacy by implementing local learning, allowing devices to learn from their peers and share only model updates with an aggregation server that produces an accurate detection end model.

- *Chapter 6* introduce a decentralized, FL-based secure and differentially private IDS for IoT and IIoT security. Our suggested system enforces the participating clients' confidentiality in two ways, primarily through a quantum-resistant Ring-Learning With Errors (R-LWE) key exchange mechanism for outbound as well as inbound differential confidentiality, to prevent confidentiality leakage from the clients' local gradients. We also consider a completely decentralized aggregation pattern in the suggested system to overcome the single-point-of-failure threat posed by the aggregation server in the conventional FL.

Chapter 1

Smart Agriculture: a holistic perspective

1.1 Introduction

Agriculture has played a crucial role in the progress of societies throughout the ages. According to estimates from the United Nations (UN) [1], the global population is projected to grow reaching approximately 11 billion in the coming century. This rapid population growth poses significant challenges for the planet in terms of supporting such a vast population and meeting their food and water demands. This means that global food and water requirements will continue to rise. Speaking of consumption patterns, it is noteworthy that agriculture stands as the heaviest water consumer, utilizing approximately 70% of the planet's annual usage of water [9]. This immense water consumption is largely attributed to activities like irrigation, which is essential for agricultural practices. However, this process can also lead to water pollution, as certain agricultural activities release significant amounts of nutrients, pesticides, and other harmful substances into water sources.

At first glance, it may seem logical to scale up food production to satisfy the demands of a constantly growing world population. However, the Food and Agriculture Organization of the United Nations (FAO) considers that eradicating global hunger and safeguarding food supplies does not inevitably require a sharp increase in agricultural production, even by 50%. [3]. Instead, a crucial aspect lies in making agricultural production systems more sustainable [2]. This objective forms a significant part of smart agriculture initiatives. To achieve the principles of sustainable agriculture, extensive research, innovation, and technological advancements are imperative. Embracing these factors to their fullest potential can pave the way for sustainability and more efficiency throughout smart agricultural systems.

In this chapter, we provide an overview of the field, starting with a description of what it is about, followed by a thorough look at the fundamental concepts that shape it. Finally, we present a taxonomy of various smart agricultural applications.

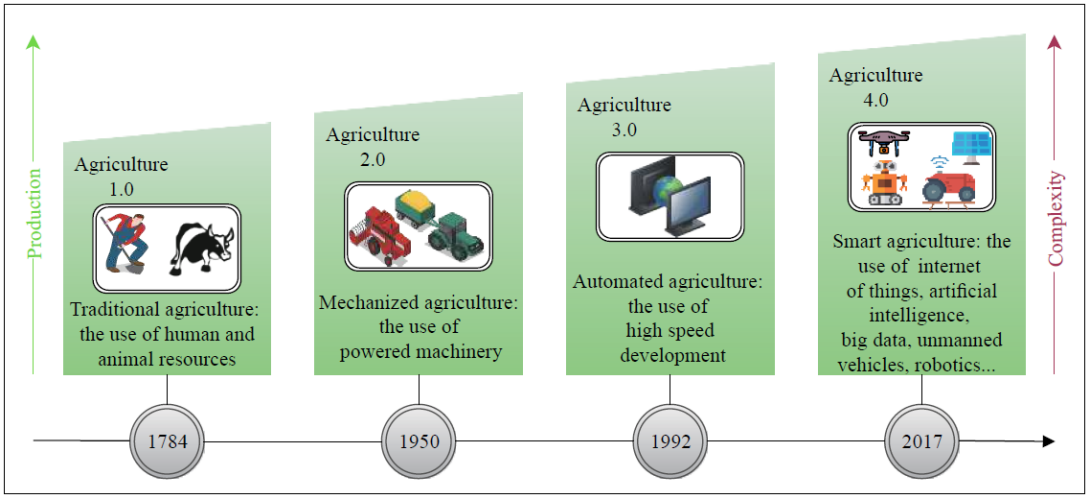


Figure 1.1: The four agricultural revolutions

1.2 Definition

Over the history of agricultural advancement, four distinct revolutions have taken place [10], namely: 1) the era of classical agriculture, characterized by the physical power of humans and animals, 2) the era of machinized agriculture, characterized by rumbling noises, 3) the era of computerized agriculture, characterized by speed development, and 4) the era of intelligent[1] agriculture, characterized by emerging technologies, as illustrated in Fig. 1.1 [4].

In recent years, agriculture has witnessed significant advancements through the integration of technologies such as the Internet of Things (IoT), cost-effective and improved sensors, advanced actuators, high-speed wireless technologies, Artificial Intelligence (AI), robotics, and more. The application of these sophisticated technologies in agriculture empower farmers with technological tools to enhance their decision-making processes and automate operations. By providing products, knowledge, and services, these technologies contribute to improved productivity, enhanced product quality, increased profitability, and ultimately transform the agricultural sector into a *smart* one [4]. Moreover, smart agriculture also offers the prospect of a sustainable future through the utilization of technology. This involves leveraging Information and Communication Technologies (ICT) within the cyber-physical cycle of agricultural operations to enhance precision in various tasks. For instance, this may involve providing each plant or animal with precisely what it requires to grow optimally, leading to optimized overall performance while minimizing waste, inputs, and pollution. By employing technology to achieve such precision, smart agriculture strives for a more environmentally friendly and resource-efficient approach to farming practices.

[1]We use *smart* and *intelligent* interchangeably throughout the text.

1.2.1 Real-life scenarios

To gain a clear view of this research area, we can closely examine some successful real-world stories that add to the overall worth of smart agriculture. For instance, as a part of the European DrainUse project, a greenhouse system that meets soilless conditions was implemented in Spain, using low salinity water [11]. The physical deployment of the system along with its architecture is presented in Fig. 1.2 (a). The project is composed of three layers, the first of which is a local Cyber-Physical System (CPS) that collects data and executes real-time tasks. The second is an edge layer, charged with controlling the virtualized nodes and enhancing system reliability in the event of network access failure. The final layer is a cloud layer, which conducts advanced computation and data analysis for enhanced decision making. Based on two cycles of the tomato crop, the project was successfully tested and showed water savings of more than 30%, and up to 80% for some nutrients.

(a) Smart Greenhouse system [11]

(b) Intelligent dairy farming [12]

Figure 1.2: Real-life examples of smart agricultural practices

Another example would be an intelligent dairy farm system, called MooCare [12], which has been implemented in Brazil, to help dairy farmers attain better productivity rates through the analysis of their production of milk and the provision of automatic and customized feeding schedules. Using the IoT collected data, milk production is predicted for each individual cow. By doing so, farmers can be knowledgeable ahead of time, and thus better respond in terms of building a better nutritional plan where each cow can enjoy a customized diet. The system is composed of two core modules, MooField and MooServer,

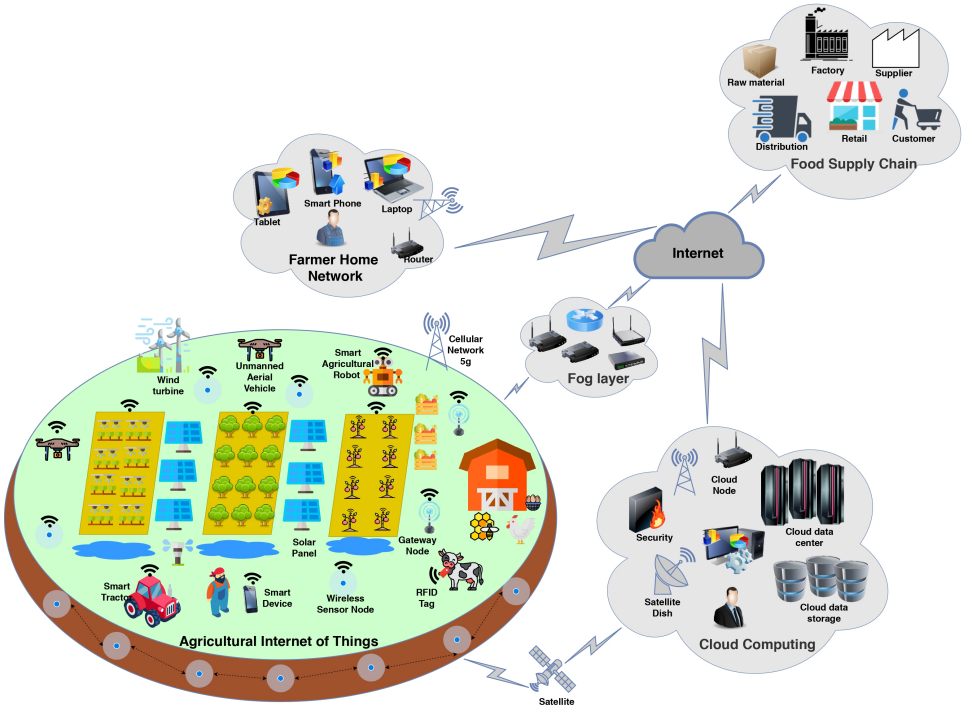


Figure 1.3: Illustrative example of smart agriculture

as shown in Fig. 1.2 (b). The first module is responsible for data collection, feeding, and identification of livestock, while the second one is a central controller responsible for data storage, visualization, and processing, as well as a prediction engine, feeding, and notification services. The prediction accuracy of the MooCare model reached 94.3%, demonstrating its ability to predict milk production adequately.

1.3 Architecture, protocols, and technologies

The development of the Internet of Things (IoT) has been a crucial factor in shaping and driving advancements in the smart agriculture sector [4]. The IoT is envisioned as a fundamental element of the future internet, comprising billions of interconnected and intelligent "things." While the definition of these "things" has evolved with technological progress, the core objective has consistently been to process and interpret computer information autonomously, without requiring direct human intervention [13]. Currently, there is no standardized architecture for IoT-based applications, including those in agriculture. Various scholars have proposed different architectures to cater to specific needs and contexts [14]. Fig. 1.3 provides an illustration of a smart agricultural system. In this context, we adopt the architecture presented in [4], which consists of 5 layers (e,i, physical, network, middleware, service, and application). These layers collectively form the foundation for an efficient and effective IoT-based agricultural application.

1.3.1 Physical layer

Also referred to by the layer os perception. It incorporates different types of sensors, actuators, Radio Frequency Identification (RFID), Wireless Sensor Networks (WSNs), etc [15]. The main task here is to capture valuable data from the surrounding environment, via deployed smart devices, and pass the processed data to the layer on top. In addition, it accept incoming control directives from the application layer, so that the concerning assets, like agricultural equipment, can perform the required actions. The most popular types of deployed equipment fall into one of the following categories:

Project	Type	WingSpan	Payload	Endurance	Coverage	Max. Speed	Main App.
ALTI Reach	HFW	6 meters	7 Kg	20 Hours	1800Km	90Km/h	Aerial imagery
AgEagle RX-60	FW	1.37 meters	N/A	60 Minutes	1.6 Km	42 m/h	Aerial imagery
M600 Pro	MR	N/A	6 Kg	35 Minutes	5 km	65 km/h	Aerial Imaging
Omni Ag	MR	N/A	2 Kg	25 minutes	1.60 km	15 m/s	Aerial imagery
eBee SQ	FW	1.10 meters	N/A	55 Minutes	41 km	110 km/h	Aerial imagery
THEA 140 Pro	MR	N/A	5 Kg	5 Hours	2km	14 m/s	Liquids Spraying
ALTI Ascend	HFW	2 meters	600 g	6 Hours	450Km	75Km/h	Aerial imagery
Agras T16	MR	N/A	16 Kg	18 minutes	0.1 km	7m/s	Liquids Spraying

Hybrid Fixed-Wing (HFW); Multi-Rotor (MR); Fixed-Wing (FW)

Table 1.1: A short side-by-side comparison of selected UAVs used in smart agriculture

Programmable devices

In this category, we highlight three of the most used device types in smart agriculture, namely, sensors (e.g. PH sensors), actuators (e.g. AC motors), and hardware boards [4]. The main role of the deployed sensors is to perceive a specific environmental situation, such as soil moisture, and produce information accordingly. As for the actuators, their role is to execute a specific set of controlled actions, such as opening a water valve. Meanwhile, hardware boards are usually applied to automatically control the two previous types. These devices have common foundational components, such as input and output units, cores for data processing, network modules, and power management blocks [16]. In addition, as required by specific applications, other components may exist, such as unites for power generation, mobilizers, and so on.

Unmanned Systems

An Unmanned System (US) (or an Unmanned Vehicle (UV)), is described as "*a powered physical system, with no human operator aboard the principal components, which acts in the physical world to accomplish assigned tasks*" [17]. The main benefit of these systems is that they are able to operate autonomously. There are different types of US, which are typically named after their operating environment, including Unmanned Aerial Vehicle (UAV) (or drones), Unmanned Ground Vehicle (UGV), Unmanned Surface Vehicle (USV), and Unmanned Underwater Vehicle (UUV). The first two types are the most widely used for smart farming, namely UAVs and UGVs. UAVs applications in agriculture include liquids spraying, hyper-spectral imaging, weed detection, and data acquisition from deployed sensors [4]. Used UGVs include unmanned agricultural machinery such as

unmanned tractors [14]. Tab. 1.2 provides a selected list of lead commercial UAVs used for smart agriculture [4].

Range	Technology	Frequency	Data Rate	Power	Range	Security
Short range	NFC	13.56 MHz	106 kbps-424kbps	1-2 mW	0.1 m	N/A
	RFID	13.56 MHz	423Kbps	1 mW	1 m	N/A
	Zigbee	2.4 GHz	250 Kbps	1 mW	20 m	AES-128 Bit
	Z-Wave	908.42 MHz	100 Kbps	1 mW	30 m	Security 2 (S2)
	Wi-Fi	2.4GHz-60GHz	1.2Mbps-6.75Gbps	1W	100 m	WEP/WPA2
	Bluetooth	2.45 GHz	1-3 Mbps	1 W	100 m	AES 56/128 bit
	Bluetooth LE	2.4 GHz	1Mbps	10-500mW	100 m	AES-128 bit
	6LowPAN	908.42 MHz-2.4 GHz	20 Kbps–250 Kbps	1 mW	100 m	AES-128 Bit
Long range	LoRaWAN	LoRaWAN	0.3-50 Kbps	Very Low	10 Km	AES-128 bit
	SigFox	908.42 MHz	10-1000 bps	Very Low	50 Km	AES-128 bit
	NB-IoT	180 KHz	200 Kb/s	Very Low	15 Km	LTE encryption
Cellular area	2G	850-1900 MHz	171-384 Kbps	1-3 W	26 Km	GEA2,3,4/A5/3,4
	3G	850-1900 MHz	40 0.73-56 Mbps,	1 W-4 W	26 Km	USIM
	4G	700-2600 MHz	0.1-1 Gbps	1 W-5 W	28 Km	SNOW 3G
	5G	700 MHz - 72 GHz	20 Gbps	1 W-5 W	28 Km	256-bit

Table 1.2: Different wireless technologies used in smart agriculture

1.3.2 The Networking Layer

The physical layer of the IoT-based smart agricultural system serves as the receiver for collected data from sensors and devices deployed in the field. It is responsible for transmitting this data to the higher layers for further processing and analysis. Additionally, the physical layer facilitates the transmission of control directives from the upper layer (application) to the lower layer (perception), enabling actions to be taken based on the processed data. Within this layer, communication technologies can be classified into three groups: *Short-range versions* encompass communication technologies like ZigBee, Bluetooth, Wi-Fi, and NFC. These technologies are suitable for relatively close-range communications between devices in the vicinity. *Long-range versions* include technologies such as LoRaWAN, SigFox, and NB-IoT. These are designed to enable long-distance communication between devices, making them suitable for large-scale agricultural operations or applications covering extensive areas. *Cellular versions* span from 2G to 5G technologies. These cellular networks offer varying degrees of data transmission capabilities and coverage, making them flexible for different agricultural scenarios.

For a concise summary of selected wireless technologies categorized by transmission range, please refer to Table 1.2 as provided in the reference [4].

1.3.3 Middleware layer

This layer is typically conceptualized as a software system designed to be used as an interface between IoT devices and related applications [15], where system or hardware technical complexities are abstracted away to enable easy and flexible development of various IoT-based agricultural solutions. For instance, the work in [18] proposed a contextual middleware framework for responsive, scalable, service-oriented, embedded IoT systems. The main objective of the proposed middleware is to streamline the process of administering, handling and exchanging the large amount of miscellaneous data produced

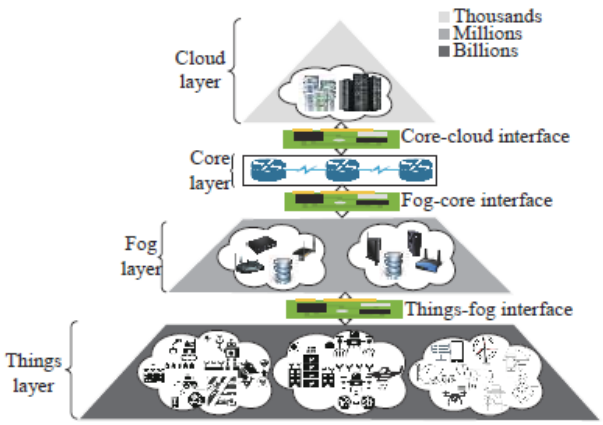


Figure 1.4: IoT, fog, and cloud computing for smart agriculture

by numerous IoT-based smart agricultural devices. There are various types of middleware approaches, including application-specific, agent-driven, and virtual machine-based. However, according to the authors, a key area of research in Agriculture 4.0 is context-aware middleware, given its ability in providing the convenience of flexibility, scalability, and extensibility for any application in any complex environment.

1.3.4 Service layer

Across this layer, miscellaneous services such as cloud, fog, AI, SDN, and Network Functions Virtualization (NFV), are delivered to the application layer, empowering smart agricultural applications with a wide range of advanced operations. Below we briefly describe some of these technologies along with practical cases of their application to smart agriculture.

Cloud and Fog

Cloud computing offers significant processing and storage capabilities that can be harnessed to benefit resource-constrained hardware devices, such as IoT devices in the context of smart agriculture. IoT with cloud integration has emerged as a promising technology with successful applications in various industries, including agriculture. An example of such a system is presented in [19], where agricultural information is provided as a service through the utilization of cloud computing and IoT. This system collects diverse agricultural data from multiple users via IoT devices deployed in different regions. Subsequently, it delivers the necessary information to users and automatically diagnoses the agricultural situation. The results indicate several advantages, including a 12.46% reduction in cost, a 15.52% network bandwidth decrease, a 10.18% improvement in operation time, and a 13.32% reduction in latency.

Another important concept in this context is fog computing, which acts as a virtualization platform that bridges end devices and cloud data centers, usually located at the network edge. Unlike conventional cloud computing, fog computing allows certain processing tasks to be performed at the network edge, leveraging the storage, computing, and

network connectivity capabilities of edge devices. This approach provides several benefits, such as proximity and geographical dispersion, as illustrated in Fig. 1.4, overcoming challenges related to network traffic, security, and privacy concerns that are commonly associated with traditional cloud paradigms [4].

A real-world case study concerning a soil-less greenhouse system exemplifies the implementation of an edge-enabled IoT-based smart agricultural platform [11]. This highlights how fog computing can be leveraged to create a robust and efficient smart agriculture solution.

AI, ML, and DL

Although the term *Artificial Intelligence* (AI) was coined in 1956, specifically by John McCarthy, who is also credited with solidifying the direction of the field as a starting point at the Dartmouth Summer Research Project on Artificial Intelligence conference [20], the concept of artificial intelligence predates that date by a very long time. AI has become more popular today, primarily due to improvements in computing power and storage, a major bottleneck that has been holding back the development of the field due to the extremely high cost of computers and the lack of large storage capacities. AI technology is currently an essential contributor to boosting efficiency and profitability in various sectors, including agriculture. There are problems in agriculture such as disease prediction and identification, weed spotting, smart water management, and predictive analysis [15], where AI can offer efficient solutions [4]. For instance, the work in [21] developed an expert system for guiding farmers in making assessments of suitable cultivable farmland, by collecting data from various sensors that feeds an AI model. This research has produced results with an accuracy of up to 99%.

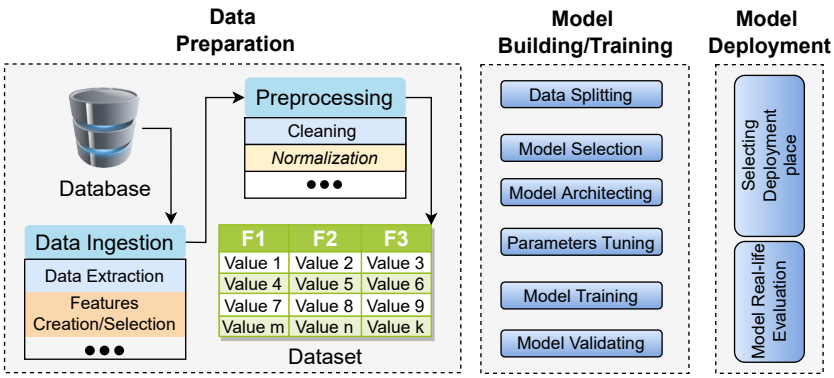


Figure 1.5: Simplified View of Machine Learning Life-cycle

Machine Learning (ML) is a class of AI that concentrates on leveraging data and algorithms to impersonate how humans learn and employ it for machines to support learning, understanding, and decision-making features. Fig. 1.5 illustrates the life-cycle for creating and deploying an ML system (called "model"). The first important step in creating models is to have appropriate data on which we can train the model. The steps to do this are outlined in the data preparation step. The first thing to do is to gather large amounts of data, and then this data will be ingested and pre-processed to be used for training the model. This includes cleaning and normalizing the data. Once the data is

prepared, we move to the Model Building stage, where data is divided into training and test sets. The model selection process is based on the specific task. Next, comes the model building and training workflow, during which the model architecture and hyperparameter tuning take place. Once the model is created and evaluated, it is ready to be deployed in a real environment. Types of ML are generally classified into four categories [22], namely:

- *Supervised learning:* where the dataset is pre-labeled, meaning that the data has one or more features that identify the data samples, and the model's task is to correctly classify those data samples.
- *Unsupervised learning:* where the dataset used is unlabeled and the model's task is to self-identify potential patterns and connections within the data.
- *Semi-supervised learning:* where the dataset consists of both structured and unstructured data, which helps the algorithm to make its decisions on its own.
- *Reinforcement learning:* where the dataset employs a so-called "reward/punishment" approach, offering the algorithm the necessary input to enable the learning from experience through trial and error.

It is worth mentioning that one of the most successful forms of AI today is Deep Learning (DL) [23]. One explanation for DL success could be the fact that we are in the *big data era*, where there is a massive amount of data available to work with. DL is a class of ML that seeks to imitate the brain of humans. To accomplish this, DL operates on a multi-layered algorithmic structure called Artificial Neural Networks (ANNs), which is composed essentially of three layers, the first is for input, the last is for output, and in the middle, there are one or several hidden layers [23]. In each layer, there are several nodes called artificial neurons, which interconnect with each other and are assigned a weight and a threshold. Once the output of a single node is above a specified threshold value, it is activated and transmits data to the following layer. Deep neural networks (DNNs) are artificially constructed neural networks made up of several layers of interlinked nodes (or neurons). Unlike traditional ANNs, which typically have only one or two hidden layers, DNNs typically have at least three hidden layers and may have many more. The architecture of a DNN can be represented mathematically as a function that maps input data to output predictions through a series of transformations. Let's consider a DNN with L layers, where the input and output are represented by 2 vectors (x and y). Each layer of the network is characterized by a set of weights and biases that define the linear and non-linear transformations applied to the input data. We can represent the output of the l^{th} layer of the network using $[z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}]$. Where $z^{[l]}$ is the input to the l^{th} layer, $W^{[l]}$ is the weight matrix for the l^{th} layer, $a^{[l-1]}$ is the output of the $(l-1)^{th}$ layer, and $b^{[l]}$ is the bias vector for the l^{th} layer. The output of the l^{th} layer is then forwarded through an activation function, such as the sigmoid, to pass the output of the next layer, by $[a^{[l]} = g(z^{[l]})]$. Where g is the activation function. The output of the final layer of the network, y, is then given by: $[y = a^{[L]}]$. where L is the number of layers in the network. The weights and biases of the network are typically learned using a process called backpropagation, which consists of iterative adjustments to the weights and biases to minimize a cost function, that measures the error between the predicted output and

the true output. This process is typically done using an optimization algorithm, such as stochastic gradient descent.

DNNs have shown great success in a wide range of applications, including computer Natural Language Processing (NLP), vision and speech recognition, among others. Their ability to learn complex and abstract features from raw input data has made them a powerful tool for solving a wide range of ML problems.

Below is a brief overview of the most common types of neural networks for supervised learning.

- *Convolutional Neural Network (CNN):* are a specialized type of neural network that are particularly well-suited for processing data with a grid-like topology, such as images [23]. The basic building block of a CNN is the convolutional layer, which applies a set of filters or kernels to the input data. Each filter corresponds to a different feature that the network is trying to learn, and the output of the convolutional layer is a set of feature maps that highlight where in the input data each feature is present. The architecture of a CNN typically consists of multiple convolutional layers, interspersed with pooling layers, which downsample the feature maps to reduce their spatial size. The output of the final pooling layer is then flattened into a vector and passed through one or more fully connected layers, which perform the final classification or regression. The parameters of the CNN, including the filter weights and biases, are typically learned using backpropagation and an optimization algorithm such as stochastic gradient descent.

- *Recurrent Neural Network (RNN):* are a type of neural network that is suited for sequential data, such as text, and speech [23]. Unlike feedforward neural networks, RNNs are able to maintain a "memory" of past inputs and use that memory to inform their predictions. The key feature of an RNN is the recurrent connection, which allows the network to pass information from one time step to the next. The basic building block of an RNN is the recurrent cell, which takes an input at each time step and produces an output and a hidden state, which is fed back into the cell at the next time step. The architecture of an RNN can take several forms, including the basic RNN, the Long Short-Term Memory (LSTM) network, and the Gated Recurrent Unit (GRU). LSTMs and GRUs are particularly well-suited for handling long-term dependencies in the input data, which can be difficult for basic RNNs.

Smart agriculture also benefits from these technological breakthroughs. For instance, the research in [24] proposed a CNN-based plant seedling classification scheme, that was implemented using a dataset of approximately 5000 snapshots of 960 unique plants, of 12 species. The proposed model achieved an accuracy of 99.48%.

NFV and SDN

The explosive raise in the number of smartphone users and the increased use of cloud services are some of the key drivers of emerging networking trends. These in turn are

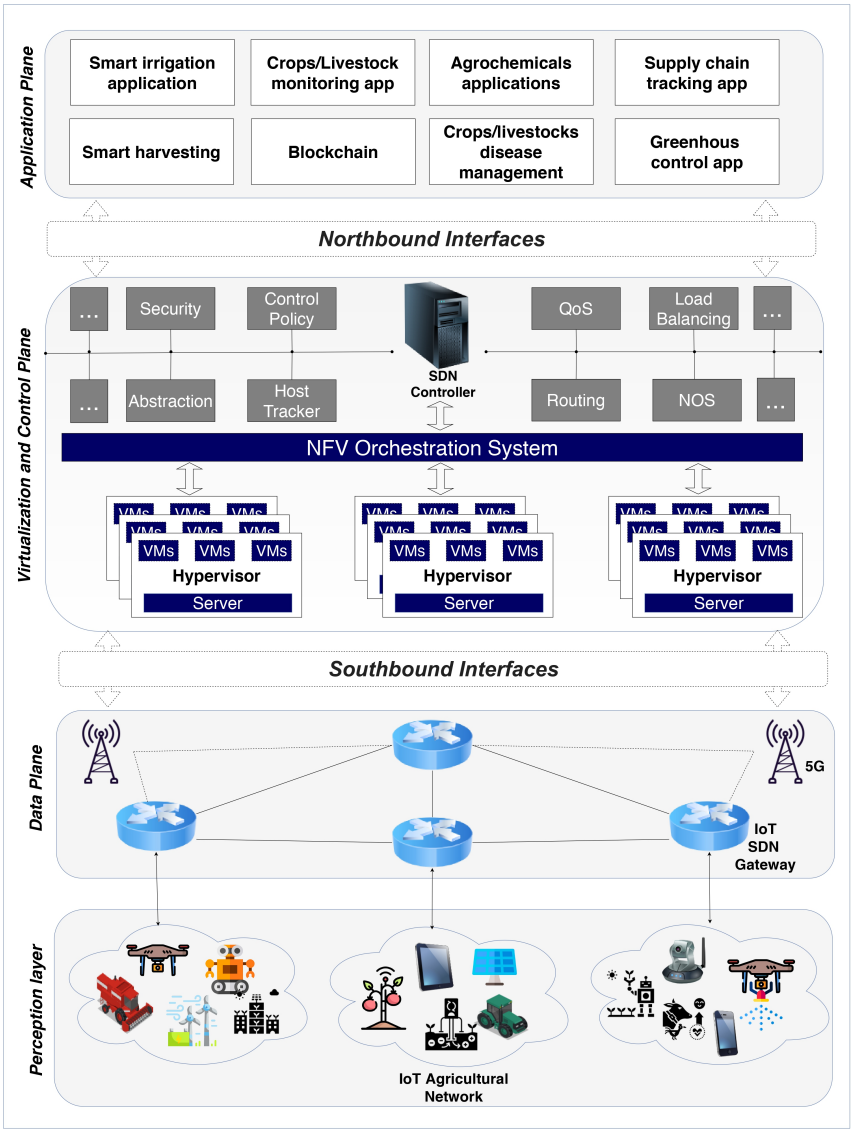


Figure 1.6: SDN and NFV in smart agriculture

pushing for a reconsideration of conventional network architectures. One of the proposed solutions is SDN, which is intended to decouple the transmission functionality (the data plane) from the control functionality of the network (the control plane), rendering the entire network intelligent and centrally controlled or programmed [25]. Another solution is Network Functions Virtualization (NFV) which focuses on abstracting the network forwarding and related network operation from the underlying hardware on which it function and make a virtual network through software, that performs path control functions [26]. Coupling SDN and NFV capabilities would potentially enhance infrastructural flexibility, and facilitate the dynamic, adaptive design, provisioning, and operation of network services, something that is required for IoT services. Fig. 1.6 illustrate such combination for smart agriculture [4].

The data plane includes network assets like switches and routers for packet routing. Nevertheless, as opposed to conventional networks, they are simply forwarding components with no embedded intelligence for independent decision-making. The SDN con-

troller defines the packet routing logic and applies it to the forwarding devices. It logically manages the network, handles application layer requests, and administers network devices through standard protocols. The southbound interface is used to connect the transfer devices to the SDN controller, while the northbound interface is dedicated to application development. The NFV framework involves a dedicated backbone server stack installed on hypervisors that support multiple virtual machines which carry out networking functions [26]. The SDN controller, combined with the NFV orchestration system, is the logical control module.

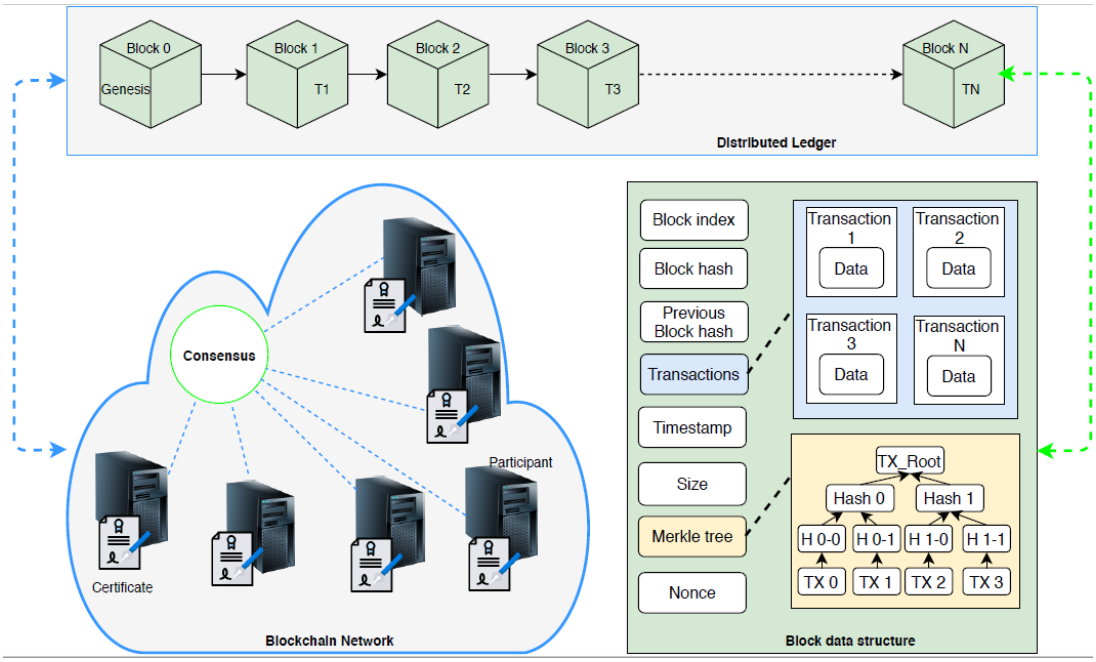


Figure 1.7: Blockchain technology's main building blocks

Blockchain Technology

Blockchain technology operates as a distributed multi-user system, where users are spread across a network and collaborate to share information. Instead of relying on a central intermediary, these users reach consensus through an agreed-upon exchange protocol called a consensus algorithm. This decentralized approach enables peer-to-peer trust and facilitates direct transactions between users, fostering a trustless and transparent environment [27].

The construction of blocks in a blockchain-based system involves the use of a consensus mechanism, cryptographic hash functions, and digital signatures, as illustrated in Fig. 1.7. These components ensure the security, integrity, and immutability of the data stored within the blockchain. Numerous blockchain-based systems have been proposed and implemented for smart agriculture, with a significant focus on food supply chain management [4].

A case in point is AgriBlockIoT, featured in [28], which serves as a decentralized blockchain-based tracking solution for food supply chain management. This system facilitates the seamless integration of IoT appliances along the entire food supply chain, en-

abling transparent, fault-tolerant, immutable and verifiable records. AgriBlockIoT relies on at least two distinct blockchain implementations, namely Ethereum and Hyperledger Sawtooth, to accomplish its goals efficiently.

1.3.5 Application layer

This layer is responsible for receiving data from the lower layers and providing the corresponding information to specific users. Typically, applications at this layer have sophisticated visualization utilities and front-end interfaces for a satisfactory user experience. And often they feature an Application Programming Interface (API) to enable communication with other programs. A thorough overview of the types of applications that have been developed along with their purpose is presented in depth in the following section.

Specific-purpose protocols

Aside from general-purpose exchange protocols like Hypertext Transfer Protocol (HTTP), IoT-based agricultural applications utilize specific-purpose application protocols. These protocols are designed to be lightweight and suitable for resource-constrained IoT devices. Two common examples of such protocols are the CoAP and MQTT [4]. CoAP is built on the principles of Representational State Transfer (REST) and operates over the User Datagram Protocol (UDP). Its combination with UDP enhances its efficiency for IoT applications. CoAP is suited for resource-constrained devices due to its ability to minimize data transmission overhead, facilitating seamless communication between IoT devices and servers. While MQTT is a lightweight and straightforward publish/subscribe messaging protocol. It is specifically designed for use in networks with high latency or low bandwidth, making it a good choice for IoT devices operating under challenging and unreliable network conditions. MQTT enables asynchronous communication between IoT devices and servers, allowing devices to publish data and other devices or applications to subscribe to specific topics of interest. This approach reduces data overhead and ensures smooth data flow within IoT applications.

1.4 Smart agriculture applications

A variety of applications have been implemented under the umbrella of smart agriculture, targeting virtually every aspect of the sector, ranging from simple routines like surveillance functions, all the way to sophisticated, more complex workflows such as food supply chain management. A graphic illustration representing the classification of smart agriculture applications, along with their sub-classes, is shown in Fig. 1.8. A thorough investigation of currently available application solutions for smart agriculture is the basis for this classification [4]. A brief explanation of each component is given below. In Tab. 1.3 we provide a short outline of the core technologies used in each layer for different selected works across the world [11, 29, 30, 31, 32, 33, 34, 35, 36], along with the contribution, performance, and limitations [4].

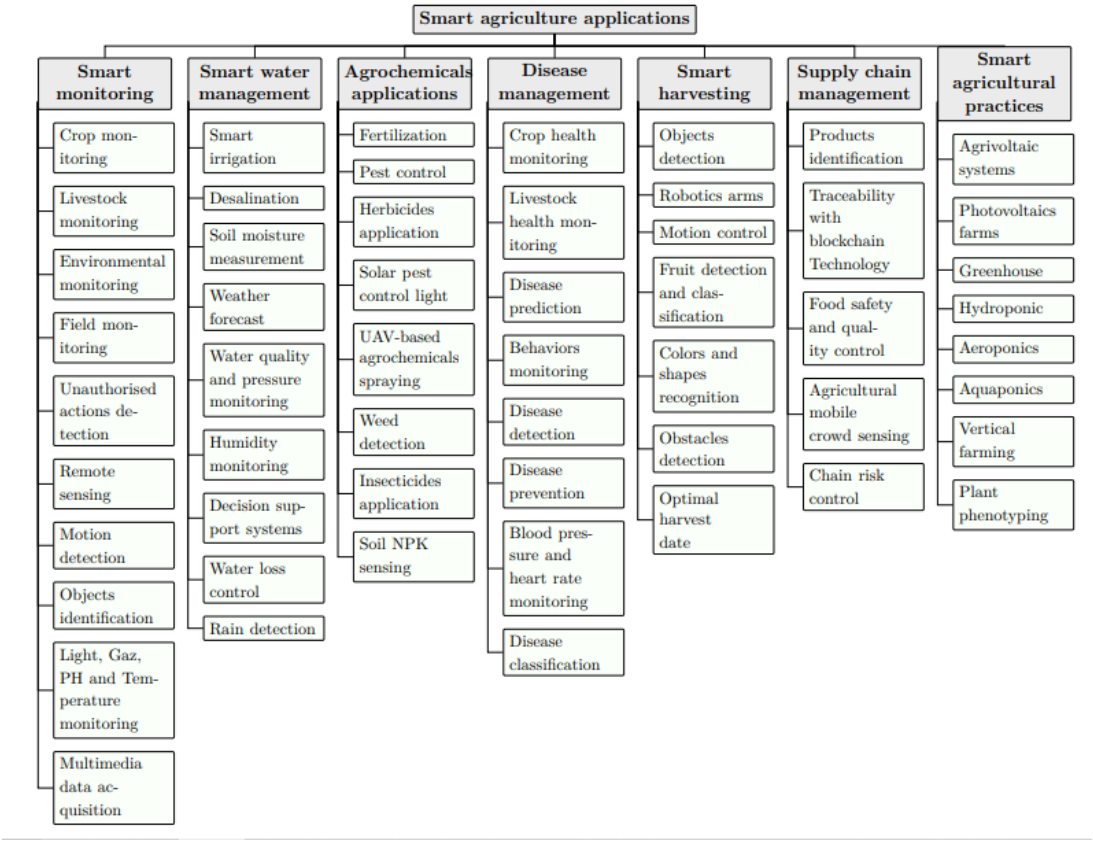


Figure 1.8: Smart agriculture applications

1.4.1 Smart monitoring

Intelligent monitoring systems are instrumental in maintaining optimal conditions for a better quality of agricultural products. Over the past years, there has been an increase in the research and development of agricultural surveillance systems [29, 11, 35], which include 1) *crop monitoring:* these applications focus on tracking crop growth and production performance at all phases of their development, which is an critical aspect of smart farm management. 2) *livestock monitoring:* smart farming equips the grower with the means of monitoring and keeping track of their livestock. Through IoT devices, the farmer can remotely maintain awareness about how things are going. 3) *environmental monitoring:* technology is a key part of knowing better the physical world by generating real-time data about land, air, and water. Such applications serve as a means of tracking environmental conditions which directly impact agricultural products. 4) *field monitoring:* outdoor sensors are used to capture data and forward it to the processing units, where corresponding software applications will be used to interpret the operational data. 5) *unauthorized actions detection:* these applications are crucial in keeping the farm safe from unwanted guests, including animals, insects, and even humans. 6) *remote sensing:* these applications involve radiation monitoring, for example, using drones equipped with 3D mapping utilities with aerial imagery. 7) *motion detection:* these applications use sensors that detect nearby moving objects, usually when there is a need to detect movement in surveyed zones for safety precautions. 8) *objects identification:* In particular, the use of AI for object classification and identification are powering such applications,

for example, by enabling environment-specific vision awareness in machines deployed in the agricultural space. 9) *light, gas, PH, and temperature monitoring:* such applications are intended to do environmental measurements of specific conditions, including CO2 amounts, PH levels, and nutrient levels. 10) *multimedia data acquisition:* is the activity of collecting multimedia inputs, for example, images and videos, whereby the real-world physical conditions are captured and refined to produce helpful information.

1.4.2 Smart water management

One important advantage of introducing intelligence into a specific sector is the ability to manage resources efficiently. Such applications tend to improve the management of water resources and obtain optimal and cost-effective outcomes [32, 36]. The judicious usage of water is essential in agriculture to both enhancing yields and reducing costs, while also representing a critical step towards sustainability [33]. Smart water management applications include 1) *smart irrigation:* such applications are designed to optimize the distribution of water on the farm according to precise needs and at the right time, thereby increasing product quality and reducing waste. 2) *desalination:* which is a method of treating seawater or salt water to obtain fresh water through desalination facilities. This process is particularly advantageous for the agricultural sector since it allows to obtain of sustainable fresh water in regions where there is a lack of other water sources. 3) *soil moisture measurement:* where having information on soil moisture status allows for greatly improved irrigation scheduling, providing the necessary water to the monitored soil, and avoiding water wastage in the context of already existing requirements. 4) *weather forecast:* these applications provide a key component for irrigation scheduling, which involves coordinating the timing and amount of water used to irrigate crops in order to maximize profits. 5) *water quality monitoring:* the main goal of these applications is to understand the chemical and physical characteristics of water and to detect and pinpoint leaks and failures in irrigation systems. 6) *humidity monitoring:* where in some situations, such as indoor agricultural systems (e.g. greenhouses), it is essential to measure specific environmental conditions such as temperature and humidity as they greatly affect agricultural products. These applications use a special sensor to measure humidity and report changes in real-time. 7) *decision support systems:* The primary function of these systems is to establish an optimal irrigation schedule and to make informed decisions about irrigation actions in real-time. They are equipped with large amounts of weather data and targeted crop information to make the appropriate choices. 8) *loss control:* is the practice of avoiding water leakage or wasted irrigation using IoT technologies. 9) *rain detection:* is implemented using IoT sensors to detect the occurrence of unpredictable rainfall.

1.4.3 Agrochemicals applications

Agrochemicals, such as pesticides, herbicides, insecticides, and fungicides, are essential agricultural chemicals used to combat insects, prevent diseases, and promote plant growth. They are referred to as pesticides and fertilizers, and play a significant role in modern agriculture by reducing crop losses and supporting crop productivity. However, it is crucial

	Contribution	P. Layer	N. Layer	M. Layer	S. Layer	A. Layer	Performance (+) Limitation (-)
[29]	Advanced UAV–WSN System for Intelligent agricultural monitoring	- Soil and Weather related sensors - Satellite	- 6LoWPAN, ZigBee - LoRaWAN, GSM - BLE, Wi-Fi	N/A	- Edge/Fog Computing - Cloud Computing	- User server for data interpretation	(+) Design of optimized trajectories that allows efficient use of limited ground sensor network resources (-) Increased complexity for multilevel data processing
[30]	Blockchain-based fish farm platform	- Temperature, Water Level, O2 Sensor, pH - Water Pump, pond heater, fish feeder	- LoRaWAN, ZigBee, Z-Wave , Bluetooth, Wi-Fi	N/A	- Cloud computing - Fog computing	- Blockchain - HTTP - Web application	(+) Scalability, high throughput, off-chain storage, and privacy (-) Application too complex for ordinary farmers
[31]	Automatic crop disease recognition system	- Weather-related sensors - Cameras	- N/A	- N/A	- AI - Big data analytics	- Application for the visualization of crop disease identified	(+) Identification accuracy of 97.5% (-) Unbalanced data structure has not been well solved
[32]	IoT Architecture for Water Resource Management in Agro-industrial Environments	- Variety of sensor and actuator technologies for soil, plant and weather activities.	- 2G, 3G	FIWARE Middleware	- Cloud Computing	- CoAP & MQTT - HTTP - Web-based Application.	(+) 75% of the operational cost could be saved. (-) The speed of computing and latency could be better on edge computing approach
[33]	IoT-based smart water management platform	- Variety of commercial sensor and actuator technologies for soil, plant and weather. - UAV	- LoRaWan - Wi-Fi - 2G, 3G, 4G	FIWARE middleware	- Cloud computing - Fog computing - Big data analytics - AI	- MQTT - Web application	(+) Real-time responses for adapting irrigation (-) Savings in consumption are not analyzed compared to Zamora-Izquierdo *et al.*[11]
[11]	Smart farming platform	- Sensors: light, humidity, temperature, CO2, PH. - Actuators: soil and water nutrition pumps, valves	- 6LowPAN - Serial/direct digital /analog I/O connections	FIWARE Middleware	-Cloud computing -Fog computing -Big data analytics	-MQTT, CoAP - Greenhouse control Web service	(+) Savings of more than 30 % in water consumption and up to 80 % in some nutrients (-) Real-time responses are not considered
[34]	Smart microservice IoT-Based supply chain management system	- N/A	- N/A	Node-RED	- AI	- HTTP -Web-based microservice	(+) Reduced overall spin-up time. (-) Did not discuss security and privacy
[35]	Smart Edge-IoT-based platform for livestock and crops monitoring	-RFID - Sensors : Weather, Soil, livestock, and Transport sensors.	- SigFox, LoRa, ZigBee, Bluetooth, Wi-Fi, 3G, and others	FIWARE Middleware	-Edge computing -Cloud computing -AI -Big data analytics	-Blockchain - Web application	(+) The introduction of Edge nodes improves the reliability of communications and reduced the costs (-) Consumers cannot access and analyze all the data in the system
[36]	IoT-based smart irrigation management system	- WSN - Sensors: Soil moisture and temperature, Precipitation, Air temperature, light radiation, humidity, Actuators: Water pump	- Wi-Fi, ZigBee -Mobile Data connection	N/A	- Cloud computing -AI	- Web-based interface for real-time monitoring - HTTP REST API	(+) The system is cost-effective, as it is based on the open standard technologies (-) Water saving analysis isn't provided

Table 1.3: A selection of smart agriculture applications

to use these chemicals properly, as improper usage can lead to adverse effects on human and environment. Smart agriculture aims to harness the benefits of agrochemicals while effectively managing their potential negative impacts. The applications of agrochemicals include: 1) *Fertilization:* Fertilizers are extensively used in agriculture and typically contain 3 primary nutrients (nitrogen, phosphorus, and potassium). These nutrients are crucial for supporting plant growth and ensuring optimal crop development. The goal of these applications is to help farmers determine the proportions of these elements in the soil for better fertilization. 2) *pest control:* in these applications, sensors are used to automatically collect relative data, such as if a pest is present or if a trap is activated to signal the capture of a pest. 3) *herbicides application:* these applications are also called weed killers because they target weeds in the agricultural crop. One among the most famous weed control techniques is herbicide spraying. 4) *solar pest control light:* is an ecological method of pest control using solar insecticide lamps (SIL). 5) *UAV-based agrochemicals spraying:* by using UAVs, the resources (cost and time) of manual spraying and sprayer rentals can be significantly reduced. And this is the main objective of such applications. 6) *Weed detection:* weeds have the potential to be a significant factor affecting crop yields. ML, in conjunction with image processing techniques, is becoming a powerful and exciting tool for the accurate, real-time identification of weeds and crops in the agricultural field. 7) *insecticides application:* these applications are used to eliminate insects in agriculture; yet, they can also harm crops. Consequently, there are opportunities for IoT to contribute to reducing the unnecessary use of chemicals. 8) *soil NPK sensing:* which is among the primary components of soil analysis for fertilization is nutrient level determination in the soil, with resulting nutrient requirements and fertilization decisions in a site-specific manner.

1.4.4 Disease management

Illnesses not only harm plants and animals but also have a significant impact on market availability and agricultural production. To mitigate the negative effects of crop and livestock diseases, maximize yields, and prevent losses, disease management using emerging technologies is considered a practical approach [31]. Disease management applications encompass: 1) *Crop health monitoring:* These applications enable regular and continuous monitoring of crop health, providing farmers with valuable insights to increase productivity on a large scale with minimal effort. 2) *Livestock health monitoring:* Such applications are employed for the supervision of livestock health, accomplished through regular monitoring and tracking of the animals' diet and daily behaviors. This proactive approach aids in maintaining the well-being of livestock and optimizing their productivity. 3) *disease prediction:* this type of application is being used to forecast the arrival of diseases in crops and livestock. This includes predicting the occurrence of disease outbreaks. 4) *behaviors monitoring:* the backbone of these applications is the utilization of wearable sensors to tag livestock and track their daily behavior in order to identify any dangerous anomalies such as those caused by injuries or sickness. 5) *disease detection:* in time and accurate detection/diagnosis of diseases are critical for successful agricultural production. Using sensors, AI, and other technologies, smart farming makes this process automatic and cost-effective. 6) *disease prevention:* these implementations are based on intelligent

controlled agricultural environments designed to prevent and control diseases that may affect agricultural products. 7) *heart rate and blood pressure monitoring:* the use of technology allows constant supervision and analysis of heart rate of livestock and their blood pressure, factors that are fundamental in determining animal stress and movement. 8) *disease classification:* these applications generally involve ML-based classification techniques with extensive data for both normal and disease classes to accurately spot anomalies.

1.4.5 Smart harvesting

Reduced work effort, time, and cost are some of the benefits of using technology in agricultural operations such as harvesting. Smart harvesting applications include 1) *objects detection:* these applications are focused on image processing techniques, whereby instances of a specific class of objects are determined in images or videos. 2) *robotics arms:* the harvest cycle is a major area of application for robotics in agriculture. Including autonomous tractors and robotic picking arms. 3) *motion control:* these applications aim to ensure that harvesting robots are always able to receive direct commands from the operator to control their movements, thus making the harvesting task more efficient. 4) *fruit detection and classification:* one of the critical necessities of a fruit harvesting system is recognizing fruits on the trees. AI in general and ML in particular are heavily employed for such tasks. 5) *colors and shapes recognition:* these two markers (colors and shapes) are sometimes the primary indicators of the agricultural products' maturation. These applications tend to use ML techniques for integrating this knowledge into machines deployed in the agricultural field. 6) *obstacles detection:* colliding with a part of the greenhouse structure can result in damage to the building or the harvesting robot. Therefore, some kind of obstacle detection mechanism is used to prevent such accidents. 7) *optimal harvest date:* yield loss will result if harvesting is carried out either earlier or later, with both being undesirable. Therefore, applications exist to predict the optimal harvest date for each specific crop.

1.4.6 Supply chain management

Supply chain management involves efficiently handling the entire process of asset and service flow, spanning from raw materials to finished products. In the realm of smart agriculture, significant advancements are being made to revolutionize the agricultural supply chain by incorporating essential technologies. These innovations aim to create a seamless flow of supply chain information, connecting all the way from farm to fork [30, 28, 34].

Smart agriculture has introduced several applications for supply chain management, including: 1) *Products identification:* These applications utilize RFID tags to categorize, identify, and effectively manage the flow of products within industrial settings, allowing for efficient tracking and control. 2) *Traceability:* Blockchain offers a shared and replicated structure for data among network members, ensuring a transparent system for exchanging food supply chain and logistical data across supply networks. 3) *Food Safety and Quality Control:* Technology integration in the food supply chain leads to improved food quality and safety by closely monitoring food conditions throughout the supply chain and shar-

ing relevant data with supervisors and consumers. 4) *Agricultural Mobile Crowd Sensing:* This technique harnesses the power of mobile devices and sensing equipment, involving a diverse range of individuals who gather and share valuable agricultural information, optimizing data collection within the agricultural sector. 5) *Chain Risk Control:* Applications designed for chain risk control provide real-time risk identification at any stage of the food supply chain, ensuring significantly higher safety levels for both consumers and manufacturers.

1.5 Conclusion

Emerging technologies not only equip the agricultural sector with specialized tools and knowledge for better product delivery with reduced labor and time, and significant financial gains, but also provide a pathway to sustainability, which is crucial to current and future life on earth.

In this chapter, we presented a brief yet holistic overview of the smart agriculture domain, in which we provided a short introduction to the research field with a historical overview and real-life case studies. Moreover, we provided a clear look into the fundamental concepts that are forming it, including both architectural and technological aspects. Lastly, we present a detailed taxonomy of various smart agricultural applications.

It is obvious that considerable efforts are being made to improve the agricultural sector. Still, despite all the advantages that smart farming can provide, there will be a price to pay, and that will be the subject of the next chapter, in which we approach smart agriculture from another angle, specifically the security and privacy perspective.

Chapter 2

Smart agriculture security: aspects, threats, and defense strategies

2.1 Introduction

Intelligent agriculture should go beyond simply increasing food production. It must aims to promote automation and enhance decision-making capabilities throughout the entire agricultural spectrum. This objective is facilitated by intelligent data-driven management and control, which contributes to establishing a sustainable global agricultural ecosystem.

A recent Juniper Research report predicts significant growth in the agricultural technology market, expected to increase by 150% over a five-year period. By 2025, the market size is predicted to reach $22.5 billion. The report identifies sensors and supply chain management as the primary revenue generators, together accounting for more than 65% of the market size during the same period. This reflects the effectiveness of cost-efficient data collection methods and the integration of sophisticated and user-friendly analytical functionalities. The growth of the agricultural technology market and the focus on data-driven solutions underscore the potential for smart agriculture to revolutionize the industry, enabling more sustainable and productive practices across the agricultural landscape.

While the prospects of using technology in agriculture are promising, it also introduces substantial security risks. Consequently, the agricultural sector finds itself more vulnerable than ever, leading to heightened concerns among governments, industries, and research communities [37, 38, 39, 40, 41, 42, 43, 44]. The alarms are ringing, drawing attention to the urgent need for addressing the security challenges arising from the implementation of technology in agriculture. As smart agriculture continues to evolve and embrace innovative technologies, ensuring robust security measures becomes paramount to safeguarding critical agricultural assets, sensitive data, and overall operational integrity. It is vital for stakeholders to collaborate and proactively develop comprehensive security strategies to protect the agricultural ecosystem from potential cyber threats and unauthorized access.

Recently, the FBI's Cyber Division issued a Private Industry Notification (PIN) [45] emphasizing the alarming increase in ransomware attacks against the agri-food sector. These attacks have resulted in significant operational disruptions, economic damages,

and have had a destructive impact on the food supply chain. As a consequence of these attacks, there was a notable 65% surge in cyber insurance payouts from 2019 to 2020. While ransomware attacks are a prominent and immediate threat to smart agriculture, other cyberattacks often adopt a stealthier approach and can go undetected for extended periods. The issue at hand is not the cyber attacks themselves, but rather the inherent vulnerabilities arising from the widespread deployment of technology in agriculture. As smart agriculture embraces cutting-edge technologies, it must address the critical concern of cybersecurity. The adoption of technology introduces potential points of vulnerability that malicious actors can exploit. To safeguard against cyber threats and protect the integrity of the agricultural ecosystem, comprehensive cybersecurity measures are essential. Proactive efforts in identifying and addressing vulnerabilities, along with robust security protocols, are crucial to ensuring a secure and resilient future for smart agriculture.

This chapter provides a comprehensive view of smart agriculture from a security and privacy perspective. Beginning with an overview of the security aspects, followed by a taxonomy of the threat model and, finally, a look at the various defense strategies that can be used to protect against these threats.

2.2 Related studies

Several reviews have examined different aspects of cybersecurity challenges and critical issues in intelligent agriculture. For instance, Barreto et al. [37] and de Souza et al. [40] have focused on cybersecurity challenges encountered in smart agriculture. Gupta et al. [38] and Yazdinejad et al. [44] have explored privacy issues related to the distributed physical cyber environment of smart farming and precision agriculture, respectively, while also proposing potential countermeasures [39]. Moreover, specific technologies have been under scrutiny, such as network security in agricultural systems [41] and the security aspects of blockchain in the agricultural context [43, 42]. Some studies have approached the digital agriculture topic from a unique perspective, focusing on cyberbiosecurity [46]. For instance, Gupta *et al.* [38] examined smart farming security and privacy issues, threats and cyber attacks, and open research challenges. However, biosecurity was not sufficiently considered, and threat countermeasures are minimally addressed. Demestichas *et al.* [39] identified key ICT technologies in the agricultural sector along with their advantages, threats, and mitigation efforts. However, the paper did not discuss the ongoing challenges and future directions and focused less on Agri-biotech related threats. Zanella *et al.* [40] reviewed security issues in open-field smart agriculture, with a discussion on key challenges and future research directions. However, some threats related to digital agriculture, such as attacks on agri-biotech and the supply chain, are absent. Nikander *et al.* [41] made an identification of cybersecurity requirements from a networking perspective, supported by actual farm use cases. However, the threat model is limited to digital agriculture networks; other areas, such as application security, are absent.

Etemadi *et al.* [42] reviewed Blockchain technology applications for cyber risk management in the food supply chain. However, the scope of the paper is limited to blockchain technology; other areas, such as physical security, are not presented. Ferrag *et al.* [43] discussed privacy and security in green agriculture, along with privacy-oriented blockchain

solutions. However, the focus was on specific blockchain-based mitigation strategies, with less emphasis on other solutions. Yazdinejad *et al.* [44] classified safety threats in smart and precision agriculture, along with directions for future research. However, the study does not offer explicit strategies for dealing with the mentioned threats and problems. Drape *et al.* [46] gave a digital agricultural Cyberbiosecurity assessment and recommendations for improvement. However, the paper did not provide a thorough and technically informative threat model. Also, mitigation strategies are limited.

2.3 Security aspects

The first step in properly exploring the threats surrounding the smart agriculture ecosystem and ensuring that we are able to effectively neutralize them is to comprehend the security aspects associated with this domain.

Indeed, smart agriculture security requires a comprehensive approach that incorporates robust cybersecurity and biosecurity policies, as well as multiple critical control points. Additionally, the impact of human behavior on the overall security landscape must be considered [47]. This interdisciplinary research area, known as "cyberbiosecurity," has emerged as a recently introduced concept and an expanding domain. It is dedicated to safeguarding data, operations, and infrastructure within the life sciences and bio-economy sectors, while integrating cybersecurity, cyber-physical security, and biosecurity [46].

The goal of the cyberbiosecurity concept is to highlight and address potential vulnerabilities that may arise within or at the convergence of cyber-physical-bio implementations in smart agriculture. It aims to develop solutions to protect and secure these interconnected dimensions. Each dimension of cyberbiosecurity in smart agriculture is presented below, as represented in Fig. 2.1.

2.3.1 Cyber security

The fundamental elements of intelligent agriculture encompass programmable and intelligent devices that collaborate with one another, which includes Agri-IoT frameworks, mobile applications, and databases. This collaborative approach minimizes the necessity for extensive human intervention. The functionality embedded in these components is typically implemented through software, granting access to various resources such as sensitive data, network keys, logs, and more.

Cybersecurity focuses on safeguarding such vital technological systems across the virtual space from cyber-attacks originating from malicious actors. The constituent elements of intelligent agriculture face a wide range of cyberattacks, covering everything from micro-scale agricultural systems to large-scale, sophisticated supply chain management infrastructures, and all the way to agroterrorism [44].

The vulnerabilities inherited from the technologies widely employed across the agricultural sector should be accurately and comprehensively identified since there are no permitted weaknesses, where the slightest technological vulnerability can be abused to cause massive damage. The cyberspace scope that should be protected involves, but is

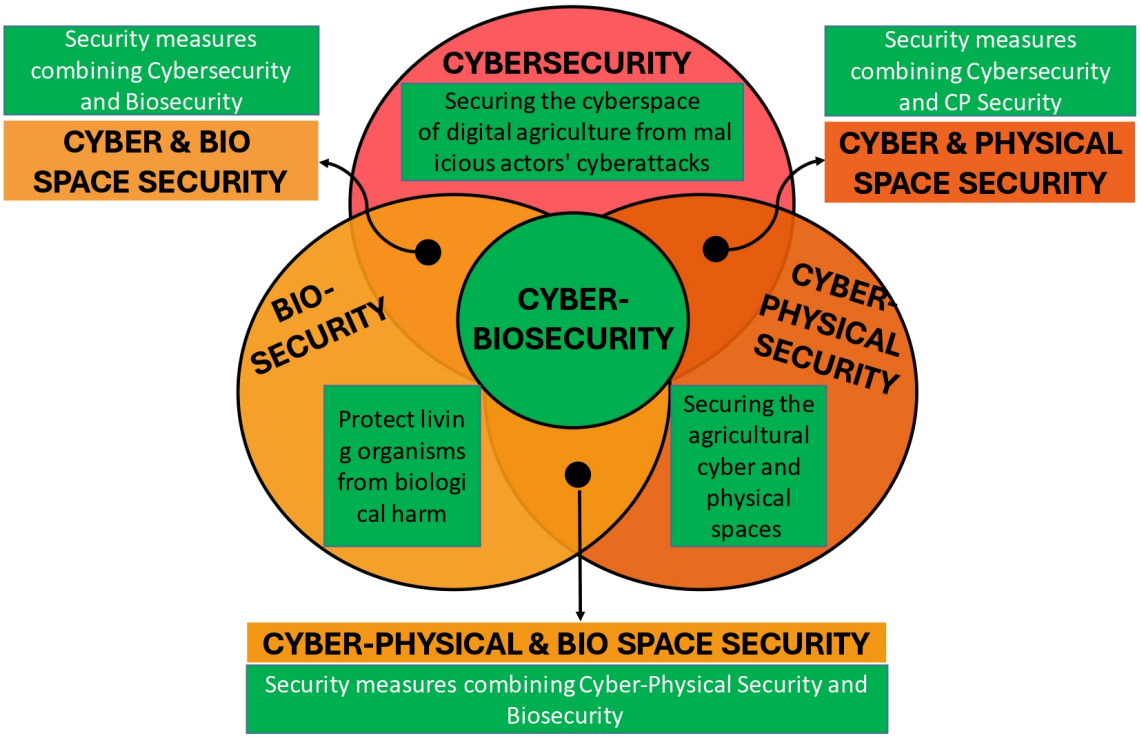


Figure 2.1: Cyberbiosecurity in smart agriculture

not necessarily limited to, the following [47]:

- Specialized software solutions tailored for specific agricultural operations (Apps, APIs, Add-ons, etc.) .
- Lightweight networking protocols, particularly those designed for IoT with weak cryptography.
- Independent analytical solutions utilized for diverse forecasting operations.
- Ensuring the security of data, communications, and users encompassing clients' data, system logs, and users' anonymity.

2.3.2 Security of cyber-physical systems

Indeed, while it is crucial to prioritize the protection of cyber assets in smart agriculture, it is essential to recognize that the scope of potential targets extends beyond just the digital domain. Cyber-Physical Systems (CPSs), such as monitoring drones, smart tractors, and irrigation facilities, are extensively used in modern agriculture [4]. This integration of physical and digital components makes the agricultural space an attractive target for malicious entities.

Even though Cyber-Physical Systems (CPSs) play a primary role in driving significant opportunities in intelligent agriculture, enabling real-time operations, automated tasks,

autonomous agricultural machinery, and advanced industrial control systems. However, the diverse and widespread deployment of CPSs also exposes them to a broad spectrum of cyber-physical security attacks, including malicious injections and hardware hijacking [48]. The cyber-physical space that requires protection in smart agriculture includes, among other components [47]:

- Software operating on dedicated hardware designed for agriculture, including mapping software used for precision agriculture and other specialized applications.
- Unauthorized Access to deployed equipment, which encompasses UAVs and UGVs, robots, and autonomous tractors. Securing access to these physical assets is crucial to prevent unauthorized control or manipulation by malicious entities.

2.3.3 Biosecurity

The Biosecurity space in smart agriculture is primarily focused on safeguarding dedicated parties, including private food industries, agricultural laboratories, and governments. However, it also involves protection against specific malicious actors who seek unauthorized access to valuable research findings or intend to cause harm to specific bio-organisms. These malicious actors may include counter-governments, espionage agents, industrial opponents, and other entities with harmful intentions.

The facilities involved in biosecurity in smart agriculture utilize special-purpose software and hardware products tailored to carry out their specific missions, which may include breeding techniques and genetic engineering. However, these facilities may also incorporate general-purpose technologies that possess inherent vulnerabilities, potentially making them susceptible to both cyber and physical infrastructure attacks. Such vulnerabilities pose a significant threat to biosecurity [49, 46]. The primary purpose of this space is to ensure the safety of living organisms from biological harm. The bio space that requires protection in smart agriculture encompasses, among other components [47]:

- Software running on dedicated Agi-biotech hardware.
- Bio-organisms at farms, laboratories, and throughout the entire supply chain.

2.4 Threat model

The rapid implementation and development of technologies in the smart agriculture sector are the primary cause of security threats. The fast-paced nature of these advancements leaves insufficient time to thoroughly assess their impact on security. While the reasons for this trend may vary, such as pursuing quick profits or tight project deadlines, the consequence remains consistent—a compromised eco-system [44, 39].

Unless appropriate security measures are in place to address these aspects, the digital agricultural industry becomes vulnerable to a wide variety of attacks capable of exploiting or damaging critical business organizations, assets and information systems. Digital farming attacks are classified according to their target.

Context	*Attacks*	*Examples of Current/Possible Impact*	*Targeted Principle*					*P*
			C	*I*	*A*	*Au*	*N*	
Data	Data Leakage/Theft	An example of such attacks is Intellectual Property (IP) theft, any data leakage or theft can breach any IP ideas such as crop models, plant breeders' rights and IoT-generated information [39].	✓	✗	✗	✓	✓	✓
	False Data Injection	Providing IoT-based platforms with fake telemetry data from IoT devices will definitely lead to faulty analytics and decisions, resulting in total disruption and loss of time and resources.	✗	✓	✗	✓	✓	✗
	Modification/ Fabrication	Spreading false information about disease outbreaks among a smart farm's crops or livestock in a falsified report [39], can result in significant financial and reputational damage.	✗	✓	✗	✓	✓	✓
Network	Eavesdropping	The intercepted information includes network addresses, employed protocol, open ports, and so on, which will assist the attacker in further attacks, such as network key extraction.	✓	✗	✗	✗	✗	✓
	Protocol Attacks	IoT-based protocols such as MQTT, ZigBee, and Bluetooth Low Energy are subject to various types of attacks, including brute force, sink-hole, and black-hole.	✓	✓	✓	✓	✓	✓
	Edge-Gateways Hijacking	Adversaries performs different types of network-based malicious actions such as sniffing, man-in-the-middle [45], re-routing [39], and malicious firmware/software updates delivery.	✓	✓	✓	✓	✓	✓
Software	Applications Attacks	A work presented in the *DEF CON 29* conference reported a supply chain vulnerability affecting the web APIs of major manufacturer John Deere enabling total control over machinery[1].	✓	✓	✓	✓	✓	✓
	Third-Party Attacks	Microsoft Exchange Servers were recently the targets with a 0-Day exploit, which caused the compromise of all the organization's emails that use such third-party technology[2].	✓	✓	✓	✓	✓	✓
	Malware Attacks	The PIN mentioned earlier, reported that a worldwide food company, suffered from a ransomware attack to obtain a $40M ransom, and lost several terabytes of sensitive data [46].	✓	✓	✓	✓	✓	✓
Service	Cloud Attacks	The Sensor-Cloud paradigm is susceptible to a wide range of attacks, including cloud-data theft [39], DoS/DDoS attacks [45], wrapping attacks [41], and man-in-the-cloud attacks.	✓	✓	✓	✓	✓	✓
	AI Attacks	Such attacks involve the introduction of intentional bias into ML data, and adversarial data poisoning, which focuses on injecting poisoned data into the training dataset to meet a harmful purpose [51].	✓	✓	✗	✓	✓	✓
	Blockchain Attacks	Blockchain technology has different vulnerabilities, including the majority (51%) vulnerability, double spending, transaction privacy leakage, and smart contract vulnerabilities [27].	✗	✓	✓	✗	✗	✗

Confidentiality (C); **I**ntegrity (I); **A**vailability (A); **Au**thentication(Au); **N**on-Repudiation (N); **P**rivacy (P).

Table 2.1: Cyber space threats in smart agriculture

2.4.1 Cyberspace threats

The targets of such attacks include data, networks, and systems of smart farming across the cyberspace. Tab. 2.1 [47] provides an overview of threat model classifications for cyberspace, along with the actual and/or potential impact of these threats, expressed in terms of casualties, targeted security principles and privacy issues. These types of attacks are further classified into four sub-classes [47]:

Data-related attacks

Data holds significant value in the agricultural sector, making it an appealing target for malicious actors. These attackers may focus on data at various stages of its life cycle, including the generation, transfer, preservation or treatment [44]. The attacks on data encompass:

- *Data leakage and/or theft*: Unauthorized access to data, whether unintentional or intentional [37].
- *False data injection*: Deliberate feeding of false data into smart agricultural systems [43].
- *Data modification and/or fabrication*:Targeting data integrity, where even documents that have been digitally signed can still be altered or falsified with exploits using techniques like XML Signature Wrapping [39].

Network-related attacks

Network technologies serve as the essential connection that binds the whole digital farming ecosystem together. Due to their significance, they become both a target for attacks and the mechanism for carrying them out. Attacks on networks in digital agriculture include [47]:

- *Eavesdropping*: Adversaries intercept private communications in real-time between components of the digital agriculture ecosystem, such as IoT devices, edge gateways, and drones, as part of their intelligence-gathering phase [44].
- *Protocol attacks*: Along with the mass deployment of IoT devices across the digital agriculture ecosystem, including smart farms, greenhouses and supply chain industry infrastructure, opponents are leveraging vulnerabilities in the communication protocols [41].
- *Edge-Gateways hijacking*: Attacking edge gateways enables malicious parties to gain control over agricultural network traffic or perform traffic eavesdropping.

Software-related attacks

These attacks occur when malicious individuals craft a piece of code to exploit a vulnerable agricultural system and use it in an unauthorized mode. It includes [47]:

- *Applications attacks:* Desktop, web, and mobile applications serve as user-friendly interfaces for accessing digital agriculture services. Despite their convenience, these applications are susceptible to various attacks, such as SQL injection, Cross-Site Scripting (XSS), and buffer overflow. These weaknesses pose significant risks to the security and integrity of the digital agriculture ecosystem.

- *Third-Party attacks:* The introduction of third-party services creates a potential avenue for proxy attacks to bypass existing application security defences, thereby enabling unauthorized access to private data owned by agri-businesses [39]. These proxy attacks can exploit vulnerabilities in the third-party services to gain access to sensitive information, posing a serious threat to the security and confidentiality of digital agriculture data.

- *Malware attacks:* Up until now, the types of attacks that have gained widespread attention in the digital agriculture sector are primarily of a public nature. These attacks involve adversaries installing malware in agricultural systems, granting them unauthorized control over these systems. As a result, these malicious actors can exploit vulnerabilities to compromise the functionality and security of digital agriculture operations. The proliferation of such attacks highlights the pressing need for robust cybersecurity measures to protect the agricultural sector from potential threats and ensure the integrity of its critical infrastructure and data.

Service level attacks

A wide range of diversified services are available to digital farmers via the Internet, which provides plenty of valuable tools for Agriculture 4.0. like enhanced transparency through the supply chain [42], agricultural object detection models, big data storage, smart contracts, content hosting, datasets, and powerful computing hardware [4]. Still, just like any technology, it's prone to cyberattacks. Attacks on service level include [47]:

- *Cloud attacks:* Integrating IoT-cloud in agriculture allows everywhere ubiquitous access to shared resources. As a consequence, agricultural data could be located in less secure regions [38].

- *AI attacks:* adversaries can exploit the rapid and fascinating advancements of AI in cyber attacks, a danger with the name *weaponized AI* [50]

- *Blockchain attacks:* although blockchain represents a successful solution for maintaining transparency and traceability in food supply chain management systems [4, 42, 38], it is not invincible against cyber attacks.

Context	*Attacks*	*Examples of Current/Possible Impact*	*Targeted Principle*					*P*
			C	*I*	*A*	*Au*	*N*	
Soft-side	Malicious Code Injection	Exploiting the cyber side of the deployed hardware and injecting malicious code, such as commands [39] and firmware updates [40].	✓	✓	✓	✓	✓	✓
	Misconfiguration Attacks	Where attackers use weaknesses in the configuration of the installed software [42].	✓	✓	✓	✓	✓	✓
	Weakening-based Attacks	Sleep deprivation involve sending queries to victim entities as frequently as necessary to keep them awake, in order to drain their batteries [41].	✗	✗	✓	✗	✗	✗
Hard-side	Malicious Hardware Injection	The work in [49] used a malicious USB device that acts as a generic human interface device, to infect the targeted system.	✓	✓	✓	✓	✓	✓
	Hardware Abuse	Optical deformation of cameras in autonomous devices [41], or destroying IoT sensors used for livestock/crop monitoring tasks.	✗	✗	✓	✗	✗	✓
Composite	Digital Twins Attacks	Any deviation caused by physical attacks on hardware components makes the mirror erroneous, and consequently, decisions and analysis get screwed up.	✓	✓	✗	✓	✓	✓
	Side-channel Attacks	include timing channel attacks, hardware glitching, power consumption, and electromagnetic leaks [39].	✓	✗	✗	✓	✗	✓

Confidentiality (C); **I**ntegrity (I); **A**vailability (A); **Au**thentication(Au); **N**on-Repudiation (N); **P**rivacy (P).

Table 2.2: Cyber-Physical space threats in smart agriculture

2.4.2 Cyber-physical space threats

The attacks falling under this category primarily target cyber-physical entities in the digital agriculture sector, including machinery, unmanned vehicles, sensors, and actuators. In Table 2.2 [47], the threats within the cyber-physical space are classified with their potential and/or present impact on targeted security principles and privacy issues. These attacks are further subdivided into three sub-classes, namely soft-side, hard-side, and composite attacks [47].

Soft-side attacks

These attacks are achieved by targeting the cyber-physical entities through their installed software/firmware. Soft-side attacks include:

- *Malicious code injection:* This kind of approach aims to overtake entities physically installed in the smart farming environment.
- *Misconfiguration attacks:* where adversaries aim to find or introduce weaknesses in device configurations.
- *Weakening-based attacks:* in which opponents attempt to undermine the physical objects implicated in farming activities.

Hard-side attacks

When the objective is to target cyber-physical agricultural objects through its hardware, thereby necessitating some level of physical access. Hard-side attacks include:

- *Malicious hardware injection:* such as including internal malicious removable devices or backdoors injection.
- *Hardware abuse:* Securing access to devices is critical since direct access to physical agricultural objects makes it much more difficult to regain control of the system [48].

Composite attacks

In these types of attacks, hackers exploit both physical hardware and related software in a composite manner, including:

- *Digital Twins attacks:* The digital twin (DT) enables digital simulation of cyber-physical systems, supplying a comprehensive range of internal data on the real system's components and their interconnections. This virtual counterpart captures real-time data and enables simulations, allowing for enhanced monitoring, analysis, and optimization of the physical system's performance and behavior. DTs offer valuable insights and facilitate better decision-making processes in various industries, including digital agriculture, by providing a comprehensive understanding of the interconnected components and their dynamics.
- *Side-channel attacks:* malicious actors may aim to gain knowledge about or influence the real-time functionalities of a specific system by closely monitoring or exploiting the side effects of the systems. These actions can be part of an attack strategy to gather sensitive information, manipulate system behavior, or disrupt normal function of the targeted system. The exploitation of side effects allows attackers to gain insights into the system's internal workings, potentially leading to unauthorized access or control over critical components.

Context	*Attacks*	*Examples of Current/Possible Impact*	*Targeted Principle*					*P*
			C	*I*	*A*	*Au*	*N*	
Cyber-Bio	Malicious Code Injection	The work in [50] presented an exploit embedded in a DNA molecule to compromise the runtime system.	✓	✓	✓	✓	✓	✓
	Bio-Data Attacks	Mirsky *et al.* [53] demonstrated the possibility of using deep learning to falsify CT scans, by either injecting or removing data into/from 3D medical imagery.	✓	✓	✗	✗	✗	✓
Biological	Bio-Organisms Abuse	Where adversaries aim to harm agricultural organisms with viruses, bacteria, fungi, insects, etc; or spreading infectious pathogens, influenza, and so on [47].	✗	✗	✓	✗	✗	✗

Confidentiality (C); **I**ntegrity (I); **A**vailability (A); **Au**thentication(Au); **N**on-Repudiation (N); **P**rivacy (P).

Table 2.3: Agri-Biotech space threats in smart agriculture

2.4.3 Agri-biotech space threats

Despite having a dedicated design and purpose, agricultural biotechnology takes advantage of certain cyberspace features and can consequently be targeted. Attacks in this

category aim to attack agricultural bio-entities or their associated biotechnologies. Tab. 2.3 [47] present agri-biotech space threat model classification, with the possible and/or present impact of such threats, in regards of the targeted security principles, and privacy issues. These attacks are further classified into two sub-classes, namely cyber-biological, and biological attacks [47].

Cyber-biological attacks

In such attacks, malicious actors tend to exploit vulnerabilities within agri-biotech systems. Furthermore, Agrobiological records are a vital asset, and any loss of integrity or confidentiality can lead to significant problems. Cyberbiological attacks include:

- *Malicious code injection:* where hackers aim at injecting malicious directives to interact with biological streamlines in the target's laboratory.
- *Bio-Data attacks:* can be used to falsify the agri-bio data of diseased livestock or corps, which is consumer-oriented, as well as to falsify this data and make healthy livestock or crop looks sick, or vice-versa.

Biological attacks

Also called bio-organisms abuse, such attacks are targeting bio-organisms, such as crops and livestock, an incident known as *agroterrorism*.

Context	*Attacks*	*Examples of Current/Possible Impact*	*Targeted Principle*					*P*
			C	*I*	*A*	*Au*	*N*	
Multi	DoS/DDoS	Which can target multiple parts of the ecosystem, such as by radio frequency jamming [38], or by botnets flooding agricultural applications with false requests [39].	✗	✗	✓	✓	✗	✗
	Social Engineering	Is the psychological manipulation of individuals to accomplish specific actions or leak sensitive data, such as spam [44], or using a combination of data gathering and access tokens faking.	✓	✓	✓	✓	✓	✓

Confidentiality (C); **I**ntegrity (I); **A**vailability (A); **Au**thentication(Au); **N**on-Repudiation (N); **P**rivacy (P).

Table 2.4: Polyglot threats in smart agriculture

2.4.4 Polyglot threats

Polyglot attacks can be considered as those attacks that can be executed in multiple contexts, or that require a mix of the previously mentioned attacks in order to succeed. Tab. 2.4 [47] present polyglot threats model classification. These attacks include:

- *DoS and DDoS attacks:* Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS) attacks are used where the attacker's goal is to make a certain agricultural service or functionality unavailable to legitimate parties.

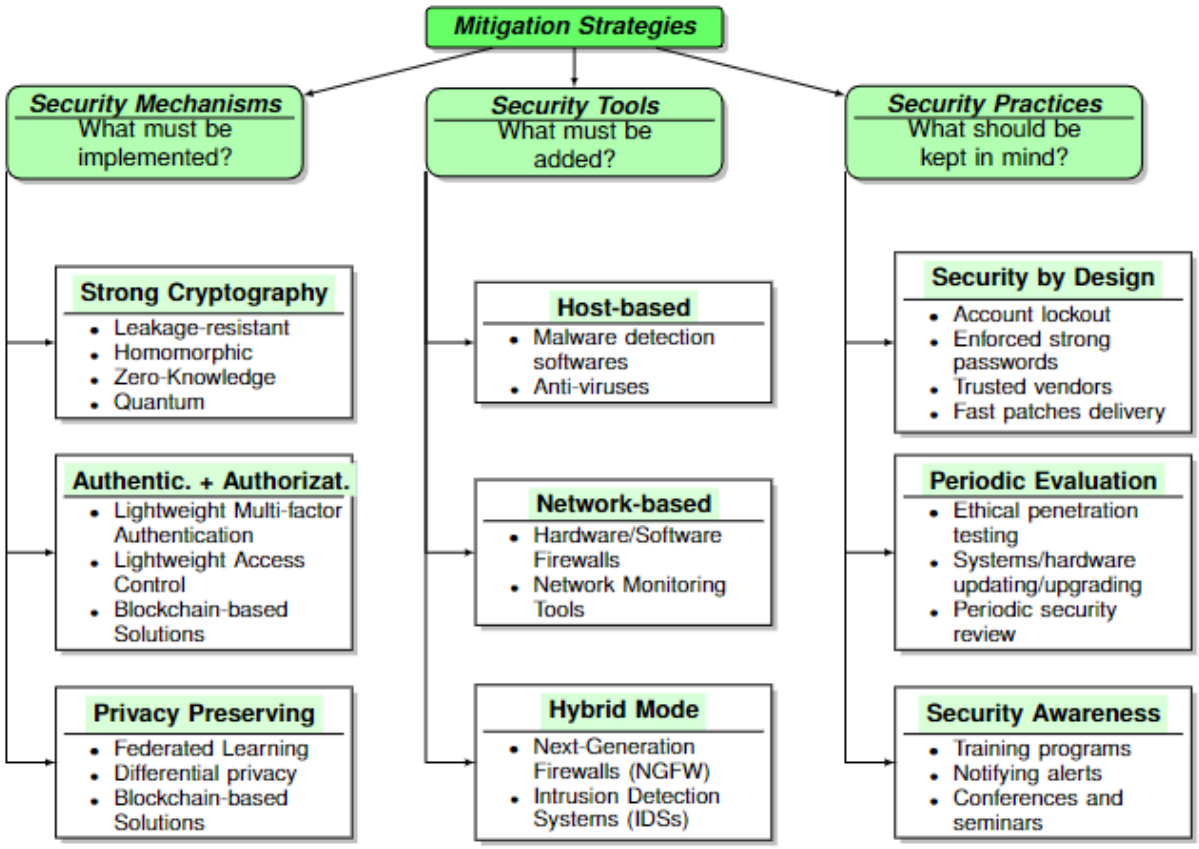


Figure 2.2: Security defense strategies for smart agriculture

- *Social engineering:* this type of attack is used to gain a certain level of access (e.g., by exploiting humans to give out their access codes) which cannot be gained otherwise, as with exploiting vulnerabilities in agricultural systems.

2.5 Security defense strategies

For their part, the bad actors have the privilege of picking and choosing among the previously mentioned threats and going after the low-hanging fruit whenever they want. On the other hand, the good guys rarely have any such luxuries and are responsible for keeping the smart agriculture ecosystem secure against as many threats as possible. The challenge is further complicated by the fact that not every threat is known (or disclosed publicly). For example, zero-day vulnerabilities require creativity and foresight to mitigate.

Despite the difficulty of the task, this does not necessarily indicate that it cannot be done. Researchers have devoted a great deal of time and effort to pushing back the boundaries of security and making them more difficult for adversaries to cross. In this section, we outline existing and potential security defense mechanisms for intelligent agriculture ecosystems, by classifying them into the following groups: mechanisms, tools, and practices, as illustrated in Fig 2.2 [47].

2.5.1 Security mechanisms

Security mechanisms consist of the techniques implemented in order to carry out specific services for security, resulting in the appropriate establishment of security in digital agricultural systems and their related data. Some of these encompass but are not limited to, the following [47]:

Strong cryptography

As noted before, networks within the intelligent agricultural sector are susceptible to a variety of threats. Ensuring that any information or communication remains confidential from everyone except the intended recipients, involves the enforcement of strong cryptography setups while respecting the hardware constraints of the deployed agricultural IoT devices.

Classical cryptography has limitations in protecting against side-channel attacks, such as those based on power usage and clock examination [51]. Leakage-resilient cryptosystems [52] aims to address this issue by designing cryptosystems that remain secure even if adversaries acquire limited intelligence about the internal state of the system. This approach is particularly relevant for outdoor implementations, where devices are distributed across vast farming areas.

To enhance privacy and security in digital agriculture, techniques such as homomorphic encryption and zero-knowledge encryption have been proposed [52]. Homomorphic encryption allows calculations to be performed on encrypted data without the need for decryption, while zero-knowledge encryption only demonstrate the correctness of specific information to the prover, without revealing any other details. Alongside identity-based encryption [52], these techniques offer robust solutions for safeguarding agricultural networks and the data stored in the cloud.

In the long run, quantum cryptography is being explored as a promising security solution for digital agriculture. By leveraging the principles of quantum mechanics, quantum cryptography aims to provide enhanced security and protection against advanced cryptographic attacks. Implementing these cryptographic techniques can significantly contribute to securing digital agriculture systems and ensuring the confidentiality and integrity of agricultural data.

Authorization and Authentication

In smart agriculture, ensuring the verification of user identities and controlling their access to specific parts of the system is crucial for maintaining security. To remedy this, lightweight approaches such as multi-factor authentication, label-based access control and message based authentication were suggested for securing hardware-constrained digital agricultural devices [43, 44]. Additionally, blockchain was proposed to enhance the reliability and integrity of these security techniques [53]. By implementing lightweight authentication methods, digital farming devices can effectively verify the identities of users and grant them appropriate access privileges, even with limited computational resources. These techniques are well-suited for the resource-constrained nature of agricultural de-

vices, ensuring that only authorized users can interact with the system. Moreover, the integration of blockchain technology further enhances the security and trustworthiness of these authentication mechanisms. Blockchain provides a decentralized and tamper-resistant ledger, ensuring the immutability and transparency of access control records. This makes it harder for malicious actors to tamper with user identities or access permissions, thus bolstering the overall security of smart agriculture systems.

Overall, these combined efforts in authentication and blockchain technology play a pivotal role in establishing a robust security framework for smart agriculture, safeguarding sensitive data, and protecting against unauthorized access and cyber threats.

Privacy Preserving

As previously mentioned, a significant threat to digital agriculture is data attacks. To tackle the data-centric learning challenge in smart agriculture, federated learning (FL) emerges as a key solution, letting peers share learning seamlessly without the need to exchange private data [54]. To further enhance data privacy, differential privacy can be integrated with FL, even it may slightly reduce the model's performance. Additionally, impose the use of certificates signed digitally for specific ecosystem components, such as servers, machines, and devices, can bolster data security. In the realm of digital agriculture, successful privacy preservation solutions based on blockchain technology have been implemented. These solutions encompass distributed key management, reputation, trust, and Software-Defined Networking (SDN) solutions [43].

2.5.2 Security tools

Such tools are stand-alone software or hardware products, with a particular security objective. These tools can be classified into 2 different modes, specifically: the specific deployment mode and the hybrid mode [47].

Specific-Deployment Mode

In the complex ecosystem of smart agriculture, security measures must be implemented at various levels of the system. The specific deployment mode refers to aligning the security objective of a product with its deployment mode. *Network-based tools* are security products designed to ensure surveillance, shielding, and rapid intervention in case of unauthorized network violations. Examples of such tools include hardware/software firewalls. Additionally, the implementation of pseudo-random frequency hopping is beneficial in the agricultural field as it helps turn aside interference and restrict eavesdropping [41]. *Host-based tools*, on the other hand, are security products that focus on safeguarding data, processes, and internal systems. These tools encompass malicious software detection software, anti-viruses, and software isolation techniques [38, 44].

Hybrid Mode

In certain areas of the ecosystem, cross-context security becomes crucial, especially in the case of malware attacks that can target both networks and hosts. Next-Generation Firewalls (NGFWs) have emerged with advanced capabilities that extend beyond traditional firewalls [41]. These enhancements include in-depth packet examination, application supervision and auditing, embedded intrusion prevention plus cloud-based threat intelligence, as described by Cisco.

Intrusion Detection Systems (IDS) play an important role in hybrid security by utilizing ML to distinguish normal vs. malicious behavior across both hosts and networks. Additionally, the implementation of honeypots is vital to gaining a better understanding of threat operatives' patterns and tactics. Honeypots enable experts to gain insights into how specific vulnerabilities are exploited, facilitating informed decisions to protect resources in the future [47].

2.5.3 Security practices

Incorporating the most advanced security defence mechanisms does not guarantee automatic and complete security for the ecosystem. To achieve a robust security posture, it is essential to instill a security mindset throughout the entire workflow, strengthening all links in the system chain, particularly the weakest parts, which are the human components in this case [47]. Humans, as end-users and operators, play a crucial role in supporting the security of the system. Raising awareness, providing training, and promoting a culture of security consciousness are fundamental in mitigating risks and ensuring a more secure digital agriculture ecosystem. Security practices represent the policies that intelligent agriculture must enforce alongside the mechanisms and tools. These include:

Security by Design

In smart agriculture, a proactive approach to security is essential, ensuring that equipment and software used are inherently secure. For instance, agricultural applications should enforce a robust password policy for their users, minimizing the risk of rainbow table attacks in case of database breaches. Additionally, implementing account lockout capabilities after a given number of login errors can effectively prevent brute forcing logins [39]. Moreover, it is crucial for agribusinesses to collaborate only with vendors who promptly release patches for identified vulnerabilities within their products. This ensures that potential security weaknesses are addressed in a timely manner, reducing the risk of exploitation. Another critical aspect is guaranteeing API security by providing authentication for programs or users invoking the API. This additional layer of security helps protect against unauthorized access and potential data breaches [47].

By adopting a security by design approach and integrating these measures, the digital agriculture sector can enhance its overall security posture and minimize potential vulnerabilities.

Periodic evaluation

Regular and informed evaluation of smart agribusiness security is critical. For instance, it is recommended that credible penetration testing professionals or Red Blue teams be involved. Similarly, frequent upgrades for software and hardware are of critical value, considering the rapidly changing IT environment. And although all of these requirements result in additional costs for the agricultural sector, they will eventually pay off [47].

Security Awareness

Promoting proper security awareness is vital in empowering farmers, staff, and all agricultural workers with essential knowledge about online security and the security issues associated with their activities [44]. By raising awareness, individuals become better equipped to identify potential attacks they may encounter. Providing comprehensive training to staff in the agricultural sector enables them to develop the necessary skills to anticipate, respond, and recover from various types of attacks, including social engineering and ransomware attacks. Moreover, the use of hacking simulators can be beneficial in smart agribusinesses. These simulators help businesses gain a better understanding of their security posture and evaluate the effectiveness of their existing cybersecurity measures and personnel [47].

By fostering security awareness and continuously improving the knowledge and skills of agricultural workers, the sector can enhance its overall cybersecurity resilience and better protect against potential threats.

2.6 Smart agriculture security challenges

To conclude this chapter, Tab 2.5 [47] presents some potential research challenges and improvements for smart agriculture safety. They are also highlighted and summarized in the following recommendations [47]:

- ***Defense in depth and zero trust:*** The intelligent agriculture concept must be established with two key concepts in mind, namely, *defense in depth* and *zero trust*. Defense in depth is about applying a multitude of security barriers throughout the entire system. The aforementioned and other potential defensive strategies must be combined, where possible, in order to reinforce the ecosystem's security. The zero-trust aspect implies that there is absolutely nothing that can be trusted by the ecosystem, inside or outside its walls, but rather audit any effort in interfacing with its underlying infrastructures.

- ***Cyberbiosecurity training and collaborations:*** Collaborations among various experts, academics, and governments are vital to properly manifest and prioritize cyberbiosecurity in intelligent agriculture. training courses, workshops, and conferences help stakeholders in the agricultural ecosystem remain cautious and ready to deal with different security incidents.

Open Challenges	*Possible Solutions*	*Future Directions*
- Built-in security throughout the entire digital agriculture ecosystem, from sensors to complex systems - Balancing trade-offs between security and privacy when designing smart agricultural systems - Privacy concerns related to technology in digital agricultural systems	- The implementation of multiple security layers - Jointly collaborate a mixture of mitigation strategies - Apply security patches to identified vulnerable parts of the system - Perform forensic analysis to investigate which components have been adversely affected. - Using privacy-preserving techniques when possible such as FL, DP, and anonymity based-encryption schemes.	- Enforcing defense in depth strategies - The implementation zero-trust philosophy in already existing systems - Identifying 0-day, unpublished, or vendor-unaware vulnerabilities affecting digital agricultural system - Enhance logging capabilities using low-cost (soft, hard, and cost) tamper-proof technologies and discover ways to add them into legacy already running systems.
- Work towards strategically and legally aligning digital agricultural applications and smart infrastructure with a foundation of standards and assurances tailored to agribusiness strategy, security regulations, and targets. - Specific-purpose digital agriculture resources (datasets, architectures)	- Improving technical standards and security assurance through mutual acceptance by the known digital farming stakeholders. - Developing security-related datasets for assessing AI-based security solutions - Unifying a deployment environment for various types of digital agricultural systems	- Ways to reach an accord on standards and security insurance policies, with a view to achieving a genuine win-win situation, on the one hand, and on the other hand, with a view to benefiting the entire sector. - Choosing the right types of equipment (sensors, actuators, machinery, etc), and protocols used for the data generation process.
- Raising awareness among the various members of the digital farming community - Benchmark agricultural industry-wide compliance with cybersecurity	- Education through lectures, workshops, and discussions. - Imitating real-world incidents and attacks employing the advanced techniques of rogue threat actors.	- Making advanced threats known to entities that are not technologically savvy. - Encouraging employers, staff, and other parties to participate in the training programs
- Environmental conditions and issues in certain agricultural areas - Impact on deployed equipment and networks	- Use of special equipment capable of withstanding difficult environments - The transfer to Networking Over White Spaces and satellite-based telecommunications	- Engineering farm equipment that is both robust and fairly affordable for all agribusiness owners operating in harsh conditions

Table 2.5: Smart agriculture security: challenges, possible solutions, and future directions

- ***Insurance and standardization:*** While cyber insurance covers a broad array of cyber risks, agricultural cyber insurance policies lag behind in providing coverage for cyber incidents [38]. Standardization is another critically significant, but still missing, piece of security in today's digital agriculture ecosystem, in which governments, industries, and businesses establish standards outlining ways to regulate the implementation of a specific security service or utility based on agreement between all parties involved in the agricultural sector.

- ***Specific-purpose AI resources:*** In intelligent agriculture, similarly to any other industrial sector, AI capabilities provide solutions for agricultural applications like intelligent monitoring and task optimization. However, specific agricultural datasets remain relatively scarce. Furthermore, specifically tailored IDS datasets for agricultural applications are scarce, making use of general-purpose public security datasets potentially inappropriate given the unique characteristics of agricultural data and operations.

- ***Environmental conditions options and associated impact:*** The climate change, in-field deployments, and related issues have to be taken into account in the design, development, implementation, and IoT equipment deployment, USs, and farm machinery with the minimum costs in order to facilitate the shift towards digital from traditional approaches. Similarly, harsh conditions could introduce connectivity issues; Networking Over White Spaces is one potential solution, an approach used in Microsoft's FarmBeats project[1].

2.7 Conclusion

Making the smart agriculture domain secure is particularly challenging considering the expected enhancement of its defense gateways across at least 3 directions, namely cybersecurity, cyber-physical systems security, and biosecurity - with this intersection designated by the term *cyberbiosecurity* [46]. Furthermore, given the rapid adoption of emerging technologies in the agricultural sector, whereby dedicated agricultural machinery and data, business decisions, and the circumstances of agricultural fields are involved, cybersecurity may not always be a priority. Consequently, smart agriculture nowadays seems to be in its infancy from a security standpoint, requiring further research before safe adoption can be contemplated.

In this chapter, a security perspective on smart agriculture has been explored. First, primary security aspects related to intelligent agriculture have been highlighted. Next, existing and potential threats confronting the agricultural ecosystems were discussed. Then, current and potential defense strategies were detailed. Finally, some challenging security concerns were discussed.

In the next chapter, a particular defense strategy will be thoroughly examined, namely the Intrusion Detection Systems (IDS). The concept will be covered from different standpoints, including its background, constituent parts, proposed works, and the different learning approaches.

[1] https://www.microsoft.com/en-us/research/project/farmbeats-iot-agriculture/

Chapter 3

smart agriculture and intrusion detection

3.1 Introduction

The food industry has undergone a transformation from isolated and independent operations to highly interconnected, interdependent, and integrated processes aimed at enhancing overall efficiency. This shift towards connectivity and integration has been made possible by the adoption of emerging technologies, especially in the smart industrial sector, commonly referred to as Industry 4.0. The concepts and advancements discussed in the first chapter highlight the potential for replicating this technological embrace in the agricultural domain. By incorporating similar smart technologies, the agricultural sector can leverage enhanced efficiency, productivity, and sustainability, leading to further advancements in the industry.

The smart food chain system is rapidly evolving, becoming highly efficient and sophisticated, but this progress also brings increased exposure to potential risks. Advancing the agricultural sector is not solely dependent on successful technology implementation. It is equally critical to prioritize and guarantee the privacy and security of the sector throughout its development. As discussed in the previous chapter, the agricultural sector faces a wide range of threats that can have significant negative impacts on food security, food supply chain performance, and agricultural productivity. Therefore, safeguarding the sector against these threats becomes paramount to achieving sustainable and resilient agricultural practices. A comprehensive approach to cybersecurity and privacy protection is essential to eliminate risks and guarantee the smooth functioning of the smart food chain system, thereby supporting the continued growth and success of the agricultural sector.

Different systems have been proposed for protecting smart agriculture from cyberattacks [55], including encryption techniques, authentication, authorization, and Blockchain [43]. In this chapter, the focus will be on a specific defense strategy, namely the intrusion detection system (IDS). First, a concise overview to the concept of IDS is provided by outlining its definition, its different types, and its fundamental design stages. Then, a discussion about the specific opportunities that these systems can offer for the security of

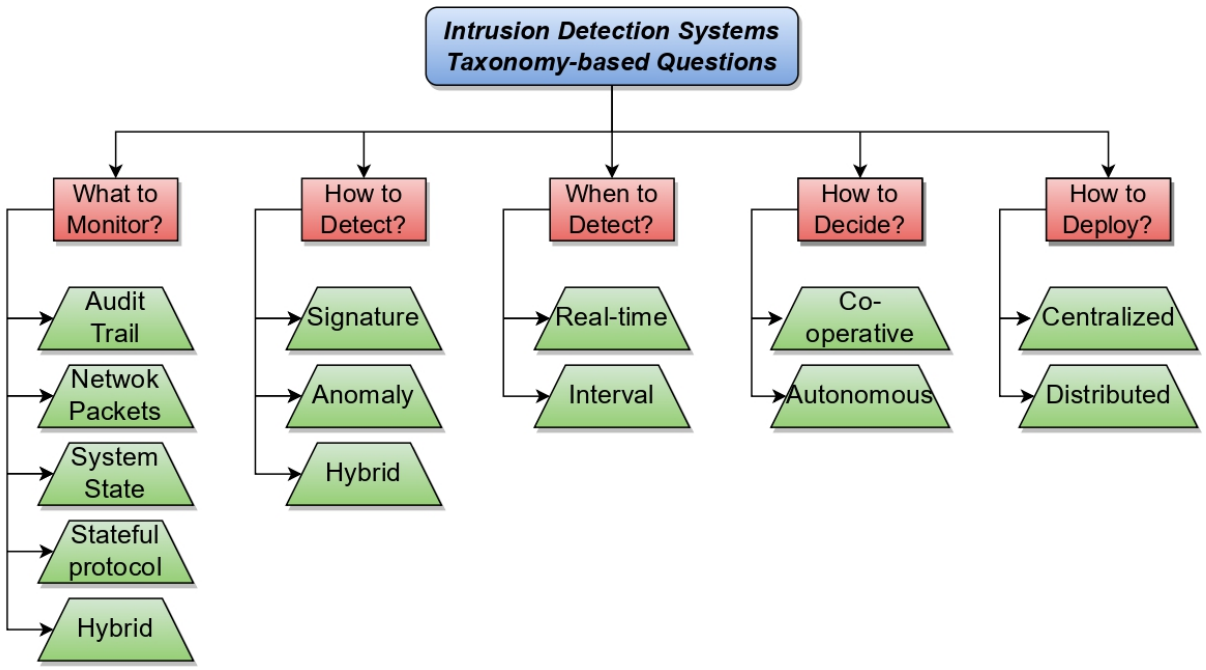


Figure 3.1: Intrusion detection systems taxonomy

smart agriculture is presented. Next, available and possible IDS-based security solutions for smart agriculture are discussed from a technological perspective. Lastly, the focus will be on anomaly-based intrusion detection with a specific learning approach, namely the Federated Learning approach, wherein a concise introduction to the topic is provided, followed by a brief overview of distributed intrusion detection, and at the end, we provide a review of selected recent works.

3.2 Intrusion detection systems

In this section, we describe the notion of intrusion detection, beginning with a brief background, followed by its definition and a rough taxonomy, and finally the basic phases of its design.

3.2.1 Where it all started

It is commonly believed that James P. Anderson's paper in 1972 [56], from the United State Air Force (USAF), established the foundation for what would later be called network intrusion detection. In which the author highlighted the observation of having the USAF "*become increasingly aware of computer security problems. This problem was felt virtually in every aspect of USAF operations and administration.*" At the time, the challenge was to determine how to the USAF could effectively provide shared user resources to access its systems, which had different levels of roles that could be used by various types of users at different levels of security authorization.

Eight years later, in 1980, Anderson presented a paper in which he introduced novel strategies for enhancing the auditing and monitoring of cyber security [57]. The concept of

automated intrusion detection is commonly credited to his research paper entitled "*How to use accounting audit files to detect unauthorized access*". Since then, and precisely since the 1990s, the domain of intrusions, and more specifically of network intrusion detection, has emerged as a significant area of research.

3.2.2 Definition and Taxonomy

A common rule of thumb in cybersecurity is that "*prevention is ideal, but detection is a must*" [58]. And this is the main objective of intrusion detection systems (IDSs), where the purpose is to continuously observe events in a computer system or network and then evaluate them for evidence of intrusion [59].

According to the NIST, the National Institute of Standards and Technology, an IDS can be defined as "*the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices*" [60]. In other words, an IDS can be viewed as an automatic process for detecting malicious incidents.

IDSs can be classified based on numerous differing characteristics [59, 61]. For instance, in Fig. 3.1 we provide a taxonomy of IDSs based on five important questions:

Q1. What to monitor?

According to the monitored (or targeted) assets, such as networks, hosts, IoT devices, and applications, the IDS can be specialized in monitoring (or detecting attacks based on), the followings:

- *Audit trail:* IDS that monitor audit trails (or logs) are known as Host-based Intrusion Detection Systems (HIDS). Systems and applications activities are maintained in such logs for different purposes including debugging and security audits.
- *System state:* Another important task for HIDSs is the current system state, which includes kernel state, sensors alerts, notifications, and so on.
- *Network packets:* the IDS that monitor network packets are known as Network-based Intrusion Detection System (NIDS). The mission of NIDS is to discover malicious traffic on a given network. NIDS typically requires unrestricted access to the network for full traffic analysis.
- *Stateful protocol:* the IDS that can keep a record of the state and analyze it for application layer protocols, to check benign protocol activity against monitored events to identify any drift, are called Protocol-based IDS (PIDS).
- *Hybrid:* A combination of the previous types may exist for the same IDS. Also, other types also do exist for example Virtual Machine (VM)-based IDS.

Q2. How to detect?

With respect to the strategies adopted for detecting attacks, IDSs can be classified as follows:

- *Signature:* also known as misuse-based approach. In this detection approach, IDSs judge the activities of the system/network relative to a collection of known attack characteristics called signatures [59]. Although this approach has obvious advantages, including high certainty and speed of detection. However, the main drawback of this approach is that it does not detect zero-day attacks, since the IDS does not have the corresponding signature [61].
- *Anomaly:* In this detection approach, the IDS is typically trained by monitoring regular system/network activity patterns and using such a profile as a reference for normal behavior. Any deviation is treated as abnormal activity [59]. This approach provides a clear way to recognize zero-day attacks. However, the disadvantage of this approach is that, depending on the sensitivity of the IDS and of the training quality, it may wrongly consider regular activities as abnormal [61].
- *Hybrid:* A combination of the previous approaches may exist for the same IDS.

Q3. When to detect?

According to the analysis timing, IDSs can be classified as follows:

- *Real-time:* also called event-based IDS [61], these systems are designed to monitor networks/systems activities in real-time to detect potential threats.
- *Interval:* also known as polling IDSs [61]. Unlike real-time monitoring IDSs, these systems do not intercept performed activities to gather analysis input data but instead obtain this data periodically.

Q4. How to decide?

According to decision-making actions, IDSs can be classified as follows:

- *Co-operative:* in this approach, decisions are made collaboratively. For instance, in multi-agents IDS (with distributed architecture as we will see below), if an anomaly is detected by a node, or where existing evidence is insufficient, a cooperative arrangement is initiated to generate a global intrusion detection response with peer nodes.
- *Autonomous:* In this approach, the IDS make decisions autonomously, and the remaining peer nodes are not engaged in this decision-making process.

Q5. How to deploy?

According to their deployment architecture, IDSs can be classified as follows:

- *Centralized:* consists of a single centralized system that collects data and one (or possibly more) monitoring unit(s) that control the activity of the host or network concerned.
- *Distributed:* consists of multiple intrusion detection subsystems, known as nodes or agents, sharing data related to intrusion detection. This communication can be done in a decentralized manner or orchestrated by a centralized entity that aggregates input information from the agents and carries out tasks such as agent administration [61].

Technology	*Application*	*Without an IDS*	*With an IDS*
Internet of Things	Smart crop/livestock monitoring	- Exploiting IoT devices - Protocol-based attacks - Crop/livestock losses - Surveillance gaps -Sensor data obstruction	- Intelligent farms equipped with IDS ensures the security of IoT systems, by mitigating attacks such as DoS, RPL, and sinkhole. This helps to avoid agricultural production losses.
	Smart water management	- Malicious commands injection - Insufficient or over-irrigation - Water loss	IoT-based intelligent irrigation systems could be protected by IDS against false control injection, DDoS attacks, which prevent crop and soil losses.
	Disease management	- Falsification of health-related collected data - Sensible data theft - Controlling health-related devices	IDS can protect IoT-based healthcare appliance information found in livestock and crops, and mitigate cyber attacks such as data theft, and integrity issues.
	Agrochemicals applications	- Malicious commands injection in agrochemicals sprayers - Irregular applications of fertilizers and diverse pesticides - Crop and soil loss	IDS' attacks detection together with fast alarm generation features provide protection against malicious sprayer control and rule-based anomalies.
Fog / Cloud Computing	Cloud-based data storage	- Agricultural IoT platform cyber attacks such as SQL injection, XSS, DDoS - Critical data losses or falsification - Authentication attacks - Malware injection attacks	IDS-secured systems, whether network-based or host-based, could prevent cyber attacks on intelligent agriculture platforms hosted in the cloud, and prevent unauthorized access to information.
	Fog-based data processing	- Fog nodes traffic jamming - Delayed or malicious field decisions - Fog nodes de-authentication attacks - Financial and resource losses - Malicious fog devices	Feature selection as well as classification-based intrusion detection techniques, such as fuzzy techniques, neural networks, and genetic algorithms, are used for the protection of network security and the improvement of the Quality of Service (QoS).
Blockchain	Food supply chain traceability	- Alter the manufacturing process of agricultural production through the installation of malware - DDoS the consortium Blockchain	Merging the Blockchain-based supply management system with the IDS could provide a more secure way to ensure traceability of agricultural products and manufacturing processes.
	Food safety and quality control	- False data injection about agricultural products leading to commercial fraud - Exploit vulnerable Quality Control (QC) systems	Ensuring the integrity of data sources using IDS could mitigate against fraudulent market actions such as counterfeiting.
SDN / NFV	Agricultural IoT-based network management	- OpenFlow protocol attacks - DDoS or Hijack SDN-enabled switches or controllers - Delayed or interrupted autonomous agricultural field tasks under network failure	IDS collects statistical flow information from the OpenFlow SDN-enabled switches and evaluates traffic information through the extraction and combination of a set of features, to mitigate cyber-attacks.
Agricultural Robotics	UGV/UAV autonomous tasks	- Malicious commands injection - Network traffic jamming - Agricultural machinery and production damages - GPS spoofing attack - Compromised surveillance	Smart farms equipped with anomaly-based IDS creates a model of normal farm machinery behavior, which is continuously updated, using data from normal use, and then applying this model to detect any deviations from normal behavior.

Table 3.1: Advantages of using IDSs for smart agriculture security

3.2.3 Fundamental design stages

In-depth research has been carried out on developing intelligent intrusion detection techniques to enhance networks/systems security [62]. The principal phases of every IDS construction workflow consist of (1) data collection, (2) data preprocessing, and (3) training [55]. A short description of each phase is presented below.

Collecting Data

Collecting data is the first and most influential step in the IDS creation process. Since this particular asset is going to be the material course from which the IDS learns. The source and point in time of data collection are key elements in the design and implementation of an IDS [63]. In addition, the data collection technique must be trusted and efficient. Trusted means that data collection techniques must be fully aware and correctly identify and label both attacks and benign data types. The efficiency part is dedicated to resource-limited environments, such as IoT-based. For instance, The authors in [64] employed Zigbee Diagnostic Reports to ascertain that IDS data collection can be safely and efficiently performed in a resource-restricted Zigbee IoT environment.

Data Pre-processing

Having acquired the necessary information in the data collection stage, it is then processed to generate baseline characteristics, also known as features [63]. The goal of the feature selection, is to decrease the computational complexity by eliminating unnecessary features (such as those with high Correlation scores) while maintaining or enhancing the IDS's performance [65]. Also in this stage, the data is cleaned (e.g. removing NULL values), numerically transferred, and normalized.Data transfer is a process that involves expressing each input data record as a vector of real numbers. Therefore, each symbolic feature in the data set must first be converted to a numerical value [63]. Next comes the data normalization step, which involves scaling the value of each attribute to a well-proportioned range, thereby discarding the bias toward features with higher values in the dataset, potentially greatly increasing the accuracy of the classification algorithm [63].

Model training

After proper data preparation and the selection of the optimal subset of features, it is then used in the classifier training phase where, for example, different ML algorithms such as Deep Neural Network (DNN), Support Vector Machine (SVM), and Recurrent Neural Network (RNN) are involved. Some classifiers such as SVM only handle binary classification problems, where it can only differentiate between attack or normal classes. Other classifiers, including DNNs, can perform multi-class classification, i.e., they can also determine the type of attack that was performed. A combination of multiple binary classification classifiers can also be used to solve the multi-classification problem [63].

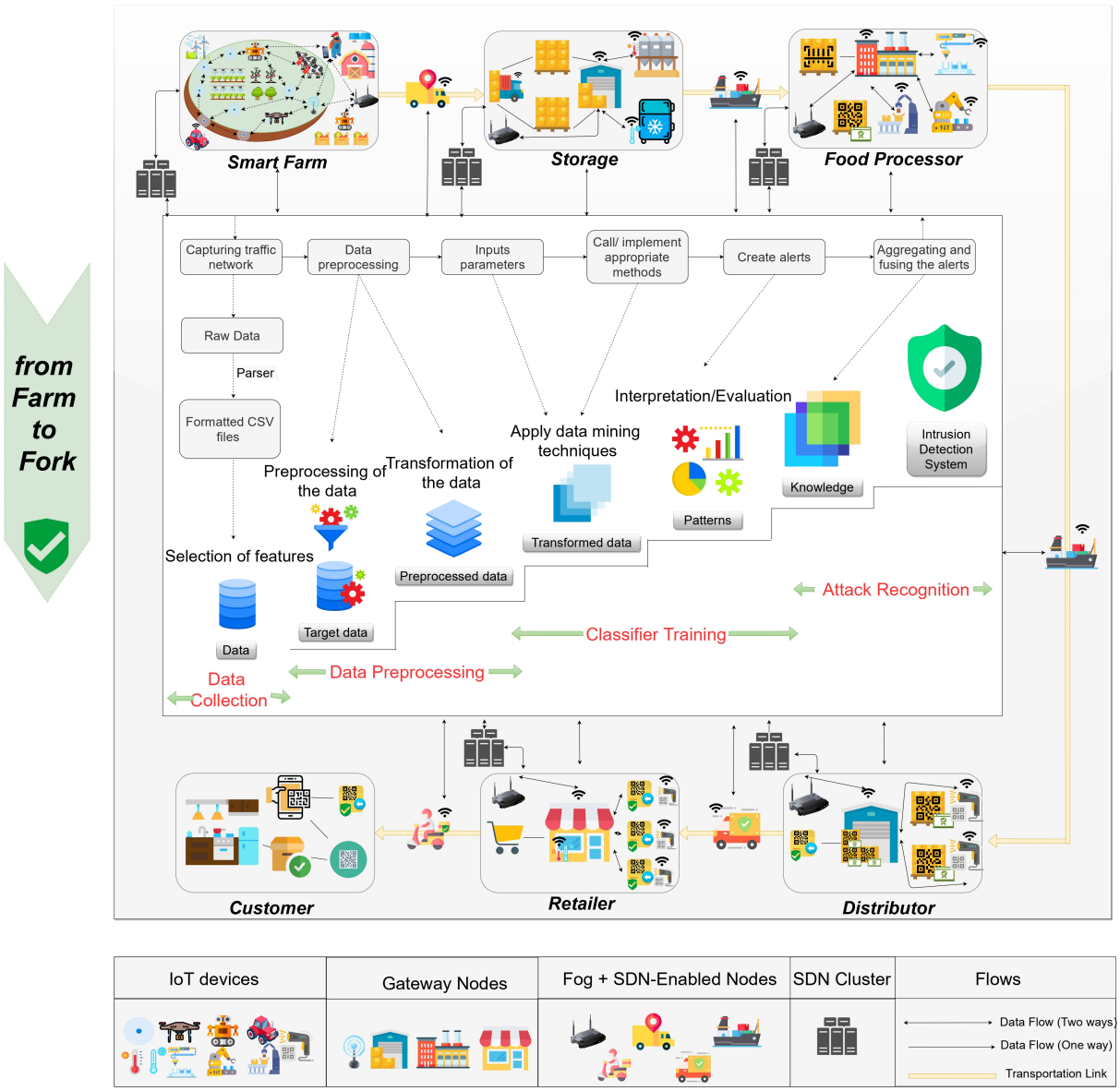


Figure 3.2: IDS in smart agriculture

3.3 IDSs for smart agriculture

Connectivity plays a crucial role in the smart agriculture domain, encompassing the management of various informational assets like sensors, the transportation of physical goods and services through the use of Internet of Vehicles (IoVs), and the integration of intangible assets such as software applications. As this level of control becomes increasingly pervasive and distributed throughout the agricultural ecosystem, securing all assets within the sector becomes a formidable challenge. The integration of smart agricultural systems undoubtedly enhances agricultural production. However, this increasing interconnections also exposes these systems to various security attacks. Such attacks have the potential to cause significant threat to agricultural services and applications, including smart monitoring, supply chain traceability, and autonomous tasks. To ensure the continued success and sustainability of smart agriculture, it is imperative to implement robust cybersecurity measures to safeguard against potential threats and protect the integrity and functionality

of these essential agricultural technologies and processes.

IDS implemented in the smart farming landscape is claimed to support real-time network packet analysis and enhanced traffic performance tracking, multi-layer network management with several protocol stacks, as well as robust security backup for a variety of technologies. In addition, it is expected to perform under strict limitations of restricted calculation conditions, high-speed performance, and high data volume handling [55]. Tab. 3.1 outlines some of the possible outcomes of having or not having IDS in place for smart agriculture [55].

3.3.1 Data sources

The most important part of ML and AI is of course the data. So without data, nothing can be trained and all the research and automation will be meaningless. When considering data, various factors must be considered, such as the original source, collection techniques, quantity, and, most importantly, its quality. For smart agriculture, there are two main sources of data [55]. The first can be obtained from each of the different components of the smart agriculture sector. The second source is the publicly available datasets that researchers provided to the research community and other sectors, which will serve as a reference and point of comparison for the research they propose.

Smart agriculture data sources

The integration of emerging technologies enables more intelligent food supply chains management, by bringing together various standalone data analysis models, historical data repositories, and multiple real-time data feeds [55]. With real-time data and automated data processing, the system can respond rapidly to dynamic circumstances in a timely manner.Thanks to innovative technologies, each agricultural component's functions are automatically merged into the food chain, thereby enabling a seamless flow from farm to fork, as illustrated in Fig. 3.2 [55]. However, it is important to note that different agricultural components may have distinct data sources that need to be incorporated. Ensuring the availability and smooth functioning of these diverse data sources across all systems is essential for effective and balanced operations. Some of the core data source assets for smart agriculture are [55]:

- *Smart farming infrastructures:* Designed by embedding cutting-edge technologies back into existing agricultural workflows, they include intelligent crop and livestock monitoring, smart water management, disease control, smart harvesting and more. It encompasses different types of sensors, actuators, drones/UGVs, smart farm equipment, and so on, focusing on connecting all objects in the IoT-based smart farm. All while monitoring, carrying out farming tasks and handling related data via the smart devices deployed.
- *Transportation services:* This segment is tasked with the transportation of agricultural products from the initial point in the supply chain to the final destination, which could be the customer's kitchen table or other distribution points. Transportation services in the agri-food chain make use of various smart technologies,

including GPS and Internet of Vehicles (IoV) communications. Through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) interfaces, vehicles can communicate with each other and with public networks. This connectivity allows for real-time data gathering and exchange of critical information, such as road conditions and updates on agricultural feedstocks. Such real-time information facilitates efficient and timely transportation, enhancing the overall performance of the agricultural supply chain.

- *Storage entities:* These entities play a crucial role in managing storage operations for agricultural products. For instance, cold storage facilities are equipped with advanced tracking equipment that continuously monitors the condition of stored agricultural products. Temperature and humidity sensors for instance are employed to gather real-time data on the stored products' status. Whenever any deviations or anomalies are detected, the system promptly notifies and alerts the management. This level of monitoring and alerting ensures that the stored agricultural products are kept in optimal conditions, preserving their quality and reducing the risk of spoilage or damage.

- *Food processors:* This category encompasses entities involved in the preparation of market-fresh food and the manufacturing of processed agricultural products. It embraces a varied set of manufacturers who use agricultural inputs or sub-assemblies from a variety of producers to develop their products. In the context of smart agriculture, food processors can leverage IoT-enabled devices to conduct a wide range of quality control operations. By integrating IoT-enabled devices, food processors can effectively monitor and manage various aspects of their production processes. This includes tracking production quantities, monitoring product temperatures, setting and regulating pressure settings, and implementing product tagging. These IoT-based quality control operations ensure the consistency and safety of the final products, enhancing the overall efficiency and reliability of the food processing industry.

- *Distributors:* Such services require entities to maintain large inventories of goods, purchasing them from producers and selling them to consumers. They respect the "when and where" policy of their customers, sending products when and where they want them.

- *Retailers:* These entities stock smaller quantities for the public selling. They also monitor preferences and inquiries from customers.

Publicly available datasets

A dataset is a collection of diverse data types structured as a digital object. The stored data can be made up of texts, images, numerical data points, videos, and so on. Since the quality of the datasets has a critical impact on ML models' accuracy, it is critical to choose the right dataset for a given task, e.g., classifying attacks and benign networks and/or systems activities. The cybersecurity research community had made great efforts to provide high-quality, high-volume datasets that can be used for training, evaluation,

and benchmarking of various types of IDS solutions. These include network flows, system logs, application protocols, IoT protocols, emerging technologies signatures, and a multitude of attack types. In what follows, we briefly present 4 recent datasets that are being used to engineer and assess IDS solutions for smart farming environments. We refer the reader to [55] and [66] for an in-detpth list of available datasets. Also, an in-depth description of these datasets will be provided in the next chapters.

The CSE-CIC-IDS2018 dataset was created as a cooperative initiative between the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE) to specifically train NIDSs [67]. Within the dataset, and in addition to the benign profiles, there exist seven separate attack categories, namely DoS, DDoS, web attacks, brute force, botnet, heartbleed, and infiltration attacks. Another dataset based on network flows is developed to train NIDS in more specialized environments, specifically, SDNs, is the InSDN dataset [68]. This dataset was made to address the shortcomings of existing datasets in the context of SDN-based network environments. The MQTTset dataset contains communications between various IoT devices to provide a simulation of an intelligent IoT environment [69]. For more network flows, protocols, volumes, and sophisticated attacks, the authors in [65], proposed the Edge-IIoTset, a realistic and extensive cybersecurity dataset of both IoT and Industrial IoT (IIoT) implementations. The proposed testbed is structured into seven layers along with 14 attacks related to IoT and IIoT protocols.

3.3.2 IDS-based security solutions

The most significant opportunity when using an IDS is the automation of the intrusion detection process, which is the main objective of its conception, as mentioned previously. A variety of IDS types have been designed, developed, implemented, and evaluated over the past two decades. In this section, we focus on looking at the proposed IDSs from the standpoint of technological perspective, given that technology is the principal suspect when dealing with cyber risks to the agricultural sector. Technology-targeted IDS-based solutions that are/can be implemented for the smart agricultural sector can be categorized as [55]: solutions for cloud computing-based applications, solutions for fog/edge computing-based implementations, solutions for SDN/NFV-enabled networks, solutions for IoT-based environments, solutions for industrial infrastructures, and solutions for smart grid systems. Tab. 3.2 [55] presents a summarized list of selected IDS, in which we provide the systems' network models, the detection and validation techniques used, together with their pros and cons.

For cloud computing-based applications

Cloud computing has emerged as a critical technology with numerous advantages, such as providing accessibility from any location and device, enabling rapid application deployment, and facilitating real-time business intelligence. However, as discussed in the previous chapter, this technology also introduces various security challenges within the smart agriculture sector if not properly secured. To address the security concerns, Gill et al. [70] developed a game theory-based Intrusion Detection System (IDS) named GTM-

System	*Network model*	*IDS model*	*Technique*	*Validation*	*Attacks*	*Pros (+)*	*Cons (-)*
Gill et al. [72]	Cloud environment divided into four layers: Attack Layer, Security Layer, Backend Layer, and End-user Layer.	Game theoretic model	- Signature-based - Anomaly-based - Honeypot	Simulation environment in MATLAB	Attacking the cloud servers with regular or sophisticated attacks	+ Reduce energy consumption by the defender system	- The threat model is limited
Rabbani et al. [73]	A cloud-based environment divided into different levels of sub-behaviors	A hybrid machine learning system	- A particle swarm optimization-based probabilistic neural network	Network intrusion detection dataset	Malicious behaviors such as Worms, Backdoors, Fuzzers, etc.	+ Good performance for malicious behavior detection and recognition	- Requires a comparison with deep learning techniques
Kushwah and Ranga [74]	A cloud infrastructure with a detector attached	A machine learning system	- Voting extreme learning machine	Network intrusion detection dataset	DDoS attacks	+ High accuracy of 99.18% with the NSL-KDD dataset and 92.11% with the ISCX dataset	- There are only three metrics used (i.e., accuracy, sensitivity, and specificity)
Aldribi et al. [75]	A cloud environment	A hypervisor-based IDS	- Online multivariate statistical change analysis	Network intrusion detection dataset	Input validation, authentication breach, backdoors, DoS, etc.	+ The overall detection rate= 96.23% and False positive rate=7.56%	- IoT data is not considered
Almogren [76]	Edge-of-Things computing	A hybrid machine learning system	- Malicious activity detection model	Network intrusion detection dataset	Nine attacks: worms, DoS, misuses, etc.	+ The deep belief network has the best overall performance as compared to the artificial neural network and support vector machine	- The network model is not defined
Abdulqadder et al. [77]	SDN/NFV enabled cloud of 5G networks	A multilayered intrusion detection and prevention system	- Artificial intelligence-based approach	Simulation environment in NS3	Host location hijacking, control plane saturation, DDoS attack	+ Efficient in terms of security between switches and controllers	- IoT data is not considered and the RoC curve is not reported
Zhou et al. [78]	Connected and automated vehicles	Distributed collaborative IDS	- Identify betray attacks in VANET	Simulation environment in Venis tool	Spoofing attacks, Black-Hole attacks, Gray-Hole attacks, and Denial-of-Service attacks	+ Can achieve a faster attack detection rate, lower false alarm rate, and higher detection rate	- The IoT data traffic is not considered
Tian et al. [79]	Distributed edge devices	A distributed deep learning system	- Analyzing URLs	Network intrusion detection dataset	SQL injection, XSS, and command injection	+ The accuracy fluctuates by approximately 0.955	- RoC curve is not reported
Murali and Jamalipour [80]	The routing protocol for low power and lossy networks (RPL) in IoT networks	A lightweight IDS	- Detect/identify Sybil attack	Simulation environment in Cooja under Contiki OS	Three types of Sybil attack	+ Average accuracy rate of the proposed IDS is 96.8%, 95.2%, and 94.8%, for type 1, type 2, type 3 attack, respectively	- RoC curve is not reported

Table 3.2: Summary of related IDSs for smart agriculture

CSec, specifically designed for cloud-based environments. The GTM-CSec system is made up of two main parts: co-operative and non-co-operative sets, and is supported by three detection techniques: signatures, anomalies and honeypots. Each of these techniques is embedded in four components: sensing, logic evaluation, computational evaluation and decision evaluation. Through performance evaluation using MATLAB with payoff functions and probabilities, the proposed GTM-CSec model demonstrated its ability to optimize the defense mechanism's power utilization and effectively protect against intrusions in cloud-based smart agriculture systems. This approach provides a promising step towards enhancing the security of cloud-based applications and services in the agricultural sector, ensuring the safety and privacy of sensitive agricultural data and operations.

Rabbani et al. [71] introduced a hybrid ML-based IDS that leverages user behavior pattern mining to detect malicious activities in cloud computing services. Their system employs probabilistic neural networks based on particle swarm optimization (PSO-PNN) to construct an automatically optimized network. Experimental results demonstrated that the PSO-PNN technique achieved high accuracy in detecting malicious behaviors within the cloud computing environment. Kushwah and Ranga [72] developed a cloud framework with an attached detector, utilizing three main components: the training database, the preprocessor, and the classifier. Their approach employs extreme voting Machine Learning to recognize DDoS attacks in the cloud computing framework. The research used two datasets (ISCX and NSL-KDD) for evaluation. The experimental findings showed high accuracies of 99.18%. Aldribi et al. [73] designed a hypervisor-based IDS that utilizes multivariate statistical analysis of online shifts to identify abnormal behavior in the cloud. The study evaluated and validated the proposed cloud intrusion detection model using the ISOT-CID dataset. Experimental evaluations demonstrated an overall detection rate of 96.23% and a false positive rate of 7.56% for the proposed system.

These studies exemplify the significant efforts and progress made in developing sophisticated IDS approaches for enhancing the security of cloud-based systems in the agricultural sector. The utilization of ML and statistical analysis techniques contributes to more robust and effective intrusion detection in cloud computing environments, safeguarding sensitive agricultural data and services from potential threats.

For fog/edge computing-based implementations

In contrast to transferring all IoT-generated data to the cloud or data storage facilities, fog/edge computing processes data at the network edge, where it is generated. However, security remains a crucial consideration for this paradigm, especially when edge computing is located indoors. To address security challenges, Tian et al. [74] proposed a distributed DL system based on CNN, to enhance security for multiple web applications in fog computing. The system leverages URL analysis to distinguish between malicious and benign requests. The research validated the edge IDS using HTTP Dataset CSIC 2010, FWAF, and HttpParams datasets. The experimental results demonstrated an accuracy fluctuating around 0.955. Additionally, Almogren [75] presented a malevolent activity recognition model for edge-of-things systems based on deep belief networks. The proposed design involved 3 key steps: data gathering, feature extraction, and classification. The edge-based intrusion detection framework was validated using the UNSW-NB15 dataset. Based on the

experimental results, the deep belief network outperformed the Support Vector Machine (SVM) and Artificial Neural Network (ANN) in regards of intrusion detection accuracy.

Haider et al. [76] proposed a novel intrusion detection design using a fuzzy Gaussian mixture. The method, named FGMC-HADS, follows these steps: 1) Learning data from the OS kernel, 2) Generating meaningful sequences through a joint characteristic construction module, and 3) Classifying segments as normal or anomalous using the Gaussian mixture. To assess its effectiveness, the FGMC-HADS method was evaluated on 3 datasets (ToN_IoT, KDD-98, and NGIDS-DS). The findings showcased the superiority of FGMC-HADS in terms of accuracy and fault detection compared to other Machine Learning approaches, such as SVM and k-nearest neighbor (KNN). Hosseini and Zade [77] proposed a hybrid IDS, which consists of 2 sections. The characteristic selection part involves a combination of a genetic algorithm and an SVM with multi-parent crossover and mutation. Attack detection involves an artificial neural network (ANN) coupling particle swarm optimization with hybrid gravitational search. The model was tested on the NSL-KDD dataset, and demonstrated a high detection accuracy of 99.3%.

For SDN/NFV-enabled networks

Both SDN and NFV concepts in intelligent agriculture provide a software-based management platform for monitoring industry standard network hardware. But the diverse nature of network attacks means that SDN/NFV-enabled agribusiness faces major security challenges that require effective solutions. Abdulqadder et al. [78] presented a multi-layered Deep Reinforcement Learning (DRL)-based Intrusion Detection and Prevention System (IDPS) for SDN/NFV-enabled advanced mobile networks in the cloud. The system comprises 5 layers, including switches, a smart controller, domain controllers, data gathering, and virtualization. Within the domain controller layer, a Shannon entropy function is used to classify packets, while self-organizing multiple maps are employed in the intelligent controller layer to detect DDoS attacks. Derhab et al. [79] proposed RSL-KNN, a framework for intrusion detection that integrates Blockchain and SDN. The framework utilizes two Machine Learning algorithms, Random Subspace Learning (RSL), and K-Nearest Neighbors (KNN), to defend against forged commands. The framework incorporates a blockchain-based integrity verification scheme to prevent misrouting attacks in SDN/NFV. Benchmarking showed that the RSL-KNN detection framework scored more than 91% and 96.70% in multi and binary classifications, respectively.

There are several significant disadvantages to signature-based detection, notably the exponential growth of the signature database over time and its inability to detect ”0-day” attacks. However, in some situations, the integration of both anomaly-based and signature-based can be rewarding. For example, Ngo et al. [80] designed an SDN-based system architecture for trusted handover devices employing signature and anomaly based IDSs. In particular, Two intrusion detection engines, called F-NIDS and F-ANIDS, have been combined in the proposed design. While the first engine uses Snort rules to classify attack packets, the second uses ML.

For UAV/UGV-enabled deployments

With the incorporation of 5G technology into the emerging concept of smart cities, the Internet of Drones (IoD) has arisen in the form of a new research area of drone-to-drone communication (D2D). Moreover, the deployment of multiple drones (i.e., a swarm of drones) cooperating to accomplish a specific goal of reduced operational overhead [55]. Despite this, such systems are vulnerable to cyber threats that can be exploited by an attacker to cause significant damage, e.g., by seizing their control, interfering with their operation, or theft of shipments they carry. As a result, ensuring the security of the system is increasingly critical, particularly for dynamic and decentralized D2D. As a result, the deployment of IDS for smart agriculture continues to be highly preferable. Sciancalepore et al. [81] presented a robust solution called PiNcH for detecting the presence of UAVs. This solution is designed to be highly efficient in the event of both severe packet loss and evasion attacks.

The increasing use of autonomous tractors in agriculture has brought about significant benefits but also raised concerns about security and privacy risks within the interconnected networks they operate in. To address these risks, IDS can play a crucial role. For instance, an IDS designed for in-vehicle networks can detect attacks like DoS and spoofing attacks, enhancing the security of autonomous tractors [82]. The high mobility and dynamic topology changes in autonomous tractors make them susceptible to various intrusions. To enable secure data collection, analytics, and inter-vehicle traceability, Zhou et al. [83] proposed a collaborative distributed IDS. This system utilized a cooperative communication method based on reputation, to ensure reliable and stable communication links between vehicles. The Venis tool was employed to simulate the traffic and network environment of vehicular communications. The experimental results demonstrated that the DCDIV system achieved a faster detection rate, reduced false alarm rate, and improved overall performance.

For IoT-based environments

In their efforts to enhance the security of IoT devices and detect cyberattacks within IoT environments in smart agriculture, Ferrag et al. [84] proposed three DL-based IDS. These models were evaluated using two classification types, binary and multiclass, and two novel real traffic datasets. The CNN-based IDS model, tested on CIC-DDoS2019 and TON_IoT datasets, demonstrated accuracy of 99.95% and 99.92% for binary and multiclass detection, respectively. Similarly, in the context of securing IoT-based environments, Ferrag et al. [85] introduced RDTIDS, a rule-based IDS based on the Decision Tree (DT) algorithm. During the training stage, tree classifiers are trained in a hierarchical model, and during the testing stage, the data is classified as either benign or malicious. RDTIDS is integrated into the fog computing layer within a three-tiered fog computing architecture. The experimental evaluations using both CICIDS2017 and Bot-IoT datasets, showed that RDTIDS achieved a maximum true negative rate of 98.855

Botnets represent a significant threat to IoT-based environments, in which a large number of exploited devices are remotely controlled by an attacker. To detect botnets, Al Shorman et al. [86] proposed a method for detecting botnets in IoT environments called

GWO-OCSVM, which combines the Grey Wolf Optimization Algorithm (GWO) with the One-Class Support Vector Machine (OCSVM). This approach optimizes the hyperparameters of OCSVM using the GWO algorithm, allowing the intrusion detection system to adapt and improve its performance based on the dataset characteristics. The experimental results using the NN-BaIoT dataset demonstrated that GWO-OCSVM achieved good performance in terms of true positive rate and false positive rate, effectively detecting botnet activities in IoT-based environments.

Sybil attacks pose a huge threat in IoT-based environments, as adversaries create multiple illegal identities to deceive or disrupt IoT nodes. To address this issue, Murali and Jamalipour [87] presented a lightweight an IDS based on the Artificial Bee Colony (ABC). The ABC emulates the foraging behavior of honeybees through an optimization technique. The results demonstrated that the proposed IDS achieved average accuracy rates of 96.8%, 95.2%, and 94.8% for type 1 attacks, where malicious nodes target one fixed region, type 2 attacks, where malicious nodes are scattered among legitimate nodes, and type three attacks, where Sybil nodes are distributed across the network with mobility, respectively. This approach effectively detects and mitigates Sybil attacks in IoT environments.

For industrial infrastructures

The potential exposure of industrial infrastructures to disruptive events can have a significant impact on the economy's robustness. To secure such facilities, Liang et al. architechted an INIDS which utilizes a multi-functional data clustering optimization model to troubleshoot, recover, and rebuild [88]. Through performance evaluation using NSL-KDD and KDDCU'99 datasets, the proposed system can achieve 97.8% accuracy [88]. Industrial Cyber-Physical Systems (ICPS), which integrate advanced communication, computing, and industrial process supervision, are regarded as a core technology for smart agriculture. Different IDS approaches have been proposed to provide robust security for ICPSs, including the work of Liu et al., who introduced a hierarchically distributed IDS to achieve all-round security protection of ICPSs [89].

For smart grid systems

The Intelligent Network Systems in Intelligent Agriculture consist of Internet-connected controllers, automation and standard protocols to manage the production and distribution of energy [55]. Kurt et al. introduced an RL-based system for online cyberattack detection in Agriculture 4.0 smart grids [90]. The method involves two phases: training the model with low-magnitude attacks and testing for slight meter measurement deviations [90]. The study used the IEEE-14 bus power system for evaluation and demonstrated RL's potential in tackling complex cybersecurity problems. For robust protection of smart grid ecosystems, Patel et al. proposed a collaborative IDS capable of identifying attacks in centralized, distributed, and hierarchical forms [91].

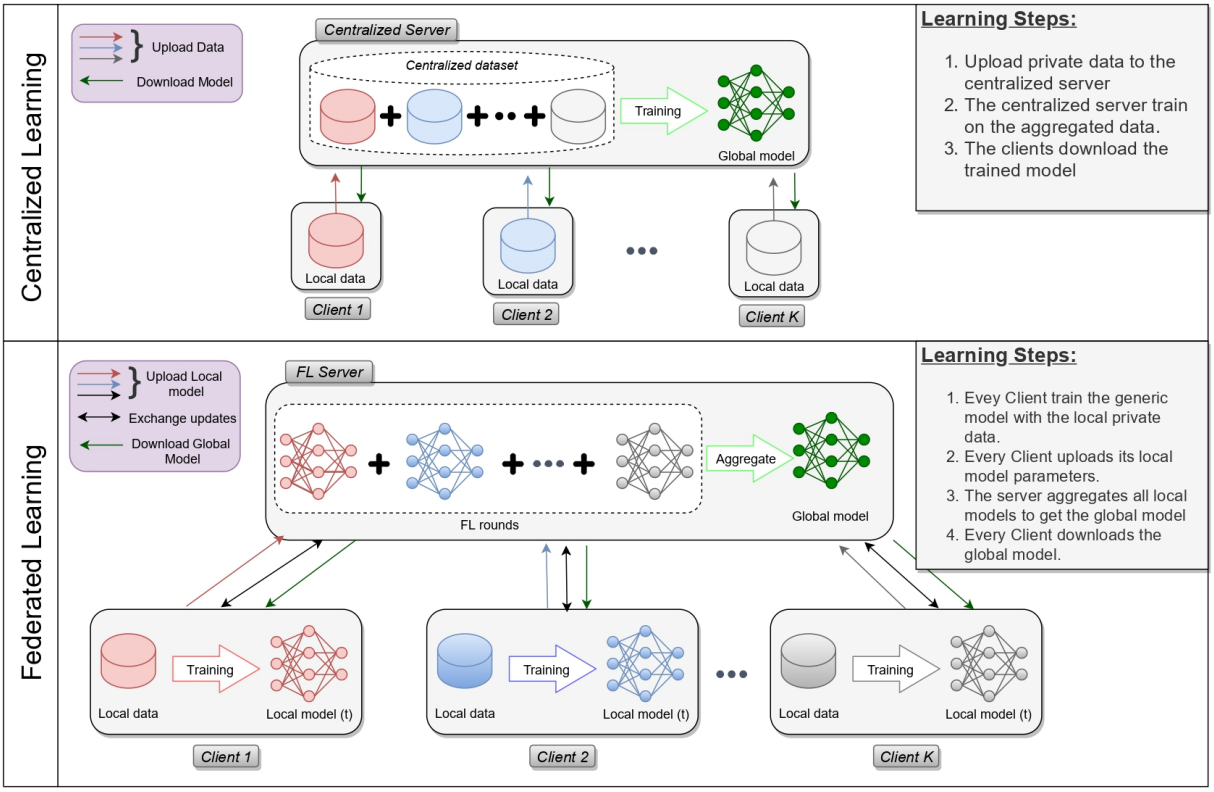


Figure 3.3: Federated learning compared to centralized learning

3.4 Federated Learning-based intrusion detection

Traditional ML techniques require data to be in a centralized location, leading to concerns about data privacy, communication overhead, and power consumption [66]. In contrast, Federated Learning (FL) offers a promising alternative that enables knowledge sharing while preserving privacy and reducing overhead [54], as depicted in Fig. 3.3 [66]. FL has gained significant success and is widely applied in various domains, including mobile edge networks, smart healthcare, ubiquitous systems, and augmented reality [66]. This section present an overview of FL and its relevance in intrusion detection, discussing its background and highlighting current state-of-the-art studies.

3.4.1 Background

The expression *Federated Learning (FL)* was first coined in a paper published by Google Research in 2016 [54]. The initial real-world integration was done on the Google keyboard, or Gboard on Android. When Gboard pushes a query suggestion, some local information is stored on the phone regarding the present context and if the query suggestion was actually clicked or not. FL process device history to provide suggestions for improvement for the next cycle of Gboard's feedback model.

FL is based on the concept that model training can be performed without the necessity to centralize or save the training data in a single location. The core idea behind FL is to create ML models using distributed data from multiple devices, without the require-

ment to share private data. The process involves downloading a generic model from the aggregation server, which is then updated on selected distributed devices using their local private data. These locally trained models are then sent back to the aggregation server, where their weights are averaged, resulting in a combined and improved global model. This process is iterated until the model converges to the desired performance level. There are three main types of Federated Learning: 1. Horizontal FL: In this type, the datasets have the same feature space but differ in the sample space. 2. Vertical FL: In this type, the datasets share the same sample space but have different feature spaces. And 3. Federated Transfer Learning: In this type, the datasets have both different feature spaces and different sample spaces.

FL with its diverse applications, addresses the privacy concerns associated with centralized data training and enables collaborative model development across distributed devices. It is a promising approach for enhancing the performance of machine learning models while preserving data privacy and security.

Federated Averaging

The initial version of FL introduced the Federated Averaging (FedAvg) algorithm [54]. In this algorithm, each client performs local Stochastic Gradient Descent (SGD) on its own data, while the server carries out model averaging. The choice of SGD is based on its success in various deep learning applications, and specifically, the large-batch synchronous SGD variant was adopted due to its superiority over asynchronous approaches in data center settings [92]. The core algorithm, known as FederatedSGD (or FedSGD), follows the Federated Averaging approach. The implementation of FedAvg is described in Algorithm 1 [54].

The FedSGD implementation with a fixed learning rate η and $C = 1$ involves each client k computing the average gradient for the current model w_t, using $w - \eta\nabla\ell(w; b)$, where ℓ represents the loss function. The central aggregation server then collect and merge these gradients to performs the update, which is known as the averaging step. Every client performs local training using its local data, and the server performs a weighted average of the resulting models. This approach allows devices to collaboratively train a distributed model using an aggregation server, while keeping all training data on-device. As a result, it enables the isolation of ML capabilities from centralized storage effectively. This feature of Federated Learning provides a promising solution to address concerns related to data privacy, communication overhead, and power consumption in traditional centralized ML techniques.

3.4.2 FL-based intrusion detection

In recent years, several FL-based contributions have been proposed by the scientific community in different areas, especially in AI-require privacy-critical domains, where the introduction of ML capabilities is essential and data sharing is strongly discouraged or extensive data acquiring is so difficult. Such domains include smart healthcare (found to be very helpful in COVID-19 pandemics), mobile edge network optimizations, industry 4.0, autonomous vehicle communications, augmented reality, the internet of drones,

Algorithm 1: *Federated Averaging*

```
Server executes():
    initialize w_0
    foreach round t = 1,2,.. do
        m ← max(C · K, 1);
        S_t ← (random set of m clients);
        foreach client k ∈ S_t in parallel do
            w_{t+1}^k ← ClientUpdate(k, w_t)
            w_{t+1} ← Σ_{k=1}^{K} (n_k / n) w_{t+1}^k
        end
    end
ClientUpdate(k, w):
    B ← (split P_k into batches of size B)
    foreach local epoch i from 1 to E do
        for batch b ∈ B do
            w ← w − η∇ℓ(w; b)
        end
    end
```

and intrusion detection. The focus in this part will be placed on a number of different FL-based systems proposed for intrusion detection tasks. A brief classification of these systems is provided below, including distributed intrusion detection, cyber security for IoT, autonomous systems, computing infrastructures, and Blockchain-assisted. For a richer and more elaborated classification of the subject, the reader is invited to consult [66]. While centralized ML-based IDSs can deliver robust anomaly detection models, the privacy-preserving feature is not fully guaranteed. FL-based IDSs can provide both privacy-preserving and high anomaly detection.

Distributed intrusion detection

FL-based intrusion detection allows users to learn from the knowledge shared by their peers regarding benign and attack patterns. For instance, Li et al. [93] designed an FL-based distributed IDS specifically tailored for network traffic security and preserving privacy in heterogeneous networks. Their system targets embedded satellite-terrestrial networks, effectively analyzing and blocking malicious traffic, such as DDoS attacks. The IDS proposed by Li et al. [93] incorporates two main mechanisms. Firstly, it leverages homomorphic encryption to enable multi-party secure computation within the FL framework. This ensures that privacy is maintained while sharing and processing data among participating devices in heterogeneous networks. Secondly, the IDS utilizes CNN to enhance its accuracy in analyzing and detecting various types of cyber threats in industrial CPSs. Additionally, GRU are employed in the training process to further enhance the model's performance. To safeguard the privacy of exchanged model parameters during FL training, Li et al. [94] employed the Paillier public-key cryptosystem. This crypto-

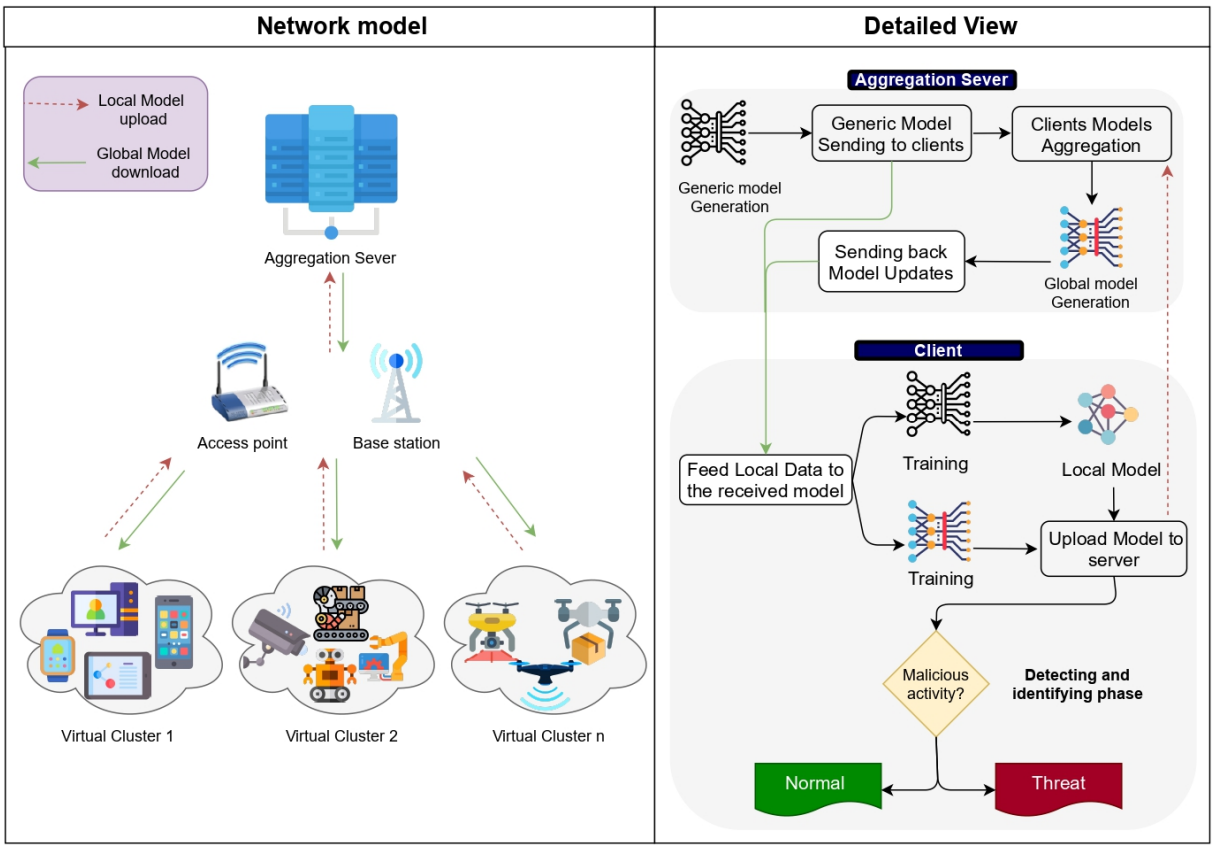


Figure 3.4: FL-based intrusion detection for IoT

graphic technique ensures that sensitive information remains confidential throughout the training phase. For evaluating the effectiveness of their proposed system, the researchers used the Gas Pipeline System dataset. The results demonstrated promising performance, achieving an F-score of 98.14%, recall of 97.47%, precision of 98.85%, and an impressive accuracy of 99.20%.

Cyber security for IoT

The rapid expansion of IoT technology has led to the proliferation of millions of interconnected embedded physical devices. Each of these devices exposes data that could potentially compromise the privacy of its users. Unfortunately, the lack of reliable security defenses across IoT devices makes them vulnerable to exploitation, turning them into a wide surface for active cyber attacks. To address these security challenges, the use of FL-based IDS for IoT is becoming increasingly important. Figure 3.4 illustrates the main abstracted steps involved in creating an FL-based IDS for IoT. These steps typically involve collecting data from distributed IoT devices, securely aggregating and analyzing the data without compromising privacy, and then deploying the trained intrusion detection model back to the IoT devices. By following this approach, IoT ecosystems can benefit from collaborative learning while ensuring data privacy and enhancing security against potential cyber threats. Different FL-based intrusion detection schemes have been proposed for protecting IoT-enabled systems including:

- *Malicious devices detection:* In the realm of IoT, there is a significant number of vulnerable devices at risk of being exploited due to insecure designs, deployments, and configurations. Such vulnerabilities pose a grave threat, especially in critical applications like surveillance. To address this issue, researchers have proposed FL-based autonomous self-learning distributed schemes for identifying compromised or malicious IoT devices. Nguyen *et al.* [95] presented DIOT, an FL-based system capable of autonomously detecting malicious IoT devices. The evaluation of DIOT demonstrated its effectiveness in identifying attacks, achieving a detection rate of 95.6% within an average response timing of 257 milliseconds. Another noteworthy contribution in this area comes from Mohammed *et al.* [96], who presented an FL-based online stateful heuristic combined with an alarm app for IoT clients. This innovative system allows clients to be promptly alerted about the presence of unauthorized IoT devices in their environments. Simulation experiments on real datasets showed a significant 27% enhancement in accuracy compared to other approaches dealing with the same problem.

- *IIoT security:* Due to their size, cost, and power consumption, these eye-catching characteristics have led to the widespread adoption of IoT in smart factories for monitoring machines, the guidance of automated processes, or assisting in generating a virtual representation of systems which can be used for advanced simulation using digital twins. Hao *et al.* [97] introduced a privacy-enhanced FL system called PEFL, designed specifically for enhancing security in industrial AI applications. PEFL utilizes Augmented Learning with Error (A-LWE) and homomorphic ciphertext of private gradients exchange to enhance privacy and protect sensitive data. Additionally, the system incorporates differential privacy with a distributed Gaussian mechanism for further safeguarding data privacy. To evaluate its performance, the researchers conducted benchmark tests on the MNIST dataset. The results demonstrated that PEFL not only achieved excellent accuracy but also showed significant improvements in terms of communication and computational costs, making it a highly efficient and effective solution for securing industrial AI.

- *Mobile Crowdsensing:* Mobile crowd sensing is a rising focus in IoT. It involves a paradigm that employs smart device-wearing individuals, called "workers," to perform various sensing tasks. Zhang et al. [98] proposed an innovative framework named FedSky for secure data aggregation in federated mobile crowd detection. The framework incorporates an intelligent worker selection strategy that avoids random user selection. Instead, it selects users based on the local data size and the processing capabilities of their mobile devices. Compared to the traditional FedAvg, FedSky demonstrates significant improvements in terms of both users' processing time and overall system latency. By optimizing the selection of users for data aggregation, the proposed system achieves more efficient and faster processing of data in federated mobile crowd detection scenarios.

- *5G-enabled IoT:* IoT environments are diverse in shape, size, nature, and operational tasks, making it hard to effectively provide secure communications among network devices and their service servers, especially in 5G-enabled IoT, due to the required high speeds and low latencies. Yu *et al.* [99] presented UDEC, an FL-based

distributed model trained on deep reinforcement learning, which is designed to secure critical service requests from uses at the edge nodes, by providing privacy of services, a cheap and dynamic scheduling, and a full usage of system resources. The performance evaluation showed the effectiveness of the UDEC model in addressing these challenges, and also in terms of energy usage.

- *Internet of Healthcare Things (IoHT):* The COVID-19 outbreak initiated a worldwide crisis that necessitated collaborative efforts to combat it. The effective identification of infected patients is a critical factor in the assessment and response to COVID-19, with AI being a key enabler. Yet, the issue with classic AI is sharing sensitive data, causing many privacy concerns, at which point the FL becomes necessary. Various proposals for the healthcare industry recommend the use of FL techniques to maintain safeguard data privacy while benefiting from the data of other hospitals. For example, Chen *et al.* [100] proposed FedHealth, a federated transfer learning framework, for securing wearable healthcare devices and protecting data privacy.

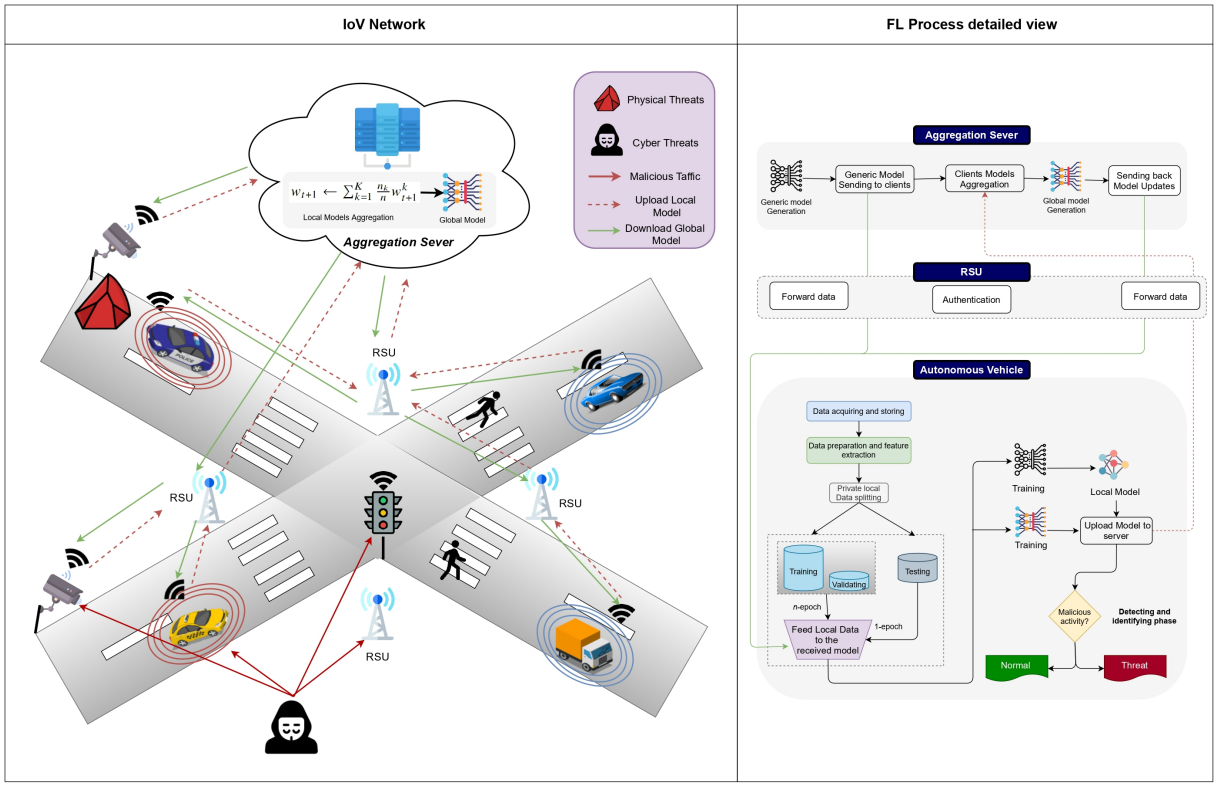


Figure 3.5: FL-based cyber and physical threats detection for autonomous systems

Cyber security for autonomous systems

FL is leading a new line of development for autonomous systems/drives in which a diversity of data collected from different sources can be employed while maintaining the privacy of the users. Vehicular IoT networks or Internet of Vehicles (IoV) enable a more secure travel experience and a more enhanced onboard experience, moving toward a smart, self-driving automotive future. FL implementation in the area of data-driven navigation

leverages the data collected by mobile users and embedded computational resources, which enhance considerably the learning speeds while providing improved assurances to data privacy, as illustrated in Fig. 3.5 [66]. Lu *et al.* [101] suggested a federated peer-to-peer vehicle learning framework that employs random updating of conservative-free sub-pots, increasing both security and reliability. Aggregation is performed asynchronously. When executing a co-learning task including data sharing or leak detection, each vehicle acts as a participant to perform FL rounds. A distributed hash table is used to record the information from the vehicle data search in the system's neighboring Road Side Units (RSUs).

Thanks to their remarkable advantages, such as low cost, faster operational deployment, and more flexible movement, UAVs enable autonomous crowd sensing at any time and in any place. Yao *et al.* [102] proposed a secure FL-based system to neutralize eavesdropping in fog-aided Internet of Drones (IoD) networks. The main concept of this proposal is monitoring UAV's power consumption patterns for optimizing the safety rates.

Cyber security for Edge/Cloud computing infrastructures

Storing data in a single location (e.g., a cloud data center) to perform ML training may not be appropriate for individual users in terms of privacy risks. Various FL-based systems have been proposed to overcome these limitations, for instance, Fang *et al.* [103] introduced HFWP, an FL-based scheme with robust privacy-preserving, intended for securing cloud computing. Built on a lightweight encryption protocol, the HFWP scheme provides robustness against colluding parties and honest but curious servers. Using real-world datasets, namely MNIST and the UCI Human Activity Recognition dataset, experimental results demonstrated improved accuracy over other existing studies.

New emerging modeling paradigms such as Mobile Edge Computing (MEC) and next-gen communication technologies are essential to sustain the fast development and rollout of IoT networks. As the networks of IoT expand, figuring out the best allocation of scarce resources for delivering high-quality services across IoT becomes a significant challenge. However, FL-based edge learning can provide some solutions, for instance, Xiao *et al.* [104] proposed FEI, a federated edge intelligence framework, for ensuring joint optimization of the IoT network and the edge server. The FEI systems involve a cluster of edge servers that build a shared model by utilizing the collected and uploaded data from IoT devices.

FL-based Blockchain-assisted cyber security

The decentralized nature of blockchain technology is being exploited by various researchers to give the FL approach a fully decentralized character. In other words, blockchain supplies an operative method for removing the central aggregation server, which presents a Single Point of Failure (SPOF) and is additionally subject to attacks in untrusted environments. For example, to carry out authentication and trust management for federated nodes along with the edge training, Rahman *et al.* [105] introduced a hybrid lightweight FL system that leverages blockchain smart contracts to enhance the security of the IoHT. This system is meticulously designed to facilitate various crucial tasks, including the inference process, model learning, and comprehensive dataset encryption.The core mechanism

involves deploying blockchain technology to aggregate the updated model parameters, a task carried out with multiplicative encryption. Simultaneously, individual federated edge nodes employ additive encryption to protect the privacy of their respective datasets.

3.5 Related Works

In this chapter, we will provide an overview of the existing literature that focuses on ML and FL approaches for IoT-based IDS.

3.5.1 FL and FL in the Edge

Recently, FL has garnered considerable attention due to its unique ability to facilitate knowledge sharing among multiple users while preserving the confidentiality of their data. McMahan et al. [54] proposed the FedAvg algorithm, enabling the creation of a global model without the necessity of transmitting clients' data to a central server. Another iteratversionion of FL, known as Federated Edge Learning (FEEL), shares similarities with FL in its design principles but situates the server at the network's edge. This approach aims to reduce latency by bringing FL closer to the data sources. In an effort to further enhance FL's efficiency, Mills et al. [106] introduced CE-FedAvg, a communication-efficient variant of FedAvg. CE-FedAvg is tailored to decrease the number of rounds required to attain the desired accuracy level and minimize the amount of data downloaded per round, as compared to the original FedAvg. Empirical assessments conducted in an edge computing environment using MNIST and CIFAR-10 datasets demonstrated that CE-FedAvg significantly expedites convergence when compared to FedAvg.

3.5.2 IDS for IoT

The increasing deployment of IoT devices in the agricultural sector has raised concerns about their vulnerability due to design flaws, poor implementation, and bad configurations. This vulnerability makes Agri-IoT networks susceptible to easy exploitation. To address this issue, researchers have proposed various IoT-based IDS. In one study, Ferrag et al. [84] introduced a deep learning-based IDS specifically designed to mitigate DDoS attacks in Agriculture 4.0 environments. The proposed system showed promising results in experiments conducted on the CIC-DDoS2019 and TON_IoT datasets. Another approach proposed by Ferrag et al. [85] is a hierarchical IDS named RDTIDS for IoT networks. RDTIDS combines multiple classifiers, including the REP tree, JRip algorithm, and Forest PA. Experimental evaluations using the CICIDS2017 and BoT-IoT datasets demonstrated a significant detection rate with minimal false alarms. While integrating Blockchain and IDS has shown success, the aforementioned studies still require centralization of data for model training and testing, which can be a drawback for privacy-sensitive data.

3.5.3 ML for SDN

SDN technology offers benefits such as enhanced network efficiency, improved management, and programmability. However, it is also susceptible to various security threats. To address these concerns, researchers have explored the application of ML for SDN security. Using ML models using classifiers like C4.5, BayesNet, Decision Table, and Naive-Bayes, trained on historical data of network assaults, is one method suggested by Nanda et al. [107] to predict susceptible hosts in SDN. The outcomes demonstrated that the BayesNet classifier has a 91.68 percent accuracy rate. The specifics of the training data, though, weren't completely shared. Detection of ransomware using ML in SDN was the focus of another study by Cusack et al. [108]. The researchers created a random forest classifier to detect ransomware and employed programmable forwarding engines to rapidly collect packet-based network monitoring data. The approach had a detection accuracy rate of about 87 percent, but it could only identify one form of attack and had a high false-negative rate of roughly 10 percent.

3.5.4 DFL-based IDS for IoT

For ML-based IDS to be successful, a model must be trained using a lot of data. However, because to privacy concerns and significant communication latency, centralizing all data for training and testing may not be viable in complicated situations involving several parties [109]. Researchers have focused on FL as a viable strategy for cyber security in IoT and IIoT contexts to address these issues. FL is appropriate for situations where data privacy is crucial since it permits knowledge sharing among numerous parties while maintaining data decentralization. Following the release of the *FedAvg* by McMahan et al. [54], the notion of FL gained popularity. Since then, a number of studies have been carried out in this area (see, for example, [66, 110]) with the goal of utilizing FL for the creation of reliable IDS solutions specifically suited to the problems that the IoT and IIoT environments provide. Preuveneers et al. [109] investigated the application of federated IDS based on DL in conjunction with a permissioned blockchain in their study. To provide transparency throughout the distributed learning process, they used an autoencoder that was coupled with the MultiChain platform and trained on the CICIDS2017 dataset. In contrast to more complex models suggested in other literature, the neural network design used in their experiment is rather straightforward. The generalizability of their findings to other applications might be constrained as a result.

Moreover, the FedAvg algorithm used in their approach may require customization for each specific application to achieve optimal performance, which could indirectly impact the interaction with the MultiChain blockchain. Through studies with centralized, device-based, and federated learning on the NSL-KDD dataset, Rahman et al. [111] proposed a FL-based IDS for IoT networks. Comparable to the results of the centralized model evaluation, the FL-based IDS had an accuracy of about 83.09 percent. But because more recent cyberattacks are always developing, using an older dataset like NSL-KDD may limit its use for IDS. With an emphasis on identifying Mirai-infected devices, Nguyen et al. [95] suggested a FL-based IDS that is specifically created for different types of IoT devices. During a physical implementation including Mirai-infected devices, the system

proved its capacity to detect 95.6 percent of attacks in a speedy 257 milliseconds with few false alarms. Other IoT/IIoT threats and attacks are not taken into account by the model; instead, it only takes malware-based threats like Mirai into account. The suggested methodology also ignores threats aimed at other ecosystem components like complex networking technologies (like SDN) and services (like FTP, SSH), and is solely intended to detect attacks against IoT devices.

An IDS based on FL called "DeepFed" was proposed by Li et al. [94] to improve the security of Industrial CPSs. An aggregate server, a trusted authority, and several learner client sets, ranging from 3 to 7, were used to test the system using real-world industrial CPS data. The threat model for the system included a variety of attacks. The evaluation's usage of a small number of clients, however, might make it harder to apply the findings to IoT systems that are more intricate and densely networked. To evaluate its effectiveness in these circumstances, more research is required. In their research, Schneble et al. created an FL-based IDS that was specifically designed for Medical CPSs [112]. Using data from actual patients, the system's performance was assessed, and its accuracy score was a remarkable 99.0 percent. It is important to note, nevertheless, that the authors' attention was restricted to just three categories of harmful data patterns. Because of this, it's possible that not all potential attack vectors—including malware attacks, IoT protocol attacks (like MQTT attacks), application attacks (like XSS and Brute-force attacks), and service attacks (like FTP and SSH attacks)—are covered by the threat model under consideration.

In order to offer a more thorough evaluation of the system's effectiveness, future study should think about extending the range of assault types. A Low-Complexity Cyberattack Detection system made specifically for IoT Edge Computing, LocKedge, was introduced by Huong et al. [113] . Using the BoT-IoT dataset, the system's performance was evaluated against those of other ML techniques. An FL-aided Long Short-Term Memory framework for intelligent IDSs, known as FL-LSTM, was proposed by Zhao et al. [114] in a separate article. This framework had an accuracy rate of about 90%. It is crucial to remember that the dataset used to test FL-LSTM is made up of command blocks that were built from a list of user commands beforehand. Due to this dataset's potential limitations, network-based IDSs may not be able to detect all dangers present in network traffic.

In order to protect FL-based schemes from adversarial attacks aimed at IIoT cloud applications, Song et al. presented a defense mechanism [115]. This solution makes it easier to gather military intelligence from many sources and makes it possible to share that insight with IIoT devices working in a cloud-based architecture. On the other hand, Hao et al. [97] produced a FL scheme known as PEFL that was intended to produce an IDS that protects privacy for commercial artificial intelligence settings. Through the use of a distributed Gaussian process, the authors proposed a secure aggregation technique based on homomorphic encryption and Differential Privacy (DP) to safeguard training data privacy. The MNIST dataset was used to evaluate the suggested system's performance. To identify failures in the IIoT ecosystem, Zhang et al. presented a blockchain-based FL platform architecture [116]. The suggested method uses the Centroid Distance Weighted Federated Averaging (CDW FedAvg) algorithm, which takes into account the positive class to negative class distance of each customer dataset, to guarantee the data integrity

of the users. Similar to this, Khan et al. suggested a security solution for supply chain 4.0 intrusion detection dubbed DFF-SC4N, based on FL [117]. With Recurrent Managed Units (RMUs), the system uses rounds of communication in a FL fashion, exchanging only the learned metrics and leaving the rest of the data intact on the local server.

Taheri et al. developed Fed-IIoT, an innovative federated learning (FL) architecture aimed at detecting Android malware apps in IIoT domains [118]. The architecture comprises two primary sections: the participant side, where data is generated from two different poisoning threats using a Generative Adversarial Network (GAN) and a federated GAN; and the server side, responsible for supervising the aggregated model and constructing a robust collaborative training model. To ensure secure communication during the training process, the authors employed a Paillier cryptosystem to protect the exchanged parameters. Similarly, Mothukuri et al. proposed a FL-based approach for anomaly detection in IoT networks to identify network intrusions [119]. Their method involved training Gated Recurrent Units (GRU) models while preserving the privacy of local IoT device data. The learned weights were the only information shared with the aggregation server, ensuring the privacy of individual device data. Fed-IIoT is an FL architecture created by Taheri et al. to find Android malware apps in IIoT domains [118]. The architecture is divided into two main sections: the participant side, which generates data from two different poisoning threats using a federated GAN and a generative adversarial network (GAN), and the server side, which is in charge of managing the aggregated model and building a reliable collaborative training model. The authors used a Paillier cryptosystem to safeguard the transmitted parameters in order to ensure secure connection throughout the training procedure. Similar to this, Mothukuri et al. suggested using a FL-based approach for anomaly detection in IoT networks to spot network intrusions [119]. Training Gated Recurrent Units (GRU) models while protecting the confidentiality of local IoT device data was their approach.

Limitations

It is evident from the previous discussion that the existing literature has certain gaps. These gaps include the use of outdated or contextually inappropriate datasets, privacy concerns associated with centralized models, and limited threat models that only address specific attack vectors. However, the mentioned works make effective use of Federated Learning (FL) advantages, such as local training without sharing private data, ensuring the protection of client data from malicious entities. Nevertheless, FL does come with its challenges [120]. The previous discussion made clear that there are several gaps in the literature that are there. The use of obsolete or contextually incorrect datasets, privacy issues related to centralized models, and constrained threat models that exclusively handle particular attack vectors are a few examples of these gaps. However, the aforementioned works effectively utilize the benefits of FL, such as local training without disclosing personal information, guaranteeing the security of client data from harmful parties. FL does, however, have certain difficulties, as noted by [120]. For instance, by examining the differences in FL-client training weights, adversaries can potentially compromise data privacy to some extent [121]. Additionally, generative sequence models may unintentionally memorize private and sensitive training data, potentially making it possible to

recover some training data information from the model [122]. Additionally, the aggregation server poses a Single Point Of Failure (SPOF) risk to the design of the entire network. Furthermore, depending on outside components, like current blockchains, may result in network and processing overheads, which may not be suitable for some time-sensitive and real-time security applications. To secure the security of smart agricultural systems, it is essential to address these issues.

3.6 Conclusion

In conclusion, this chapter has examined the use of IDS as a defense strategy for enhancing the security of smart agriculture. We started with an overview of IDS, including its definition, types, and design stages, followed by a discussion of its potential benefits for the agricultural sector. We then delved into the technological solutions available for IDS-based security in smart agriculture, before turning our attention to the anomaly-based intrusion detection approach, specifically the Federated Learning approach. The chapter provided an introduction to the topic, explored distributed intrusion detection, and reviewed some state-of-the-art works. Overall, the chapter highlights the importance of implementing effective and recent both centralized and FL-based IDSs as security measures in smart agriculture to safeguard against potential cyber-attacks.

Chapter 4

A robust security architecture built on Blockchain-supported fog and SDN for agricultural IoT

4.1 Introduction

The Internet of Things (IoT) is a network of cutting-edge physical objects with communication capabilities that make data transmission and aggregation easier [123].

In the framework of intelligent agriculture, sensors are installed all over the farm to track and collect information on many factors, such as the quality of the soil, the weather, irrigation, light, and air. The farmer, as well as any farm robots or drones operating in the field, receives this data after that. A "smart" farming system's networked data interchange is a key component that enables informed decision-making and effective administration of agricultural operations. The relevance of real-time data in agricultural IoT networks cannot be overstated. According to Cisco, fog computing is the best option for addressing the requirement for lower latency. Extending cloud computing capabilities to the edge of the network is known as fog computing. Fog computing uses devices with storage, processing power, and network connectivity to enable partial processing at the network's edge rather than handling all data in the cloud. These edge devices gather information from IoT devices linked to the IoT application, making data processing and analysis more effective and quick. Fog computing improves the overall performance and dependability of agricultural IoT systems by shifting processing jobs closer to the data sources.

Agricultural IoT networks exhibit specific characteristics, such as high scalability, heterogeneity, and stringent management and control requirements. Conventional network architectures may not suffice to meet the demands of IoT networks, especially concerning reliability and low latency requirements for IoT applications. To address these challenges, emerging technologies like SDN and NFV offer promising solutions. SDN involves separating the network's control functionality from its data transmission functionality, providing a more flexible and programmable network environment [25]. On the other hand, NFV abstracts network transfer and related functions from the underlying hardware, enabling the dynamic deployment of network services. Combining SDN and NFV allows leverag-

ing the advantages of both approaches to enhance infrastructure flexibility. This dynamic and adaptive network design, provisioning, and operation are crucial for supporting the diverse requirements of agricultural IoT networks.

Blockchain technology has emerged as a promising solution for establishing traceability and transparency in product and food supply chains [43]. It addresses the issue of trust by offering an innovative approach to record-keeping and transaction verification [124]. This electronic Distributed Ledger Technology (DLT) operates on a Peer-to-Peer (P2P) system architecture, enabling multiple users to transparently share and create immutable records of transactions, known as blocks. Each block is time-stamped, write-once, append-only, and linked to the preceding one, ensuring data integrity and security. Blockchain is often referred to as the *internet of value* due to its ability to safeguard the storage and facilitate secure transactions of valuable assets [124].

The preceding chapter focused on the technological components of various IDS security solutions for smart agriculture. In this chapter, we offer a thorough security architecture for the agricultural IoT that combines SDN, fog computing, and blockchain technologies. Three primary parts make up the security architecture that is being suggested. In order to make real-time data gathering, analysis, visualization, and device management possible, an agricultural IoT data management system is first put into place. This system ensures efficient handling of IoT data generated from sensors deployed across the smart farm. Secondly, a blockchain-based integrity monitoring scheme is incorporated to safeguard against erroneous delivery of controls and information within the agricultural IoT network. By utilizing blockchain's immutable and transparent nature, the system can ensure the integrity and reliability of data and commands exchanged between devices and the central management entity. Thirdly, a virtual switch software is introduced to support software-defined networking technologies, which enhances network management capabilities. By decoupling the network control and data planes, SDN allows for dynamic and adaptive network provisioning, crucial for meeting the unique demands of IoT networks. Furthermore, we conduct extensive testing of the proposed security architecture on an open-source IoT platform. We specifically integrate Hyperledger Sawtooth blockchain and software-defined networking technologies to evaluate the architecture's performance under simulated DDoS attacks. The results of our performance evaluation demonstrate the effectiveness and robustness of the proposed security architecture in mitigating threats and ensuring a secure smart agricultural environment.

By combining blockchain, fog computing, and SDN, our security architecture presents a comprehensive solution that addresses the specific challenges of the agricultural IoT domain, providing enhanced data management, integrity monitoring, and network management capabilities. The successful evaluation results further validate the efficiency of our proposed approach in safeguarding the agricultural IoT network from potential security threats.

4.2 System and network model

In this section, we present a detailed explanation of the various components that constitute our proposed security architecture, as well as their interrelationships. Fig. 4.1 provides a

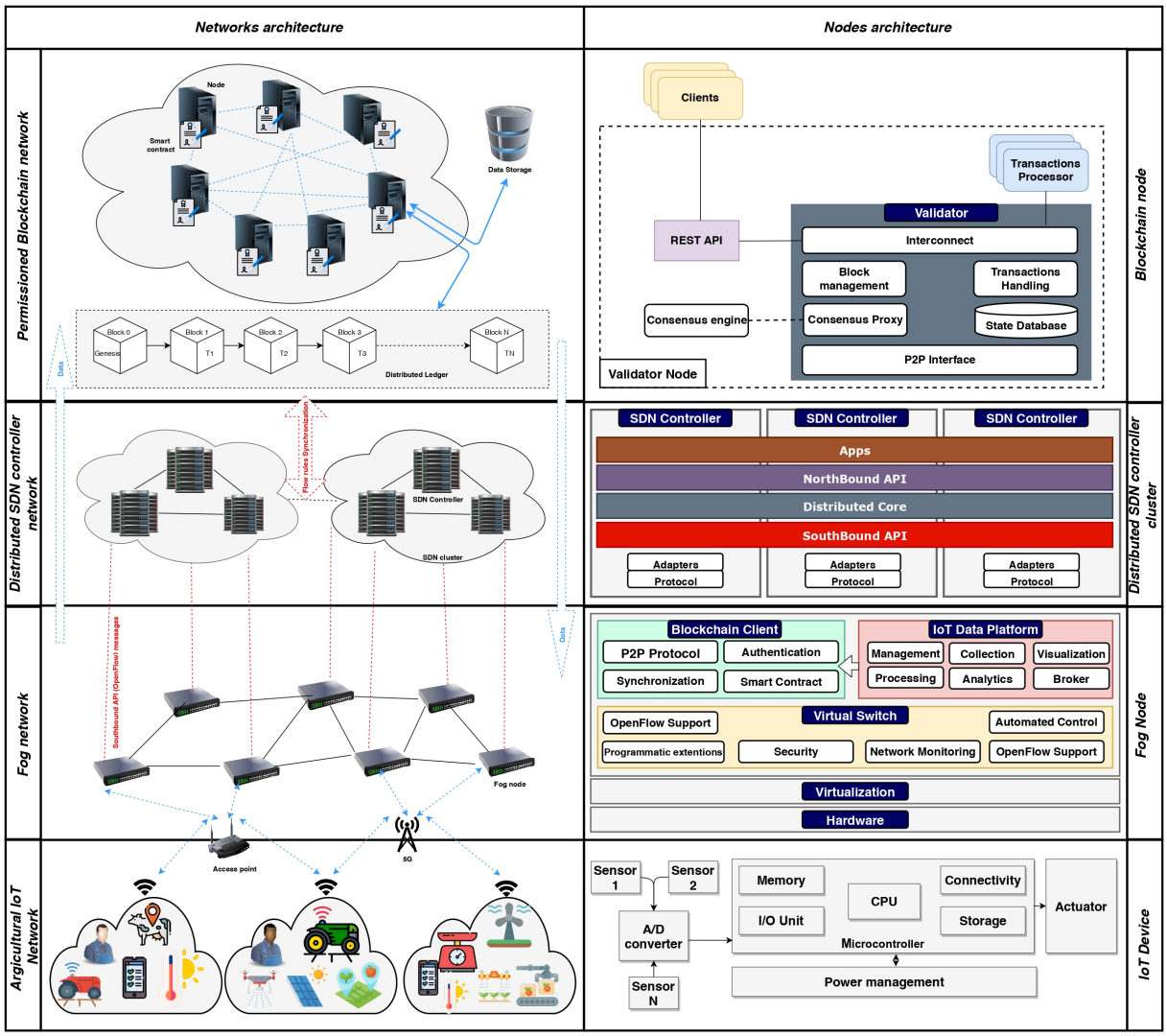


Figure 4.1: Proposed security framework architecture

comprehensive visual representation of our system architecture.

4.2.1 Agricultural IoT Layer

The foundation for gathering important information from the physical environment using a variety of sensors and enabling different actions is the main fuction of this layer. The various IoT devices with various form factors and functionality are included in this layer. Inpu and output modules, data processing units, network modules, and power management units are among the common components shared by various devices. The linked fog nodes receive the sensor data from the IoT devices and process and store it before sending it back to the IoT devices for further processing. Additionally, these gadgets are capable of receiving instructions to carry out particular tasks in an agricultural area. Smart and adaptive agricultural operations are made possible by this data sharing and action execution at the IoT layer.

4.2.2 Fog Layer

The Fog layer comprises multiple fog nodes, lightweight devices equipped with computation, networking, and storage capabilities to handle tasks that cannot be efficiently executed by end devices alone. In contrast to transmitting all gathered data to the cloud for processing, the Fog layer facilitates real-time analysis and latency-sensitive applications. This enables quick decision-making and responsive actions within the agricultural IoT system. For permanent, large-scale data storage and in-depth analysis of the global dataset, the Cloud computing layer comes into play. The Cloud resources are accessed periodically and in a managed manner, ensuring optimal and efficient utilization of cloud resources. By distributing computation and storage responsibilities between the Fog and Cloud layers, the system achieves a balance of real-time responsiveness and comprehensive data analysis. The core components of the fog node are:

Virtualization

To streamline the setup and configuration process for multiple fog nodes, virtualization techniques can be employed. By encapsulating the fog node within a Docker container, the setup becomes more efficient and consistent for all other fog nodes. Docker containers provide a lightweight and portable environment, ensuring that each fog node operates within a standardized and isolated container. This approach simplifies the deployment and management of fog nodes, reducing the configuration overhead and optimizing resource utilization.

Agricultural IoT Data Management

A real-time IoT data gathering, analytics, visualization, device management, and local storage is provided by the Agricultural IoT Data Management node. While effortlessly adjusting data intended for the cloud, it allows users to access data from the fog layer. To facilitate efficient data communication, a Publish/Subscribe messaging pattern is employed. In this pattern, the publisher in the perception layer sends messages to a message broker installed in this node. The message broker filters and broadcasts the messages to the appropriate subscribers, ensuring that relevant data reaches the intended recipients in a timely manner. This data management node plays a crucial role in processing and distributing IoT data within the agricultural IoT ecosystem. Alg. 2 presents a pseudo-code of the steps the fog node conducts after getting an event from the perception layer; u is the unit ID or smart farm ID in which the d device is located; s introduces the sensor; and r_v the value received from it.

Blockchain client logic

The Blockchain client is a software program that resides within the fog node and is designed to facilitate the synchronization of real-time sensitive IoT data to the DLT. Algorithm 3 provides a pseudo-code representation of the operations performed by the Blockchain client [125]. The main functions of the client include creating transactions

Algorithm 2: IoT data management process

```
Input: u, d, s, r_v;
Initialization: Create a key-value array : res ← {"k" : "v"};
for i = 1, 2, ..., input.size() do
    Copy inputs into res: res[input[i].key()] ← input[i].value();
end
Add timing: res["t"] ← current_timestamp();
if The reported value r_v exceeds the threshold then
    Create alert with notification type nt: a ← {res, nt};
    Send a to the admin;
end
Update the dashboard with the values in res;
Sent res to the blockchain client; /* Run Algorithm 2 */
Generate a packet p that includes res and the cloud broker IP, and port;
Transmit p; /* Run Algorithm 3 */
```

containing the IoT data, packaging these transactions into batches, and subsequently transmitting them to the validators. The validators, in turn, group the received transactions into blocks and finalize their inclusion in the Blockchain network. This process ensures the secure and transparent recording of real-time sensitive IoT data, enhancing data integrity and traceability within the agricultural IoT environment.

Virtual switch

The virtual switch node operates as a forwarding device, leveraging virtual switch software that supports NFV and SDN technologies to enhance network management.This node may accept many network virtual services, such as security, Quality of Service (QoS), and automated control, by enabling programmable extension and control of forwarding functions. The main goal is to create a switching stack suited for situations using hardware virtualization and supporting the OpenFlow protocol [126]. This enables direct interaction between the virtual switch node and the SDN controller. An example of such an implementation is *Open vSwitch*[1], which is an open-source, multi-layered software switch. Algorithm 4 describes a pseudo-code representation of the steps the SDN-enabled virtual switch takes after receiving a packet. These actions contribute to effective packet forwarding and management within the agricultural IoT network, enhancing overall network performance and control.

4.2.3 Distributed SDN controller network

The distributed SDN controller network is a critical component of the proposed security architecture, comprising multiple geographically distributed and interconnected SDN controllers. These controllers collectively function as a shared Network Operating System

[1]http://www.openvswitch.org/

Algorithm 3: Blockchain client

Input: *pa*;
let *sk* be the client's private key, and *pk* the client's public key;
Encode the received Transaction payloads *pa* with the Transaction Processor format *f*: $fpa \leftarrow \text{encode}(pa, f)$;
Create a Transaction header *th* that includes *pk* and a unique transaction ID *tid* : $th \leftarrow \text{SHA-512}\{pk, tid\}$;
Sign *th* with *sk* : $tsig \leftarrow \text{sign}(sk, th)$;
Create a transaction : $t \leftarrow \{th, sig, fp\}$;
if *One or more Transaction instances ready* **then**
    Put all lists of Transactions into an array *tl*;
    Put a list of Transaction IDs into an array *tlid*;
    Create a batch header : $bh \leftarrow \{pk, tlid\}$;
    Sign *bh* with *sk* : $bsig \leftarrow \text{sign}(sk, bh)$;
    Wrap all transactions into the batch : $b \leftarrow \{bh, bsig, tl\}$;
    Encode the batch(es) in a batch list : $bl \leftarrow \text{encode}(b, f)$;
    Generate a packet *p* that includes *bl* and the blockchain validator IP, and port;
    Transmit *p*; `/* Run Algorithm 3 */`
end

Algorithm 4: SDN-enabled virtual switch main tasks

Input: *p*;
Check local flow tables for an entry matching packet *p*;
if *No match is found in a flow table* **then**
    Encapsulates *p* into a "Packet-in" message *pi*;
    Forward *pi* to the SDN controller over the OpenFlow channel;
    Wait for a response;
    if *Received a flow modification packet "Modify-State"* **then**
        Add a flow entry to the flow table;
        Take actions on *p* based on the matching flow entry;
    end
    if *Received a "Packet-Out" message* **then**
        Route *p* based on "Packet-Out";
        exit;
    end
end
Take actions on *p* based on the matching flow entry;
Route *p* based on the matching flow entry;

(NOS), ensuring a combination of logically centralized control and decentralized operations [25]. The SDN controller serves as the core element of the SDN network, by handling all communications between apps and network devices, supporting effective network management, enabling effective management and modification of network flows in response to altering requirements [127]. Having a coherent global view of the network, the SDN controller relays information to fog nodes through southbound APIs (e.g., OpenFlow), and to applications through northbound APIs (e.g., MQTT, CoAP, etc.) [127]. This bidirectional communication allows seamless interaction between the SDN controller and various network components, enabling efficient flow control and management.

To maintain the SDN flow rules' integrity and security, the SDN controller incorporates a Blockchain client. This client is responsible for securely storing SDN flow rules in the Blockchain, mitigating the risk of rules forgery and providing a tamper-proof record of network configurations. By leveraging the Blockchain's distributed nature, the SDN controller enhances the trustworthiness of flow rule management and prevents unauthorized modifications to the network's operation. This integration of SDN and Blockchain technologies adds an extra layer of security to the agricultural IoT network, safeguarding against potential attacks and ensuring the reliability of network operations.

4.2.4 Blockchain network

The proposed security architecture includes a Blockchain network, consisting of interconnected Blockchain nodes that continuously share the latest data with each other, ensuring all nodes remain updated with the most recent information. This network is designed to maintain the integrity and transparency of data stored in the Blockchain, allowing for credible transactions to be executed without the need for third-party intervention [128]. Smart contracts play a crucial role in the Blockchain network, as they are responsible for triggering and enforcing appropriate actions based on the stored data. These smart contracts enable the execution of predefined actions when specific conditions are met, facilitating automated and secure transactions within the Blockchain network.

The *Hyperledger Sawtooth* [125] Blockchain platform is the foundation for the development of the Blockchain network. This platform was selected because of its extremely adaptable and modular design, which distinguishes the main system from the range of applications. Due to this division, smart contracts can establish application-specific business rules without the requirement for in-depth knowledge of the core system's underlying design. A reliable and scalable foundation for the Blockchain network is provided by the modular design of Hyperledger Sawtooth, ensuring effective and safe management of agricultural IoT data and transactions. A full node in the blockchain consists of:

Validator node

The validator node plays a critical role within the Blockchain network. It is a core component responsible for validating transactions batches and consolidating them into blocks. Additionally, the validator maintains consensus among the network nodes to add blocks to each node's version. This process ensures that the Blockchain remains consistent and secure across the entire network. One of the key features of the validator node is its

dynamic consensus algorithm, which can be modified without the need to re-initialize the Blockchain network or restart the node. This flexibility is achieved through the implementation of the consensus algorithm as a separate module, operating as an independent process. As a result, the validator can seamlessly adapt to changing consensus requirements and ensure the continuous operation of the Blockchain network. In addition to its consensus-related functions, the validator node also plays a crucial role in managing communication between clients, transaction processors (smart contracts), and other validator nodes. Each validator node maintains its own instance of the Blockchain and interacts with other validators using a P2P network [125]. This decentralized communication mechanism ensures the integrity and resilience of the Blockchain network, enhancing its overall security and efficiency.

The Application

outlines the definition of permissible transactions or operations on the blockchain, encompassing:

- *Data model:* which establishes the operations allowed and specifies the content of transaction payloads.
- *Transaction Processor:* responsible for articulating the application's business logic [125], validating transaction batches, and modifying the blockchain state in accordance with the application's defined rules.
- *Client:* outlining the logic governing the application's client-side operations, encompassing transaction generation, submission to the validator, and displaying blockchain data.
- *REST API:* serving as the communication interface between the client and the transaction processor.

Blockchain application

The blockchain application plays a vital role in specifying the permitted transactions or operations within the blockchain network. It encompasses several key components:

- *Data model:* This component defines the applicable operations and specifies the payload of each transaction. It essentially outlines the structure and format of data that can be stored and manipulated on the blockchain.
- *Transaction Processor:* The transaction processor is responsible for defining the application's business logic [125].Based on the rules and regulations established by the application, it verifies batches of transactions and updates the blockchain's state. By doing this, it is made sure that only legitimate and authorized transactions are handled and recorded on the blockchain.

- *Client:* The client component defines the application's client logic. It is responsible for generating transactions and sending them to the validator for processing and inclusion in the blockchain. The client also facilitates the display of blockchain data to end-users, enabling them to interact with the blockchain network.

- *REST API:* The REST API provides a communication interface between the client and the transaction processor. It allows the client to interact with the transaction processor, submit transactions, and retrieve information from the blockchain. This API serves as a bridge that enables seamless communication and data exchange between the client and the blockchain application [125].

By integrating these components, the blockchain application governs the behavior and functionality of the blockchain network, enabling secure, transparent, and efficient data operations on the distributed ledger.

4.3 System architecture perspectives

This section discusses the security, networking, and user perspectives of our proposed architecture.

4.3.1 Security and privacy perspective

The security perspective of our proposed architecture, concerning security principles that should be enforced to achieve a secure framework for the users, devices, networks, processes, and data includes:

- *Confidentiality* is a critical aspect of our architecture, ensuring that data remains secure and accessible only to authorized users. To achieve this, we implement various measures: a) **Authentication and Data Encryption for IoT Devices:** Upon powering on, each IoT device must undergo authentication on the network before collecting or transmitting data. IoT devices that support cryptography encrypt all data before transmitting it to the fog layer. Data encryption tasks are handled by access points or microcontrollers for less capable devices. b) **Authentication and Secure Communication for Fog Nodes:** Fog nodes collaborate to efficiently manage network resources and execute specific tasks. Before collaboration can begin, each fog node undergoes authentication on the network and secures the communication channel, guaranteeing the integrity/confidentiality of data exchanged between them [129]. c) **Secure Communication Channel for SDN Controller:** The OpenFlow protocol is securely run on TLS, providing encryption and authentication for fog nodes communicating with the SDN controller. This ensures that the communication between fog nodes and the controller remains secure. d) **Authentication and Encryption in the Blockchain Network:** A distinct pair of keys is held by each Blockchain node or client and is used for data encryption and authentication. Only peers who have successfully authenticated are permitted

to use our architecture's permissioned network to access the system. As a result, the Blockchain network maintains the integrity and confidentiality of data.

- *Integrity:* A crucial aspect is to ensure the integrity of the data; that the data originates from the legitimate sender and that the data is not altered during the transmission process either intentionally or unintentionally. our architecture ensures data integrity by a) As we said before a device connects to the network it has first to authenticate itself, the fog node will check the integrity of data based on the device identity. b) To avoid tampering with OpenFlow rules, the SDN controller cluster will synchronize every rule to the blockchain network to protect the fog node against rules forgery and misrouting. c) Blockchain members will sign every transaction with their private key, so everyone in the network will know the source of the transaction.

- *Availability:* Users are expected to have all the data at their disposal anytime they need it. our architecture ensures availability by a) Accessing IoT data from the fog layer reduces the risque of losing access to data if the cloud platform or the path to it is down. b) Distributed SDN clusters are looking like one single entity for the fog nodes, which means even if one SDN controller goes down, rules can still be accessible for all the fog nodes.

4.3.2 Networking perspective

The networking perspective of our proposed architecture, concerning the reliability of the network model, includes:

- *Quality of Service (QoS):* is usually expressed as the capacity of a network to deliver the required services for specified network traffic, such as bandwidth, delay, jitter, loss, and location awareness. Fog computing would help to address various Cloud computing Limitations in meeting QoS requirements, especially latency since the fog layer is closer to the data origins [130]. The characteristics of the SDN, including global network overview and flow management, enable it to provide QoS for applications more simply and flexibly than traditional network architectures [131].

- *Scalability:* is assessed in terms of performance measures, i.e. how a given performance measure varies as we scale the network up and down. One of the advantages of the implementation of SDN is the scalability that brings to the network. Hence, the deployment of the virtual gateway in the fog layer and its connection is straightforward. The difference in scalability is remarkable between SDN-enabled networks and traditional networks, where resources need to be configured manually.

4.3.3 Users perspective

The users' perspective of our proposed architecture, regarding the users' experiences of the system and network models. This includes:

- *Real-time interactions:* With Fog Computing, users can examine highly sensitive data at the edge of the network, close to where it is generated [132].
- *Transparency:* Using blockchain, network participants can keep track of pertinent data for more efficient supply chain management [128]. By ensuring that this information is present throughout the supply chain, losses brought on by gray market and counterfeiting are reduced and the traceability of products is improved.
- *Cost:* the overall system can be deployed using open-source technologies, meaning that it could reduce the time and cost required for its implementation.

4.4 Performance evaluation

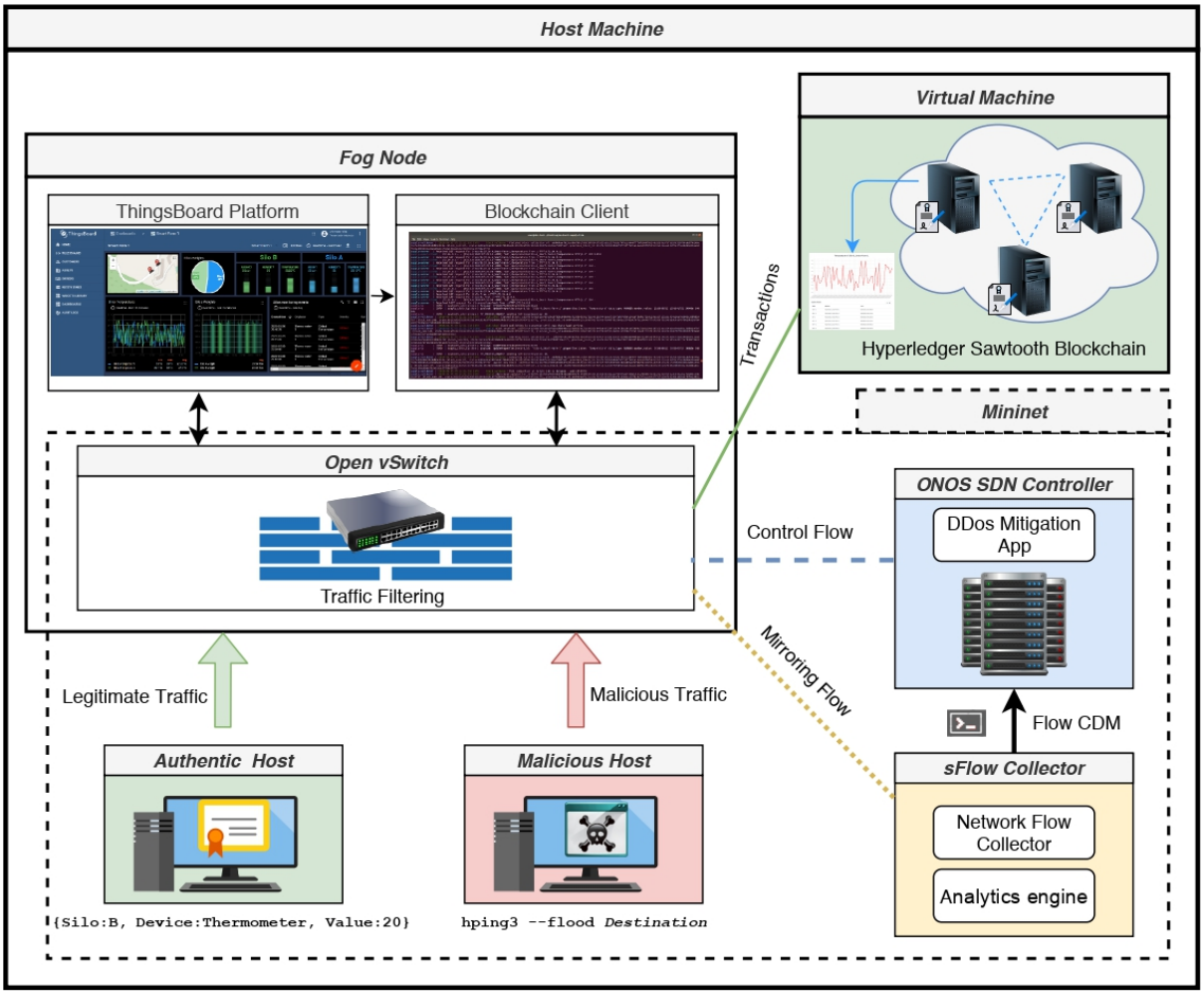


Figure 4.2: Testbed Architecture

DDoS attacks can have a significant impact on both availability and integrity, depending on their targets and objectives. In the context of a Blockchain network, DDoS attacks can be particularly harmful if they target validator nodes. In public Blockchain systems, the distributed design provides some protection against DDoS attacks. For instance, targeted Flooding attacks that overload a single node do not directly affect other

nodes in the network. Once the attack subsides, the targeted node can reconnect to the network and catch up with any missed blocks.

However, the situation is different in private Blockchain networks with a reduced number of nodes in a confined network segment. Even though they may be partially decentralized, their implementation in a single network segment increases the vulnerability to successful DDoS attacks. To address this concern, our architecture aims to test its resilience against DDoS attacks in such scenarios.

This section discusses the deployment cases and presents experimental results of testing the architecture under DDoS attacks. By simulating and analyzing different DDoS attack scenarios, we can evaluate the effectiveness of our security measures and assess the system's robustness in the face of such threats.

	Host Machine	***Virtual Machine***
CPU	Intel Core i5-6300U @ 2.4GHz 2.5GHz, 4 lCores	2 vCores
RAM	8 GB	2 GB
OS	Ubuntu 18.04 x86_64 kernel 5.3.0-53	
Virtualization	- Docker v.19.03.11 - Docker-compose v.1.25.5	
Applications	- ThingsBoard v2.4.3 - Open vSwitch v2.9.5 - Mininet v2.2.2 - ONOS v2.5.0 - sFlow v3.0 - Hping v3.0.0	Hyperledger Sawtooth Supply Chain v0.10.4

Table 4.1: Testbed environment

4.4.1 Test-bed components

As an illustration of the practical application of our proposed architecture, In the experiment, we used the settings shown in Table. 4.1. The test-bed architecture used in the experiment is depicted in Figure 4.2. The experiment attempted to assess our architecture's performance and efficiency in various scenarios and configurations. We tested the system's response to various DDoS attack intensities, different numbers of fog nodes, and varying levels of network traffic. By analyzing the results, we gained insights into the architecture's ability to handle real-world challenges and its resilience in the face of potential attacks. In our experimental setup, we deployed the open-source IoT platform, *ThingsBoard*[2], on the host machine to support fog layer deployment and seamless synchronization with the cloud. The smart farm use case consists of two silos, each containing

[2]https://thingsboard.io/

multiple IoT devices. These devices report pseudo-random values every 1 second. To emulate the Blockchain client, we used a script that receives the reported values from the IoT devices and creates transactions based on them. These transactions are then committed to the Blockchain network using *Open vSwitch*, which we implemented using the *Mininet*[3]. For network management and control, we installed the *ONOS* SDN controller and the *sFlow*[4] collector, linking them to the Mininet virtual network. We installed the *Supply Chain AssetTrack*[5] application and set up the *Hyperledger Sawtooth* Blockchain network in the virtual machine. To keep track of their updated data, we created users for our host machine (the fog node) and various assets. Each reported value is used by the blockchain client to update the targeted asset in the blockchain network. We were able to assess the functionality and performance of our suggested design using this experimental arrangement.

By simulating the smart farm environment and testing different scenarios, we were able to assess the system's ability to handle data collection, blockchain synchronization, and network management effectively and securely. The experiment provided valuable insights into the usability and effectiveness of our proposed architecture for securing agricultural IoT networks.

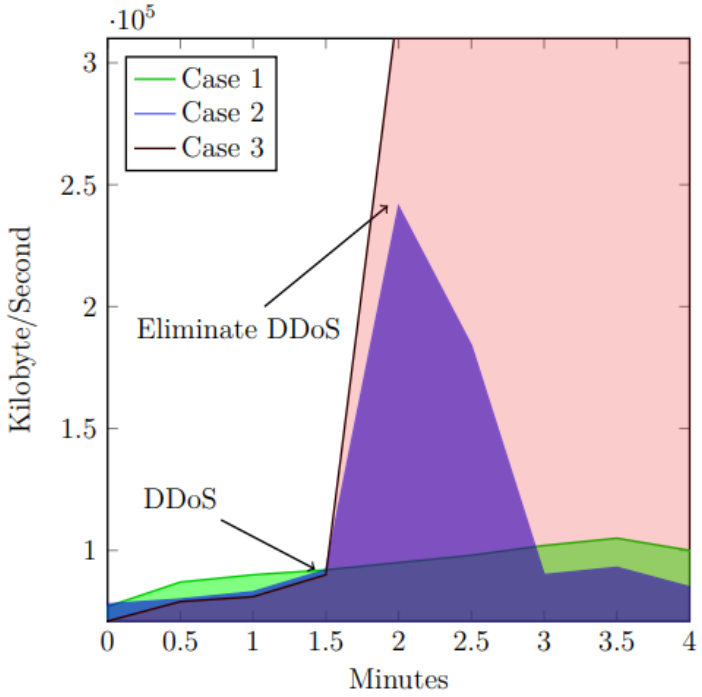


Figure 4.3: Packets received by the Blockchain

4.4.2 Case studies and experimental results

In our evaluation, we conducted three case studies to thoroughly test the functionality and resilience of our framework architecture

[3]https://mininet.org/
[4]https://sflow.com/
[5]https://https://sawtooth.hyperledger.org/docs/supply-chain/nightly/master/

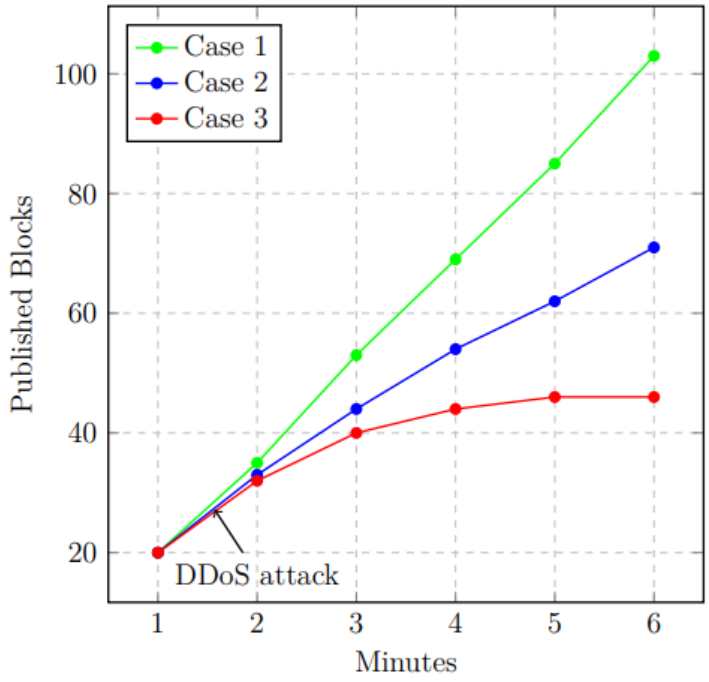


Figure 4.4: Number of published blocks to the blockchain

- ***Case 1:*** This case involved evaluating the regular workflow of the platform without any deliberate attacks on the Blockchain network. We aimed to assess the system's performance under normal operating conditions, including data collection, synchronization with the Blockchain, and network management.

- ***Case 2:*** In this instance, we used the *Hping* program on the host PC to deliberately launch a DDoS assault against the Blockchain network. We turned on a DDoS mitigation application on the SDN controller to lessen the impact of the DDoS attack. The goal was to assess the efficiency of our DDoS mitigation method and its effects on system security and performance.

- ***Case 3:*** We repeated the DDoS assault on the Blockchain network in this instance. This time, however, we turned off the DDoS mitigation software on the SDN controller. The intention was to observe how the system would behave in the absence of any security precautions during a DDoS attack and to assess how it would affect the overall performance and security of the network.

During the simulation of each case study, we closely monitored two critical metrics to evaluate the system's performance and behavior: 1) **Number of Network Packets Received by the Blockchain Every Second:** This metric reflects the rate at which the Blockchain network receives data packets from the fog nodes. It helps us assess the data processing capacity and efficiency of the Blockchain network under different scenarios. 2) **Number of Blocks Published in the Blockchain Network:** This metric represents the frequency at which blocks are added to the Blockchain network. It indicates the overall performance of the Blockchain network in terms of transaction processing and block validation.

The results of these metrics for each case study are presented in **Figure 4.3** (Number of Network Packets) and **Figure 4.4** (Number of Blocks Published) respectively. By analyzing these figures, we can draw meaningful insights into the impact of regular operation, DDoS attack with mitigation, and DDoS attack without mitigation on the performance and resilience of our proposed framework architecture. These metrics provide valuable information about the system's ability to handle data flow and transaction processing under different conditions, aiding in the assessment of its efficiency and robustness in securing agricultural IoT networks.

The experimental results of the three cases provide valuable insights into the performance and effectiveness of our proposed framework architecture under different scenarios. In the first case, where the regular workflow of the platform was evaluated, the system demonstrated its ideal performance, with 103 published blocks in 6 minutes and an average data rate of 90 Kb/sec. This indicates that under normal operating conditions, the architecture efficiently handles data flow and transaction processing, providing a reliable and robust solution for securing agricultural IoT networks. The third case exhibited poor performance, with only 46 published blocks, representing a loss of more than 55% compared to the ideal results of the first case. The network became unavailable due to the DDoS attack, highlighting the vulnerability of private Blockchain networks to such attacks, especially in reduced network segments. This case underscores the importance of implementing effective DDoS mitigation strategies to safeguard the Blockchain network's availability and resilience.

In the second case, a DDoS attack was launched on the Blockchain network, but the SDN controller's DDoS mitigation application quickly intervened, detecting and stopping the attack. As a result, 71 blocks were published during this case, indicating an improvement of about 24% compared to the third case where no mitigation was applied. This demonstrates the efficacy of the DDoS mitigation application in protecting the Blockchain network from such attacks and ensuring a more stable and reliable performance.

Overall, the experimental results shed light on the critical role of DDoS mitigation measures in safeguarding the availability and integrity of the Blockchain network, especially in private Blockchain networks deployed in specific network segments. The combination of Blockchain technology, fog computing, and SDN provides a powerful security architecture for agricultural IoT networks, and proper DDoS mitigation ensures its resilience and ability to handle real-world scenarios effectively.

4.5 Conclusion

In this chapter, we introduced and presented in detail our proposed security architecture tailored for enhancing the security of agricultural IoT networks. We evaluated the proposed security architecture on an open-source IoT platform, which provided a realistic testing environment. The combination of blockchain and SDN technologies demonstrated an overall good performance, showcasing the effectiveness of our solution in securing agricultural IoT networks. The integration of these cutting-edge technologies offers several advantages, such as enhanced data security, real-time data processing, and efficient network management. The experimental results verified the robustness and reliability of our

architecture under various scenarios, including normal operating conditions and DDoS attacks. With the DDoS mitigation application enabled, the architecture effectively protected the blockchain network, preventing significant disruptions and ensuring continued data processing and transaction publishing.

In conclusion, our proposed security architecture offers a comprehensive and practical solution to address the unique challenges faced by agricultural IoT networks. By leveraging blockchain, fog computing, and SDN, we establish a secure, scalable, and adaptable infrastructure that enables smart and efficient agricultural practices. The positive experimental results further reinforce the viability and efficiency of our proposed solution, providing valuable insights for real-world implementation in agricultural IoT environments.

Chapter 5

FL-based IDS for agricultural IoT

5.1 Introduction

The past years have witnessed significant efforts in harnessing technology for smart farming, aimed at enhancing agricultural production in terms of both quality and efficiency [4]. By integrating technologies like IoT and AI, data can be efficiently sensed, collected, and analyzed across diverse conditions and scenarios [133]. This wealth of data can then undergo comprehensive analysis to enable real-time decision-making and foster situation awareness, ultimately leading to improved decision-making processes and higher-quality agricultural output. A crucial aspect of the agricultural sector is the food supply chain, encompassing the entire journey of food products from growers to consumers.Production, storage, processing, distribution, retailing, and consumption are just a few of the stages that make up this journey. A sophisticated network of physical objects with communication capabilities is formed by the interconnection of numerous smart items along this supply chain, enabling the exchange and aggregation of data. This networked environment is referred to as the agricultural IoT (Agri-IoT), a key technology advancing smart agriculture [4].

Specific qualities, such as extensibility, reliability, heterogeneity, security, and privacy, are present in agri-IoT networks and applications. Conventional systems and network architectures often fall short in meeting these requirements, leading the industry to explore emerging technologies like edge computing, Blockchain, SDN, and more [4]. However, these technologies come with potential security flaws, and leveraging these vulnerabilities can result in severe consequences. For instance, the NotPetya malware attack in 2017 caused substantial damage to Maersk, a prominent shipping company, with estimated losses of hundreds of millions of dollars [109]. Similarly, in 2021, JBS, the largest meat supplier globally, suffered a ransomware attack that disrupted operations in multiple countries and impacted thousands of workers[1].

The cybersecurity industry is actively looking for quick-response solutions to protect against such threats and reduce financial risks. Use of ML-based solutions, notably anomaly-based IDS, is one strategy gaining popularity. However, conventional ML techniques often require centralizing learning data on a single machine or cloud platform, lead-

[1]https://www.bbc.com/news/world-us-canada-57318965

ing to concerns about data privacy [134], high communication overhead, and increased resources consumption. FL emerges as a promising alternative, enabling knowledge sharing while preserving privacy and reducing costs. This approach holds the potential to revolutionize security practices in Agri-IoT networks and foster more efficient and secure data analysis.

In this chapter, we introduce FELIDS, an innovative FL-based IDS designed to enhance the security of agricultural-IoT infrastructures. The FELIDS system addresses data privacy concerns by adopting a local learning approach, where IoT devices collaborate by sharing model updates with an aggregation server. This process enables the creation of an improved detection model without compromising the privacy of individual device data.

Through a series of experiments, we demonstrate that the FELIDS system surpasses conventional centralized ML methods in safeguarding the privacy of IoT devices' data while achieving high accuracy in intrusion detection. The results highlight the effectiveness and superiority of FELIDS as a privacy-preserving and robust IDS solution for Agri-IoT environments. By utilizing FL, FELIDS has the potential to revolutionize the security landscape in the agricultural sector and ensure the confidentiality of sensitive data without compromising detection accuracy. This work's main contributions are:

- We introduce FELIDS, a Federated Intrusion Detection System that leverages the power of FL to enhance the security of Agri-IoT environments while preserving data privacy.
- We implement and investigate three deep learning classifiers - DNN, CNN, and RNN - to assess their effectiveness in intrusion detection within the FELIDS framework.
- We evaluate the performance of each classifier using three recent real-world traffic datasets, and provide an in-depth evaluation of the FELIDS model's performances, comparing it with the centralized ML model and state of the art works in the domain. This comparative analysis showcases the superiority and efficacy of FELIDS in addressing cyberthreats while maintaining high accuracy and data privacy.

5.2 The Agri-IoT Landscape

In this section, an overview of the Agri-IoT environment is given, with a description of its structure and highlighting the threat model.

5.2.1 Architecture description

The Agri-IoT framework, depicted in Fig 5.1, represents the widespread adoption of emerging technologies in Agriculture 4.0. It involves both high-level and low-level architectures, seamlessly fusing robots, SDN, blockchain, and smart sensors. The framework as displayed integrates a number of technologies, including blockchain, smart sensors, robots, and SDN. These technologies attempt to give the agricultural industry the tools it needs to assist its automation and decision-making processes by offering the ideal mix

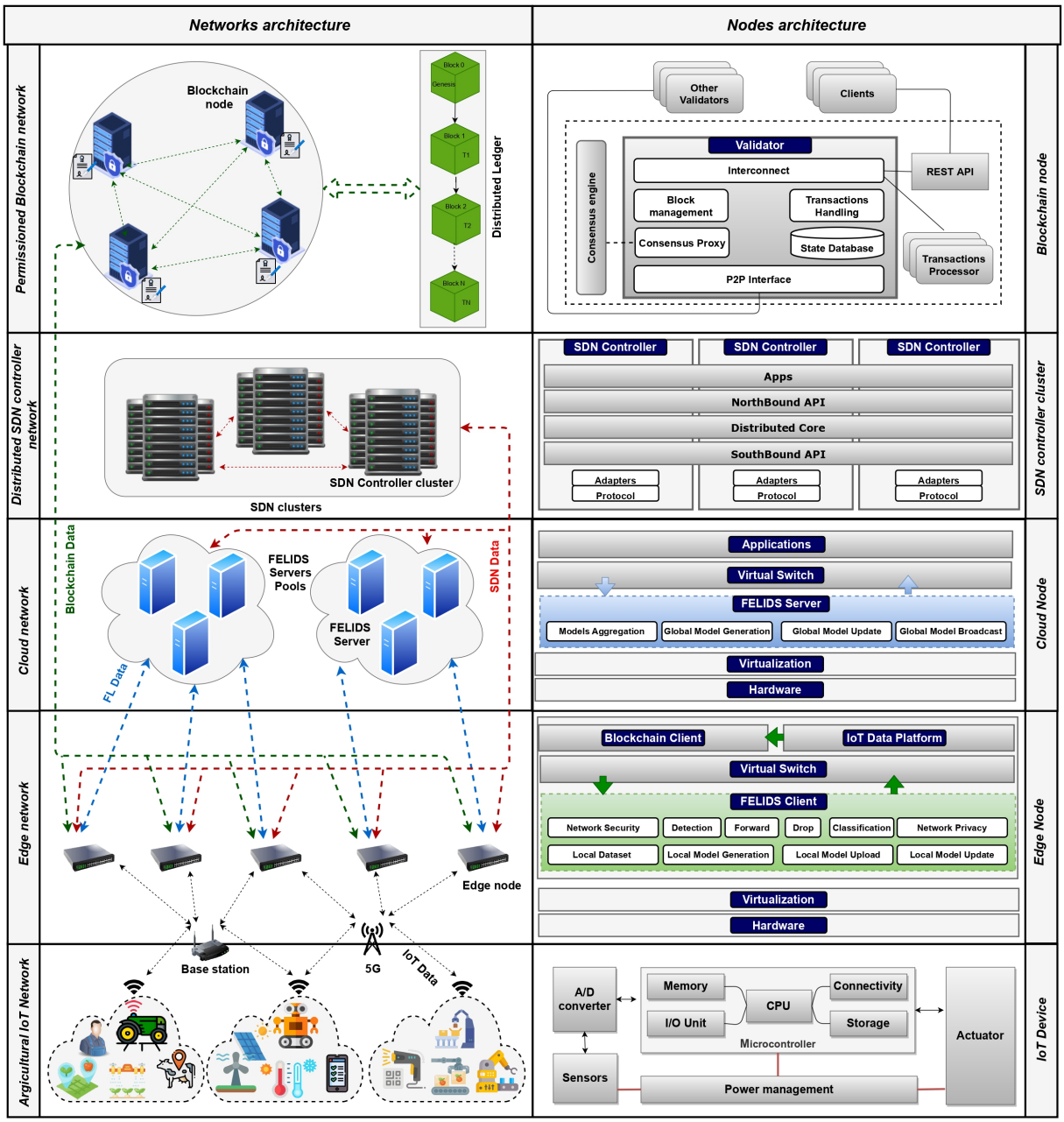


Figure 5.1: Agri-IoT framework architecture

of resources, knowledge, and services, which will increase revenue. Making sure that agricultural data is secure and only accessible to authorized individuals is crucial. To achieve this insurance, authentication and data encryption are necessary. The Agri-IoT framework ensures data privacy and confidentiality by:

- *Agri-IoT layer:* serves as the foundation for data collection and communication in the agricultural IoT framework. To ensure secure and authorized data exchange, each IoT device within this layer must undergo an authentication process before being allowed to collect or transmit data. For crypto-capable IoT devices, an additional layer of security is implemented by encrypting data before reaching the fog layer. This encryption ensures that sensitive information remains confidential and protected from unauthorized access or tampering during transit. In contrast, less

powerful IoT devices without built-in encryption capabilities rely on access points or microcontrollers to handle the encryption task. This ensures that even devices with limited processing capabilities can maintain data security while communicating with the fog layer. By implementing strong authentication and data encryption measures within the Agri-IoT layer, the framework ensures the confidentiality and integrity of agricultural data, preventing unauthorized access to sensitive information.

- *Edge layer:* The *Edge layer* plays a role in the efficient management of network resources and the execution of specific tasks within the agricultural IoT framework. Edge nodes, which are distributed throughout the network, collaborate and work together to achieve these objectives. Before engaging in any collaborative activity, each edge node undergoes a rigorous authentication process to verify its identity and ensure it is a trusted and authorized participant in the network. This authentication step is essential for maintaining the security and integrity of the edge layer. Furthermore, once authentication is successful, a secure communication channel is established between the authenticated edge nodes. This secure channel ensures that all data and information exchanged between the nodes remain confidential and protected from unauthorized access or tampering. By authenticating edge nodes and securing communication channels, the edge layer ensures that network resources are optimally managed, and collaborative tasks are executed efficiently, while also upholding the overall security of the agricultural IoT infrastructure.

- *SDN layer:* The *SDN layer* in the framework leverages OpenFlow, which operates securely on the TLS protocol [25]. This ensures that the communication between the fog nodes and the SDN controller is encrypted and authenticated, safeguarding the integrity and confidentiality of the data and network rules exchanged between them. To handle large-scale systems, the use of an SDN cluster, also known as a distributed SDN controller architecture, is recommended. This approach addresses the potential problems related with a standalone SDN controller, such as the SPOF problem. With a single SDN controller, if it fails, the entire system may become non-functional. However, by deploying multiple controllers in a cluster, the SPOF issue is mitigated, providing enhanced fault tolerance and system resilience. The integration of blockchain technology with the SDN layer is another aspect explored in the Agri-IoT framework. In this context, additional SDN controllers can be utilized as miners in the blockchain network, validating transactions and contributing to the overall efficiency and scalability of the system, as demonstrated in prior work by Barka *et al.* [135]. This integration further enhances the performance and security of the Agri-IoT infrastructure.

- *Blockchain layer:* The permissioned network used in the design of this layer of the framework allows only known peers to interact with it. Each Blockchain node or client has a set of keys that allow for data encryption and authentication. This approach ensures secure and private communication within the network, as unauthorized entities are restricted from accessing the system. The use of blockchain in the Agri-IoT framework offers various advantages, especially in terms of supply chain management. By leveraging blockchain, network members can track and verify relevant information, enhancing the traceability of materials and mitigat-

ing losses due to counterfeiting and gray market activities. This transparency and authenticity in the supply chain contribute to improved efficiency and security in agricultural production and distribution processes. Additionally, blockchain's capabilities extend to creating multinational Industrial IoT (IIoT) platforms. This integration can effectively mitigate threats such as DoS/DDoS attacks, message tampering attempts, and authentication delays, as proposed in the work by Rathee *et al.* [136]. These security enhancements further strengthen the overall resilience of the Agri-IoT infrastructure against various cyber threats.

The implementation of encryption and authentication is crucial to ensure secure communications and maintain data confidentiality in the Agri-IoT network. However, these measures alone may not be sufficient to protect against cybersecurity attacks initiated by authorized internal parties who misuse their privileges. In such scenarios, a dedicated IDS becomes essential to detect and prevent unauthorized or malicious activities within the network.

FELIDS, is specifically designed to bolster the security of the Agri-IoT implementations against these insider threats. FELIDS leverages federated learning, a privacy-preserving approach, to enable collaborative model training across IoT devices while safeguarding individual data privacy. By adopting this federated learning-based approach, FELIDS can effectively detect and respond to cybersecurity attacks originating from authorized entities within the network.

In the subsequent section, we will present a comprehensive overview of the FELIDS system, detailing its architecture and the mechanisms it employs to ensure robust cybersecurity for the Agri-IoT infrastructure.

5.2.2 Threat model

The adoption of Agriculture 4.0 brings with it several potential threats that could hinder its widespread adoption and acceptance. While some threats have historical origins, such as rough weather conditions, the increased adoption of technology has introduced new security vulnerabilities and critical attack vectors. These threats include:

- *Conventional attacks:* Attackers target the Agri-IoT infrastructure in this class of attacks by taking advantage of the well-known TCP/IP Internet protocol stack. Compromise of authentication mechanisms, traffic snarls, malware exploitation in agri-product manufacturing processes, commercial fraud via the insertion of fake product data, compromise of weak quality control systems, SQL and XSS injections aimed at cloud-based applications, and DDoS attacks on the consortium Blockchain are a few examples of such attacks.

- *IoT protocol-based :* These attacks focus on taking advantage of holes in IoT protocols like MQTT to attack vulnerable IoT devices. Examples include manipulating livestock health-related IoT devices to harm reputations, GPS spoofing, breaching security measures, and tampering with agricultural machinery to create delays or make nefarious judgments in the field.

- *Advanced networks-based attacks:* Attacks on the Agri-IoT framework are carried out by adversaries using SDN-based protocols like OpenFlow. This category covers assaults on the OpenFlow protocol itself, DDoS assaults, and the rerouting of SDN-enabled switches or controllers to bring about network disruptions that interfere with Agri-IoT activities.

These threats pose significant challenges to the security and reliability of the Agri-IoT network, emphasizing the need for robust intrusion detection mechanisms to counteract and prevent potential cybersecurity breaches.

5.3 FELIDS: An IDS for privacy-preserving

In this section, we give a thorough and in-depth explanation of the solution that has been suggested, highlighting its design objectives, the FL-based methodology that has been employed, plus the system design.

5.3.1 Defence Goals

An adversary *mathcalA* can be external or internal. An external adversary is a remote entity that launches cyberattacks from the internet with the goal of disrupting the Agri-IoT network, abusing applications that can be accessed via the internet.Agri-IoT insiders, on the other hand, operate within the network of the Agri-IoT and can be considered an adversary. This could entail a compromised IoT device or another networked entity launching assaults, finding, and taking advantage of insecure IoT services and devices. The primary goal is to effectively detect and counter both known and novel cyberattacks, regardless of their origin from external or internal entities within the smart agricultural ecosystem. Swift identification of these attacks enables timely and appropriate response measures to be taken, mitigating potential damages and enhancing network security. Additionally, we make the following assumptions:

- ***Edge nodes are secure:*** In Agri-IoT infrastructures, edge nodes serve as the network's security gatekeepers, therefore we presumptively none of them are vulnerable. As their breach would result in a breakdown in the security offered by the Agri-IoT network, it is essential to maintain the integrity and security of these nodes.
- ***FL aggregators are trustworthy:*** We trust the FL aggregators, which are the servers responsible for coordinating the training process and aggregating model updates from IoT devices. The level of trust in these servers is essential as they play a critical role in the learning process.
- ***No default malicious devices:*** We assume that IoT devices do not come with pre-existing malicious components or vulnerabilities when initially released by manufacturers. Devices generate only valid communications because there are no default harmful parts, which enables FELIDS to gain insight from benign patterns before any possible adversary *mathcalA* finds and exploits weaknesses.

These assumptions are essential for the effective functioning of FELIDS and contribute to maintaining the security and trustworthiness of the smart agricultural network. Data is essential at every stage of the food chain and is included in the value chain of smart agriculture, which includes numerous stakeholders with varying operational functions. However, the majority of this data is private and cannot be centralized in order to properly train an IDS based on machine learning to provide a reliable anomaly detection model. An effective IDS model should also be able to learn and distinguish between good and bad behavioral patterns. However, attaining this goal becomes more difficult because to the wide variety of IoT device kinds and applications, as well as problems with network latency, constrained processing and storage capacity, and other difficulties. Our target is to create an IDS for Agriculture 4.0 that is decentralized and distributed and can take advantage of the important insights found in Agri-IoT networks without sacrificing data privacy or cost effectiveness. Our IDS can effectively learn from data gathered at the edge nodes without the requirement for centralized by using a federated learning technique. This ensures that data remains local and private, while still contributing to the collective knowledge of the IDS. With this approach, we aim to overcome the challenges associated with traditional centralized ML-based IDS systems in the context of smart agriculture, enabling a more efficient and privacy-preserving solution.

5.3.2 Preliminaries

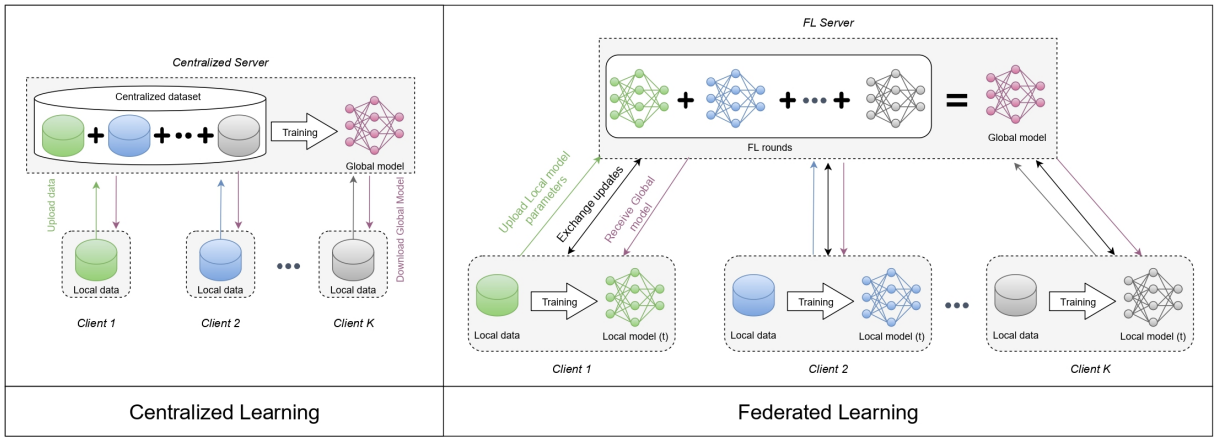


Figure 5.2: Centralized vs. federated learning approaches

Federated DL

FL is especially effective for issues that fit a specific profile. These qualities include 1) circumstances in which training on actual data from dispersed devices offers a benefit across data that is centrally maintained in a single location, 2) situations in which data is private or extensive, and 3) circumstances in which marking data can be acquired from dispersed devices for supervised tasks [54]. Since our work satisfies these requirements, FL is the best strategy. Every client submits its input to the server to train the IDS model in a centralized learning environment, as demonstrated in Fig. 5.2. However, in FL the model is not trained and evaluated on a single machine, however, all K clients create

a local model with the same structure that is trained on various local datasets. After being shared with an aggregation server, these local models are combined to produce an enhanced global model with optimum parameters. This decentralized and collaborative learning process allows the IDS to learn from diverse data sources without compromising data privacy and enables efficient training with distributed resources.

DL Classifiers

The rapid advancements in DL have ushered in a new era of ML [23]. DL has proven to be effective in data representations extracting, leading to the development of more efficient and intelligent IDS technology. The ability of DL to uncover complex patterns and relationships in data makes it a powerful tool for creating sophisticated models that can accurately detect anomalies and cyber threats in Agri-IoT networks. In neural networks, there are fundamental components such as neurons, weights, biases, and activation functions. The primary purpose of a neural network is to associate an input with an output, denoted by x and y, respectively. This is represented by the function $f(x, \theta)$, given that θ is the vector of parameters. Each input x is a data record represented as a vector in $\mathbb{R}^d$, with d being the number of input features. In supervised learning, each input x is linked with a label y, indicating whether the record represents normal (benign) communication or an intrusion.

An artificial neuron, denoted as f_j, takes the input vector x, a set of weights $w_j = (w_{j,1}, ..., w_{j,d})$, a bias b_j, and an activation function $g(.)$ (as shown in Eq. 5.1). The neuron performs the weighted input features' sum, adds the bias, and applies the activation function (Eq. 5.2) to produce the output.

$$f_j(x) = g(< w_j, x > + b_j) \tag{5.1}$$

$$g(x) = max(0, x) \tag{5.2}$$

As a result of our multi-class classification, the output layer is a SoftMax layer that contains a neuron per class (i) and predicts $\mathbb{P}(Y = i/X)$. These values add up to a total of 1. The SoftMax function (Eq. 5.3) evaluates the probability for each class in the manner shown below in order to determine the probability distribution between the classes:

$$SoftMax(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{5.3}$$

In FELIDS, the loss function used is *categorical_crossentropy*, which is commonly used for multi-classification. The Adam optimizer is used [137] along with error backpropagation, which updates the model's weights in the training stage. To prevent overfitting, FELIDS employs two techniques: dropout and L_2 regularization. We used three popular deep learning techniques, namely: DNN, CNN, and RNN, which are formulated as follows:

Table 5.1 summarizes the used values for different classifiers parameters in the FELIDS scheme.

Classifier	Parameter	Value
DNN	Hidden nodes	64-80
	Hidden layers	1-2
	Dropout	0.1-0.4
CNN	Convolutional layers	2-3 Conv1D
	Filters	16-74
	Kernel size	3
	Pooling layers	1 *Global Average Pooling 1D*
	Hidden nodes	120-130
	Hidden layers	2-3
	Dropout	0.1-0.4
RNN	Hidden nodes	20-80
	Hidden LSTM layers	2
	Dropout	0.2
Global	Batch size	100
	Local epochs	1
	Global epochs	50
	Learning rate	0.01-0.5
	Regularization	L_2
	Loss function	*categorical_crossentropy*
	Activation function	*ReLu*
	Classification function	*SoftMax*
	Optimizer	*Adam*

Table 5.1: DL classifiers settings

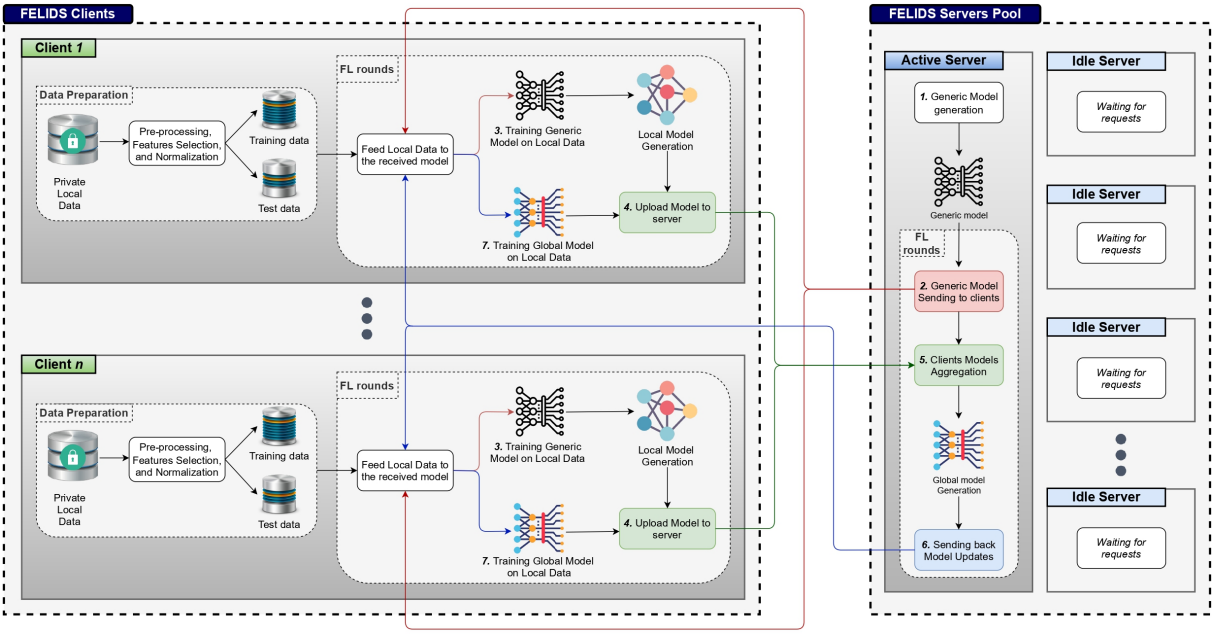


Figure 5.3: FELIDS architecture

5.3.3 Training stage

The server initially chooses C portions of K nodes at the edge (FELIDS clients) to take part in the FL process and carry out calculation for R FL cycles. A secure gRPC channel is used for client-server connections since all data exchanged must be encrypted. To authenticate both clients and servers and to encrypt all of their connection, gRPC features

built-in SSL/TLS capability[2].

At the beginning of the FL process in FELIDS, a fraction C of K edge nodes are chosen by the server to participate in the computation for R FL iteration. Through a secure gRPC channel, clients and servers communicate with one other, ensuring that all data exchanged during the FL process is encrypted[3]. gRPC is a modern and efficient framework for remote communication that supports various programming languages and platforms. It allows clients and servers to communicate over the network transparently, as if they were making local function calls. In the context of FELIDS, gRPC provides the means for secure communication between the clients and the server. The use of a secure gRPC channel in FELIDS ensures data privacy and integrity during the FL process. All communications are encrypted using SSL/TLS protocols, which provide authentication and encryption for both the clients and the server. This means that only authenticated clients can communicate with the server, and all data exchanged between them is protected from eavesdropping and tampering. By employing gRPC with built-in SSL/TLS support, FELIDS ensures that the FL process is conducted securely and that sensitive data remains protected throughout the collaboration between the clients and the server. This is essential in preserving data privacy and maintaining the overall security of the Agri-IoT infrastructure.

After all the selected clients are connected, the training follows the steps illustrated in Fig. 5.3. The FL process is also presented in Algorithm 5 and Algorithm 6, which are adapted from [54], such as:

Algorithm 5: FELIDS server

```
1  Algorithm StartServer(K, C, R)
3      while K ≠ length(ConnectedClients()) do
4          loop
5      end
7      FedAvg()
9      ReleaseClients()
1  Procedure FedAvg()
1  2   w_1 ← GenericModel()
   3   for t = 1, .., R do
   4       S_t ← Subset(max(C · K, 1), "random")
   5       Parallel.for k ∈ S_t do
   6           w^k_{t+1} ← Client_k.FedAvg(w_t) /* run FELIDS client      */
   7       end
   8       w_{t+1} ← Σ_{k=1}^{K} (n_k/n) w^k_{t+1}
   9   end
```

- *Step 1:* At $t = 0$, A basic neural network model is started on the FELIDS server with a set of random initial weights, denoted by w. The NN, thelocal and global epochs are defined at this stage.

[2]https://grpc.io/docs/guides/auth/
[3]https://grpc.io/docs/guides/auth/

- *Step 2:* Each of the K FELIDS clients gets that model from the server.

- *Step 3:* Each FELIDS client k, where $k \in [1, .., K]$, re-trains the model in local using its private data. This results in a new local set of weights w_{t+1}^{k} for the updated local model. The training process includes multiple local epochs E, where each local epoch uses a minibatch b with size B. The weights are updated using the average gradient $\nabla f(w, b)$ and learning rate η.

- *Step 4:* Only the updated model parameters that have been established using local data are shared by the clients.

- *Step 5:* The server aggregates the received weights from the clients to create a new updated global model. The aggregation is performed by weighting the average of the local models, given that the weights are based on the number of local examples n_k for each client k.

- *Step 6:* The FELIDS server communicate back the new model parameters to all clients.

- *Step 7:* Each client further enhances the received model using its local data.

Steps 4 to 7 are repeated for ongoing learning and enhancement of the global model. The FELIDS clients regularly update their local models with private data, which also allows them to communicate model improvements to the server.

Algorithm 6: FELIDS client

```
Algorithm StartClient(D, B, E)
    P ← PreProcess(D)
    while ServerConnect() do
        FedAvg(w ← FetchParams())
    end
    SaveParams(w)
Function FedAvg(w)
    B ← Split(P, B)
    for i = 1,..,E do
        for b ∈ B do
            w ← w − η∇f(w, b)
        end
    end
    return w to Server
```

5.3.4 FELIDS complexity

Due to limited resources, such as processing power and energy supply, the effectiveness of FELIDS in the edge layer is of utmost importance. The system was created with a focus on minimal complexities and power consumption to address this.

Time complexity

Temporal complexity of the final model is dependent on the temporal complexity of the clients, the aggregation server's temporal complexity as well as the complexity of the parameters exchanged, excluding transmission times, as these factors generally vary significantly from one network to another [138]. The time complexity analysis of FELIDS involves several components. For the clients, the training complexity depends on factors like the number of local epochs (E), the number of local examples (n_k), and the sizes of the layers in the neural network. The overall complexity for the clients can be expressed as $\mathcal{O}(E.n_k.(l_0.l_1 + l_1.l_2 + .. + l_L.l_{L+1}))$, where l_x represents the size of layer x. Regarding the server, its time complexity is determined by K and W. The server's complexity is $\mathcal{O}(K.W)$. Additionally, the complexity of the communicated parameters is directly related to W, resulting in a complexity of $\mathcal{O}(W)$. The time complexity is given by Eq. (5.4), as follows:

$$\mathcal{O}(FELIDS) = \mathcal{O}(K.(E.n_k.(l_0.l_1 + l_1.l_2 + .. + l_L.l_{L+1}))) + \mathcal{O}(K.W) + \mathcal{O}(W) \tag{5.4}$$

Furthermore keep in mind that in order to represent the worst scenario, we have chosen to compute the complexity using all clients (k) as opposed to the subset of clients chosen by the server (S_t).

Energy complexity

FELIDS's analysis of energy complexity takes into account both the aggregation server's (Eq. 5.5) and the individual clients (Eq. 5.6) [139].We consider both the aggregation server's and the clients' energy consumption when calculating the overall training power usage of the global model. FELIDS energy consumption is given in Eq. 5.7.

$$E_{Serv} = \sum_{r=1}^{R} (t_{S_r} e_{S_r}) \tag{5.5}$$

$$E_{Clis} = \sum_{r=1}^{R} \sum_{k=1}^{K} \mathbb{1}_{\{k_r\}} (t_{k_r} e_{k_r}) \tag{5.6}$$

$$E_{FELIDS} = E_{Serv} + E_{Clis} \tag{5.7}$$

In this context, t_{x_r} denotes the time taken by entity x during round r, and e_{y_r} signifies the energy consumption of entity y at the same round. Furthermore, $\mathbb{1}_{\{k_r\}}$ serves as an indicator function, determining whether a client k is selected for participation in FL training during round r.

Dataset	Flow Type	Count	Training	Testing
IDS2018	Benign	+12.6M	123008	30755
	DoS attack-Hulk	466664	36952	9239
	DoS attack-SlowHTTPTest	139890	11191	2798
	DoS attack-GoldenEye	41508	3320	831
	DoS attack-Slowloris	10990	879	220
	DDOS attack-HOIC	686012	54880	13721
	DDOS attack-LOIC-HTTP	576191	46095	11524
	DDOS attack-LOIC-UDP	1730	1384	346
	Brute Force-XSS	187589	183	47
	Brute Force-Web	193360	488	123
	FTP-BruteForce	193354	15468	3867
	SSH-Bruteforce	187589	15007	3752
	SQL Injection	87	69	18
	Infiltration	161934	12851	3213
	Bot	286191	22895	5724
MQTTset	Benign	165463	115824	49639
	DoS	130233	91156	39077
	Brute Force	14501	10150	4351
	Malformed	10924	7646	3278
	SlowITe	9202	6441	2761
	Flood	613	429	184
InSDN	Benign	68424	54739	19627
	DoS	53616	42892	10724
	DDoS	121942	97553	24389
	Probe	98129	78502	19627
	BFA	1405	1124	281
	Web-Attack	192	153	39
	BOTNET	164	131	33
	U2R	17	13	4

Table 5.2: Datasets settings

5.4 Performance Evaluation

The experimental setup is described in this part, and then our application of the FELIDS system is reviewed. We chose the most up-to-date datasets that include some of the newest attack types that can be utilized to target the Agri-IoT framework in order to evaluate the efficacy of FELIDS. To be more specific, we utilized the CSE-CIC-IDS2018, MQTTset, and InSDN datasets [68].

We give a statistical summary for each attack type in each dataset for training and evaluation purposes in the Tab. 5.2, where we show a statistical summary for each attack type.

5.4.1 Experimental setup

For our experiments, we used Google Colaboratory as the execution platform, running Python 3 code. The FELIDS system was implemented using popular libraries such as NumPy for efficient array manipulation, Pandas for data handling and analysis, TensorFlow and Keras for deep learning tasks, and Scikit-learn for various machine learning algorithms. Additionally, SMOTE was employed for oversampling minority classes in the dataset, and the Sherpa.ai FL framework facilitated the Federated Learning process [140]. To monitor FELIDS training energy consumption, we utilized Carbontracker [141]. These libraries provided the necessary tools and functionalities for building, training, and evaluating the FELIDS model, ensuring effective data manipulation, and implementing the Federated Learning approach to secure the Agri-IoT infrastructure.

Objectives

The following are the goals of our experiments: 1) Establish and assess a centralized model having a single location for both the training and test data. 2) Assess the FELIDS system's application as a proof-of-concept. 3) Conduct a comparison of the FELIDS model's performance and privacy findings with those of the centralized model.

5.4.2 Centralized Learning Benchmarking

We used a Centralized Learning (CL) strategy to carry out anomaly identification in this initial set of examinations. As shown in the sections above, we used three different DL algorithms to train a multi-class IDS. The model is developed and trained using data from a single node using the CL approach. The standard detection process used today, in which data is collected and evaluated by a single entity, is reflected by this emulation. In the context of Agriculture 4.0, when sites are spread out over large areas, the central entity may be a trusted security party in charge of compiling and analyzing the data from multiple sources. Table 5.3 presents the Precision, Recall, and F_1-score metrics for multi-class classification alongside the outcomes from a centralized approach employing several DL models. These metrics of performance evaluate how well the model can differentiate between the various classes. Fig. 5.4 illustrates the accuracy achieved by

		Precision			Recall			F_1-Score		
D.	Class	DNN	CNN	RNN	DNN	CNN	RNN	DNN	CNN	RNN
CSECICIDS2018	Benign	91%	92%	92%	99%	99%	99%	95%	95%	95%
	DoS attack-Hulk	100%	100%	100%	100%	100%	100%	100%	100%	100%
	DoS attack-SlowHTTPTest	54%	76%	74%	89%	42%	49%	67%	54%	59%
	DoS attack-GoldenEye	99%	99%	100%	99%	100%	100%	99%	100%	100%
	DoS attack-Slowloris	96%	98%	99%	98%	98%	99%	97%	98%	99%
	DDOS attack-HOIC	100%	100%	100%	100%	100%	100%	100%	100%	100%
	DDOS attack-LOIC-HTTP	100%	100%	100%	100%	100%	100%	100%	100%	100%
	DDOS attack-LOIC-UDP	97%	96%	97%	99%	100%	100%	98%	98%	98%
	Brute Force-XSS	0%	100%	100%	0%	53%	53%	0%	69%	69%
	Brute Force-Web	44%	70%	89%	18%	71%	47%	25%	70%	62%
	FTP-BruteForce	85%	68%	70%	44%	90%	87%	58%	78%	78%
	SSH-Bruteforce	100%	100%	100%	100%	100%	100%	100%	100%	100%
	SQL Injection	0%	83%	89%	0%	56%	44%	0%	67%	59%
	Infilteration	75%	69%	65%	12%	23%	24%	20%	34%	35%
	Bot	100%	100%	100%	100%	100%	100%	100%	100%	100%
MQTTset	Benign	92%	91%	92%	94%	94%	94%	93%	93%	93%
	DoS	91%	91%	91%	89%	89%	89%	90%	90%	90%
	Brute Force	67%	67%	67%	90%	90%	88%	76%	77%	76%
	Malformed	93%	70%	71%	40%	35%	45%	56%	47%	55%
	SlowITe	100%	100%	100%	100%	96%	100%	100%	98%	100%
	Flood	100%	98%	76%	48%	48%	38%	65%	65%	50%
InSDN	Benign	100%	100%	100%	100%	100%	100%	100%	100%	100%
	DoS	98%	98%	99%	99%	100%	98%	99%	99%	98%
	DDoS	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Probe	100%	100%	99%	97%	99%	99%	99%	99%	99%
	BFA	45%	97%	98%	91%	80%	80%	60%	88%	88%
	Web-Attack	71%	40%	36%	100%	100%	97%	83%	57%	53%
	BOTNET	97%	97%	94%	100%	100%	100%	99%	99%	97%
	U2R	33%	75%	50%	50%	75%	50%	40%	75%	50%

Table 5.3: Centralized model evaluation

the deep learning techniques in multi-class classification for three distinct datasets. The InSDN dataset demonstrated the highest accuracies, with DNN with 98.54%,RNN with 97.84%, and CNN with 97.71%. Conversely, the MQTTset dataset exhibited the lowest accuracies, with DNN 90.40%, RNN 90.05% and CNN with 90.76%.

Although the centralized model has shown good performance, it is accompanied by privacy issues, network latency, and other problems that can make the entire system ineffective. These challenges can significantly affect the system's reliability and practicality in real-world scenarios. As the system deals with sensitive data and operates in a distributed IoT environment, ensuring data privacy and security is of utmost importance. Moreover, network latency can hinder timely responses and decision-making, which is critical in applications like agriculture where quick actions can have a substantial impact.

5.4.3 FELIDS Benchmarking

In this stage of the study, we mimicked the FELIDS detection process, which makes use of FL. With this method, edge nodes, or FELIDS customers, work together to exchange knowledge without jeopardizing the confidentiality of their personal information. We set up an environment with one FELIDS server and various sets of FELIDS clients, abbreviated as K, in order to conduct the tests. With $K = 5$ in the first group, $K = 10$ in the subsequent group, and $K = 15$ in the final group, we made three sets of clients. We took into consideration 2 use cases for propagating data across the clients, *Independent*

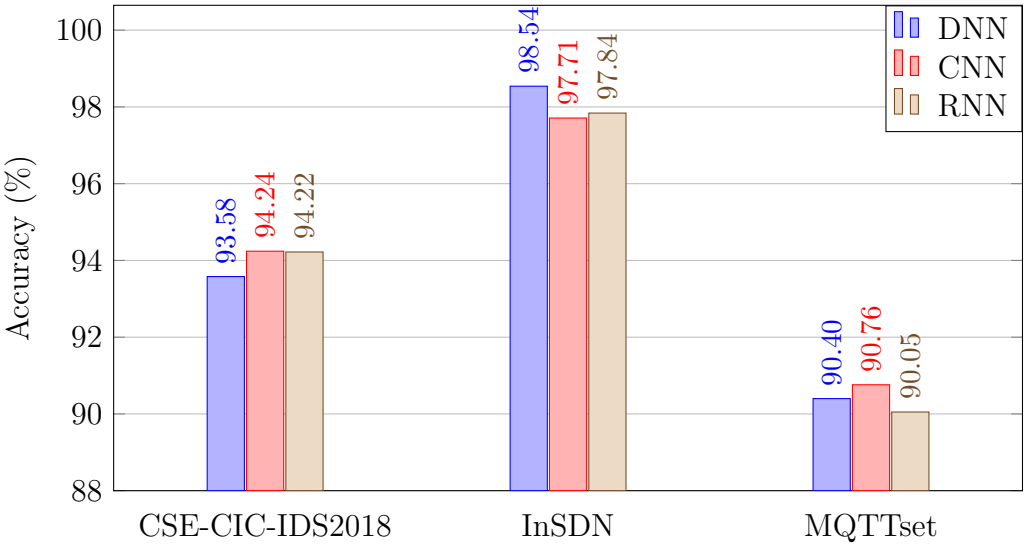


Figure 5.4: Centralized model performance

and Identically Distributed (IID) and *Non-Independent and Identically Distributed (Non-IID)*. The experiments were carried on Ubuntu 20.04.3 LTS with Intel CPU Core i5-6300U @ 2.4GHz, 8.00 GB of RAM.

Each client in the FELIDS configuration has a particular and exclusive partition of the global dataset, therefore simulates exclusive local data. Individual FELIDS system components were also run in segregated contexts to prevent data leaking, making sure that they only connect with the FELIDS server over secure networks. After completing one local epoch during each of the fifty FL rounds, FELIDS clients update the model parameters and send them back to the FELIDS server. Each FL round features active participation from all connected FELIDS clients, emulating the real-world Agri-IoT framework situation where different edge nodes work together to improve the overall model while protecting the privacy of their particular data.

The validation accuracy compared to the centralized model and FELIDS models throughout all three client deployment sets and both data distribution scenarios is shown in Figure 5.5. For the CSE-CIC-IDS2018 dataset, the results of validating accuracies of the centralized model against FELIDS models are shown in Fig. 5.5 (a). The validation accuracy of the centralised model and FELIDS using the InSDN dataset is shown in Fig. 5.5 (b). Lastly, Fig. 5.5 (c) is for the MQTTset dataset. These results offer insightful comparisons of how well the CL strategy and the FELIDS approach perform at finding anomalies across various datasets and model setups.

The results indicate a notable trend where the accuracy of the FELIDS global models improves with the increase in the number of FL rounds. This observation suggests that as FELIDS clients participate in more rounds of collaborative learning, their local models benefit from the knowledge shared in the global model, resulting in performance enhancement across all clients simultaneously.

Furthermore, a significant finding is in certain cases, the global FELIDS models have demonstrated the ability to match or closely have the performance levels of the centralized model. This observation highlights the efficacy of the federated learning approach in

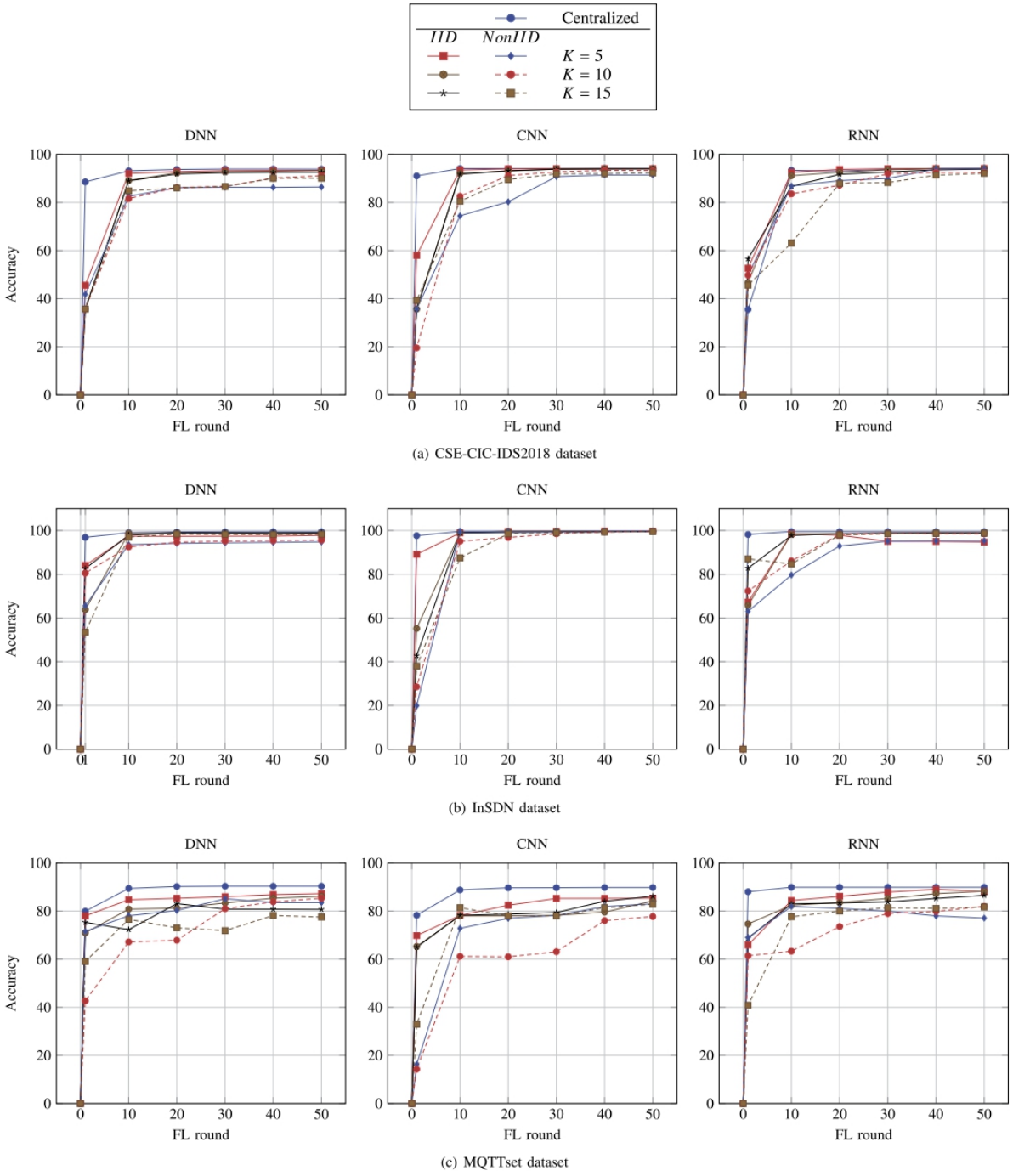
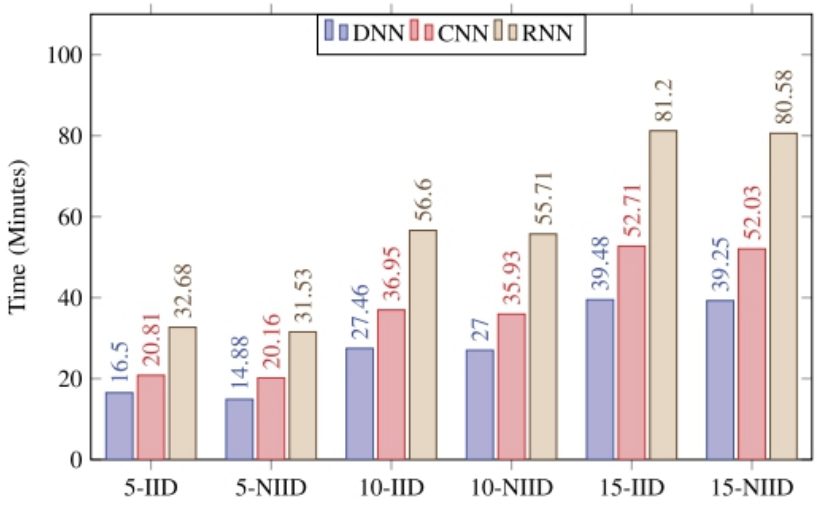


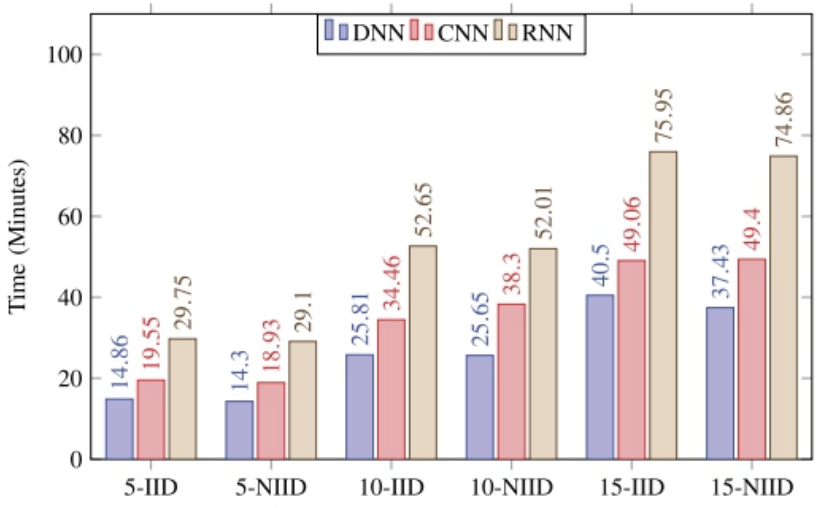
Figure 5.5: FELIDS accuracy over different datasets, neural networks, data distribution techniques, and client sets

anomaly detection, as the distributed and collaborative learning process enables FELIDS clients to collectively improve their models while maintaining the privacy of their local data.

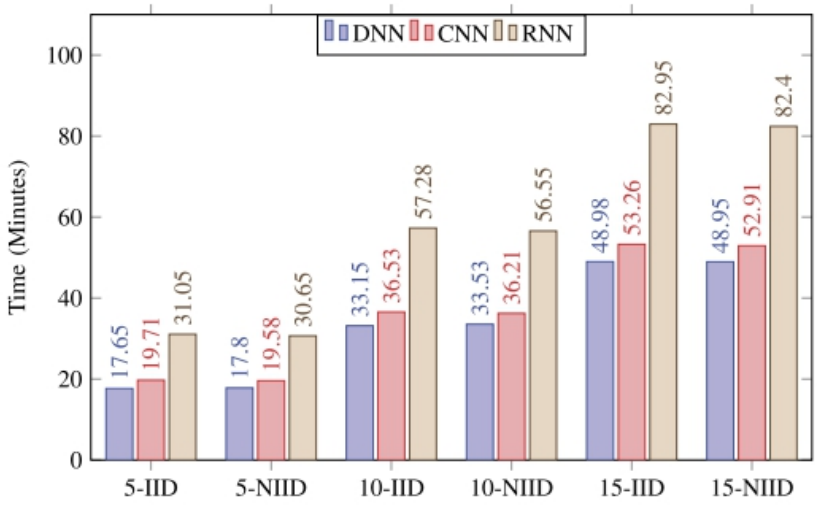
Fig. 5.6 and Fig. 5.7 represent the time and energy consumption of FELIDS after 50 FL rounds of global model training, considering the datasets used, the number of clients (k = [5, 10, 15]), and the data distribution technique (IID or non-IID). Taking into account the datasets employed, the number of clients, and the data distribution technique, Fig.



(a) CSE-CIC-IDS2018 dataset

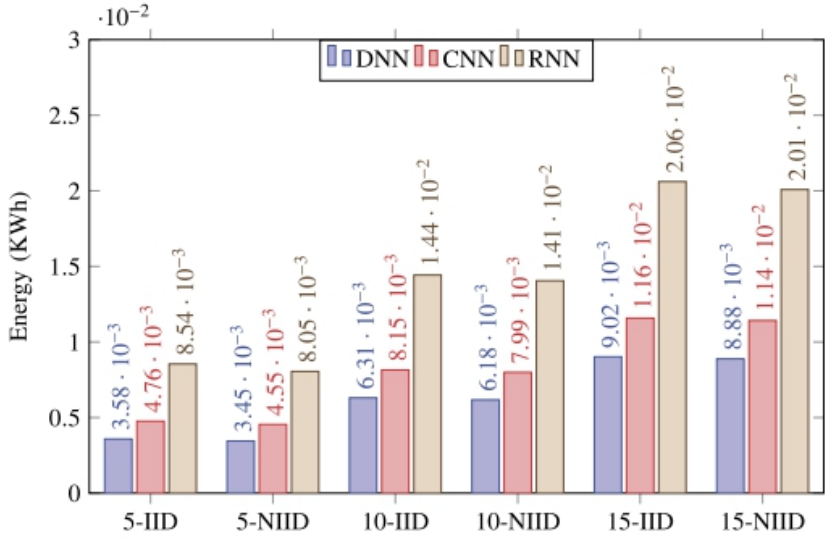


(b) InSDN dataset

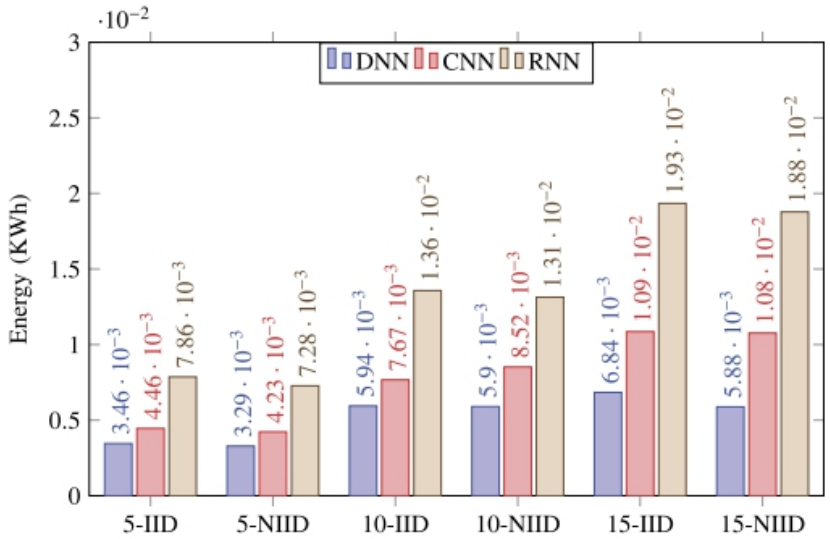


(c) MQTTset dataset

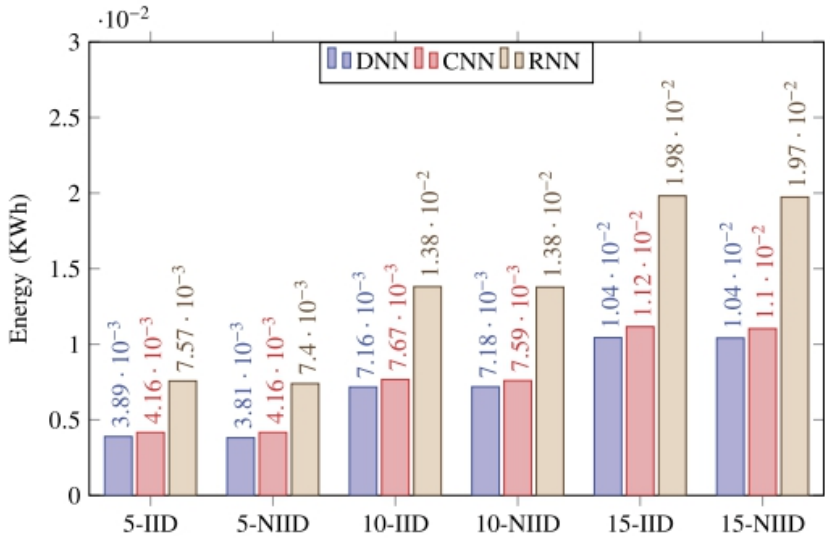
Figure 5.6: FELIDS training time



(a) CSE-CIC-IDS2018 dataset



(b) InSDN dataset



(c) MQTTset dataset

Figure 5.7: FELIDS energy consumption

5.6 and Fig. 5.7 illustrate the time and energy consumption of FELIDS after 50 FL cycles of global model training.

We can see that DNN is the most efficient with the lower time and energy consumption, followed by CNN, whereas RNN comes after, having consumed higher amounts of time and energy. Another observation is that the data distribution technique does not have a large impact on time or energy consumption, although the Non-IID-based experiments usually tend to consume slightly less than their peers with the same number of clients. It is evident that DNN exhibit the highest efficiency with lower time and energy consumption, closely followed by CNN. RNN demonstrate comparatively higher time and energy consumption. Another noteworthy observation is that the data distribution technique employed does not significantly affect time or energy consumption. These findings lead us to the conclusion that the DL algorithm employed and the number of customers taking part in the FL training are the two key determinants of the time and energy consumption, respectively.

The validation accuracy performance for all models, best and worst clients for various client distribution sets, taking into account both IID and non-IID data distributions, is compared in Table 5.4 for the 50^{th} FL round and for various client distribution sets. Due to everyone's access to all classes in the IID data distribution, there is little difference between the weakest and best client's performance. While maintaining client data privacy, the global models performed almost as well as the centralized model. While, the performance gap is considerably wider when non-IID data is distributed, though, because some clients can access fewer classes. For example, the lowest client's performance was only 0.40% while utilizing the InSDN dataset (with the CNN classifier and $K = 10$ clients), while the best client obtained 50.90% accuracy and the global model scored 28.53% accuracy during the initial cycle. The top client achieved an accuracy of 99.38 percent after the completion of all FL rounds, and the entire global model achieved an accuracy of 99.60 percent, which is even better than the results of the centralized model. The worst client, fortunately, managed to able to boost its performance to 66.55 percent. As a result, it is clear that FELIDS can effectively protect client privacy while yet enabling them to gain expertise from their peers without disclosing their raw data.

Table 5.5 presents a comprehensive comparative analysis of the performance FELIDS versus other FL-based IDS approaches. The focus of the comparison is on various key aspects such as the environment of deployment, dataset selected, classifier used, client numbers involved and data distribution techniques adopted.

5.5 Conclusion

In conclusion, FELIDS has demonstrated to be an reliable solution for intrusion detection in Agri-IoT networks. Through federated learning, FELIDS enables collaboration among edge nodes while maintaining data privacy, making it suitable for distributed environments like Agriculture 4.0. The experimental results demonstrated that FELIDS achieved comparable or even better performance than centralized models while maintaining the privacy of clients' data.

We conducted experiments using different deep learning classifiers and real-world

D.	C.	K	First round						After 50 rounds					
			IID			Non IID			IID			Non IID		
			B	W	G	B	W	G	B	W	G	B	W	G
CSECICIDS2018	DNN	5	49.19%	45.39%	45.59%	47.76%	15.92%	41.82%	93.39%	92.72%	93.29%	92.22%	40.67%	86.43%
		10	35.68%	35.68%	35.68%	35.68%	09.88%	35.68%	93.30%	92.49%	93.22%	91.21%	40.70%	91.14%
		15	35.68%	35.68%	35.68%	45.40%	08.15%	35.68%	92.75%	91.95%	92.53%	90.64%	81.00%	90.08%
	CNN	5	57.99%	57.94%	57.98%	44.84%	07.70%	35.68%	94.08%	93.72%	94.08%	87.36%	31.32%	91.42%
		10	35.68%	35.68%	35.68%	43.76%	05.32%	19.54%	94.06%	93.36%	94.02%	92.90%	54.17%	93.29%
		15	35.68%	35.68%	35.68%	53.57%	00.80%	39.27%	94.00%	93.01%	93.97%	92.65%	55.40%	92.29%
	RNN	5	52.66%	52.58%	52.64%	64.28%	13.39%	49.79%	94.18%	94.11%	94.15%	93.98%	78.24%	93.97%
		10	48.60%	46.43%	46.66%	56.78%	11.61%	49.76%	94.07%	94.00%	94.05%	92.49%	75.14%	92.56%
		15	57.68%	56.42%	56.59%	58.10%	03.95%	45.63%	93.90%	93.61%	93.86%	91.55%	89.19%	92.08%
MQTTset	DNN	5	66.99%	66.96%	66.97%	71.78%	44.73%	71.53%	87.24%	86.90%	87.23%	83.76%	54.51%	83.52%
		10	76.31%	70.47%	70.94%	65.27%	37.20%	42.68%	86.19%	85.54%	86.13%	78.55%	42.26%	85.33%
		15	79.12%	63.82%	75.32%	76.21%	02.73%	59.04%	81.13%	80.50%	80.69%	78.03%	20.91%	73.62%
	CNN	5	74.36%	68.27%	69.80%	56.85%	16.23%	16.25%	85.30%	85.29%	85.35%	83.29%	41.88%	83.03%
		10	65.43%	65.08%	65.29%	19.49%	00.85%	14.28%	86.27%	84.01%	84.22%	76.26%	41.53%	77.73%
		15	65.04%	64.75%	64.96%	69.59%	23.09%	32.93%	86.94%	84.25%	86.16%	82.89%	51.68%	82.82%
	RNN	5	66.26%	63.99%	65.89%	64.20%	39.35%	69.00%	89.56%	87.66%	88.13%	81.74%	42.10%	77.07%
		10	75.67%	73.57%	74.64%	50.77%	02.78%	61.43%	87.08%	88.47%	88.25%	83.06%	44.74%	81.95%
		15	76.40%	61.33%	68.71%	68.19%	04.43%	40.85%	86.98%	84.43%	86.53%	82.44%	51.71%	81.71%
InSDN	DNN	5	84.86%	82.14%	84.06%	63.29%	32.51%	65.58%	98.50%	97.39%	97.80%	98.43%	43.34%	94.80%
		10	63.84%	63.84%	63.84%	60.61%	11.24%	80.56%	98.86%	97.47%	98.63%	98.75%	59.72%	95.61%
		15	82.85%	81.52%	82.60%	63.84%	15.59%	53.42%	99.03%	98.71%	98.93%	98.88%	91.14%	98.30%
	CNN	5	89.27%	89.08%	89.11%	78.83%	43.84%	19.89%	99.77%	99.58%	99.73%	99.71%	94.91%	99.02%
		10	55.20%	55.19%	55.20%	50.90%	00.40%	28.53%	99.75%	99.57%	99.74%	99.38%	66.55%	99.60%
		15	43.63%	42.26%	42.77%	65.15%	19.89%	37.93%	99.73%	99.09 %	99.52%	99.34%	63.89%	99.48%
	RNN	5	67.36%	67.21%	67.24%	64.54%	19.48%	63.01%	95.04%	94.14%	94.75%	93.49%	76.18%	95.30%
		10	66.19%	65.34%	65.74%	65.38%	01.09%	72.35%	99.27%	98.71%	99.05%	98.57%	93.13%	98.46%
		15	83.31%	82.46%	82.78%	86.18%	15.59%	87.04%	98.83%	98.12%	98.56%	98.74%	94.18%	98.75%

(**B**): **B**est client accuracy; (**W**): **W**orst client accuracy; (**G**): **G**lobal model accuracy;

Table 5.4: FELIDS performance evaluation

IDS	Year	Deployment	Dataset	Classifier	K	IID	Non IID
Preuveneers et al. [112]	2018	General Purpose	CICIDS2017	Autoencoder	12	✓	✗
Nguyen et al. [97]	2019	IoT Networks	Generated dataset	RNN-GRU	[5, 9, 15]	✗	✓
Schneble et al. [115]	2019	Medical-CPSs	MIMIC	MLP	[2, 4, 8, 16, 32, 64]	N/A	N/A
Rahman et al. [114]	2020	IoT Networks	NSL-KDD	DNN	4	✓	✓
Zhao et al. [117]	2020	General Purpose	SEA	RNN-LSTM	4	N/A	N/A
Huong et al. [116]	2021	IoT Networks	BoT-IoT	LocKedge	4	✗	✓
Li et al. [96]	2021	Industrial-CPSs	Gas Pipeline	CNN-GRU	[3, 5, 7]	✓	✗
Our IDS		Agri-IoT Networks	CSE-CIC-IDS2018	DNN	[5, 10, 15]	✓	✓
				CNN	[5, 10, 15]	✓	✓
				RNN	[5, 10, 15]	✓	✓
			InSDN	DNN	[5, 10, 15]	✓	✓
				CNN	[5, 10, 15]	✓	✓
				RNN	[5, 10, 15]	✓	✓
			MQTTset	DNN	[5, 10, 15]	✓	✓
				CNN	[5, 10, 15]	✓	✓
				RNN	[5, 10, 15]	✓	✓

Table 5.5: FELIDS and related works comparisons

datasets. In certain instances, FELIDS demonstrated performance that closely rivaled the centralized model, while in other scenarios, it surpassed the centralized model in terms of performance. This flexibility and adaptability make FELIDS a robust and reliable IDS for Agri-IoT networks.

By employing FL, FELIDS overcomes the limitations of centralized models, such as privacy worries, network latency, and the need for high-authorization third parties. Instead, FELIDS allows edge nodes to collaboratively share knowledge without threatening the data privacy.

Overall, FELIDS represents a step forward in the field of intrusion detection for Agri-IoT networks. Its ability to achieve high accuracy while maintaining data privacy makes it a promising solution for enhancing the cybersecurity of Agriculture 4.0. With the growing adoption of IoT devices in agriculture, FELIDS can play an important role in safeguarding sensitive data and protecting against cyber threats.

Chapter 6

Differentially Private and Decentralized FL-based IDS for IIoT

6.1 Introduction

IACS, or industrial automation and control systems, have historically been separate from digital networked environments in a variety of industries, including agriculture. But as IoT technology becomes more widely used, industrial system architectures are changing, resulting in more connected and intelligent industrial systems, or the Industrial Internet of Things (IIoT). Fig. 6.1 provides an illustration of a generic IIoT ecosystem, where every layer consists of various technologies serving different purposes, ranging from sensing and actuation to networking and complex computations. The IIoT industry is anticipated to surpass $100 billion by 2026[1]. While IIoT offers significant advantages, it also introduces security risks, making this sector more vulnerable to cyber attacks. Consequently, it has become a prime target for malicious activities.

The significant expansion of the IoT necessitates the implementation of effective security and privacy protocols to mitigate potential risks to the security and confidentiality of systems. Within the IoT framework, certain threats pose higher risks, and conventional security measures may prove insufficient [8]. The security of IoT stands as a critical bottleneck, limiting the successful deployment of IIoT. Consequently, without adequate security measures, the full realization of IIoT's envisioned potential remains doubtful. As a result, research aimed at improving IIoT security has significantly increased in recent years.

In the current literature, there has been a notable surge in interest regarding cybersecurity concerns pertaining to IIoT ecosystems, particularly focusing on FL-based IDS in recent times [55]. Nevertheless, many of the existing systems suffer from several drawbacks. For instance, they encounter challenges due to the scarcity of complex cyber attack patterns, since information owners are often reluctant to disclose sensitive information about their critical systems, making the task of constructing effective models extremely arduous. Additionally, the security of parameter exchange is frequently

[1] https://www.marketsandmarkets.com/Market-Reports/industrial-internet-of-things-market-129733727.html

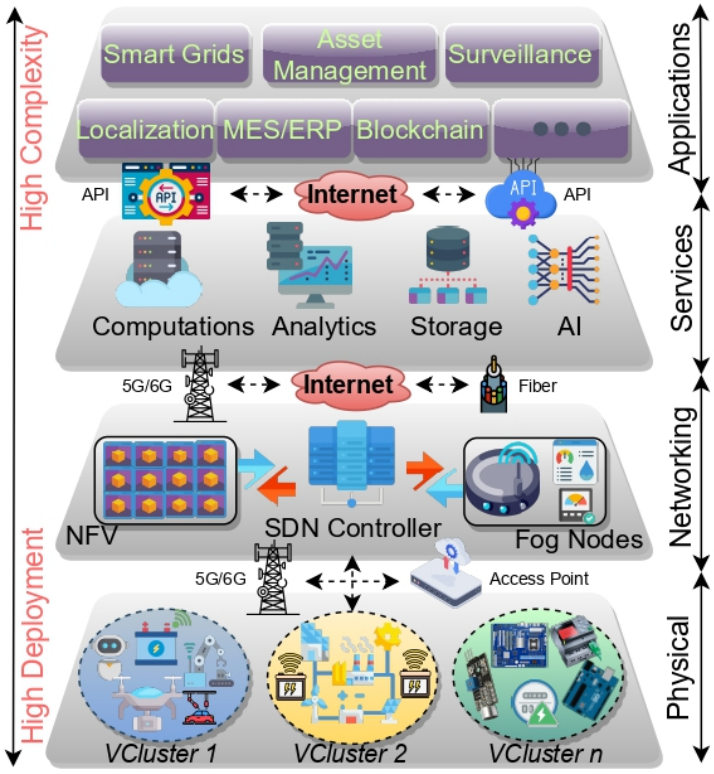


Figure 6.1: IIoT environment illustration

inadequate, leaving the systems vulnerable to both external and internal threats.

6.1.1 Contributions

Numerous shortcomings are evident in most existing systems within the FL-based cybersecurity systems. One major limitation is the scarcity of high-quality cyber attack instances, primarily because information owners are hesitant to disclose sensitive data concerning their critical systems, making it exceedingly challenging to construct effective models. Furthermore, the exchange of parameters in these systems often lacks the necessary level of security, rendering them susceptible to threats both from external sources and within the internal environment. Furthermore, systems with a centralized designs may expose the entire ecosystem to dangers.

In Figure 6.2, specifically in the high-level network design section. The proposed system aims to address the vulnerability of a SPOF present in conventional FL aggregation servers. This is accomplished by removing the necessity for a central organization to oversee the training sessions and allowing involved parties to train a model on their own. By eliminating this centralization, the system reduces the risk of compromise resulting from normal failures or malicious cyber-attacks. Additionally, FL has been discovered to be vulnerable to model extraction and model reverse attacks, wherein hackers can learn whether sensitive data is there by examining the gradients that are transferred [121]. To counter this threat, the proposed system suggests the utilization of quantum-based cryptography from outsiders, along with Differential Privacy (DP) mechanisms to

safeguard against any potential security incidents from insiders. Additionally, the system is built to guarantee low network usage and user-friendly resource utilization. In contrast, CL approaches may encounter performance and privacy concerns since they require the uploading of training data, leading to substantial data exchange before every centralized training session. The proposed system avoids this inefficiency and enhances privacy and performance in the learning process. In summary, our proposal aims to enhance security by eliminating a SPOF, protecting against various types of attacks, and ensuring efficient resource utilization and low network overhead compared to traditional CL approaches.

In this chapter, we demonstrate improvements in FL-based IDSs by addressing the previously mentioned issues, resulting in enhanced security, efficiency, and privacy preservation. Through a comprehensive performance analysis and a comparative study involving CL, FL, and state of the art approaches, we provide validation for our claims. Our work offers three primary contributions, as outlined below:

- Introduction of 2DF-IDS: We propose 2DF-IDS, a novel FL-based IDS designed to enhance security in both IoT and IIoT environments. This system ensures decentralized and secure operations, contributing to better protection against potential threats.
- Privacy Measures for Participating Clients: We place a strong emphasis on preserving privacy through 2 distinct approaches. Outbound privacy is upheld through the implementation of a quantum-resilient Ring-Learning With Errors (R-LWE). Additionally, for inside privacy, we incorporate DP.
- Decentralized Aggregation: To eliminate the SPOF vulnerability often encountered in conventional FL due to the central aggregation server, we adopt a fully decentralized aggregation. This step ensures robustness and improved security in the overall system.

By combining these contributions, our research establishes that FL-based IDSs can achieve higher levels of security, efficiency, and privacy protection in IoT and IIoT settings compared to existing methods. The detailed analysis presented in this chapter reinforces the significance of our proposed system and its potential impact on the field of cybersecurity.

6.2 2DF-IDS presentation

In this section, we present the 2DF-IDS system, which is designed to be secure, decentralized, and privacy-preserving for protecting IoT/IIoT networks. We define the threat model, outlining potential adversaries, and then describe the structure of the system.

6.2.1 System Overview

Figure 6.2 illustrates the proposed 2DF-IDS in high-level detail. The remainder of this section examines each part in detail.

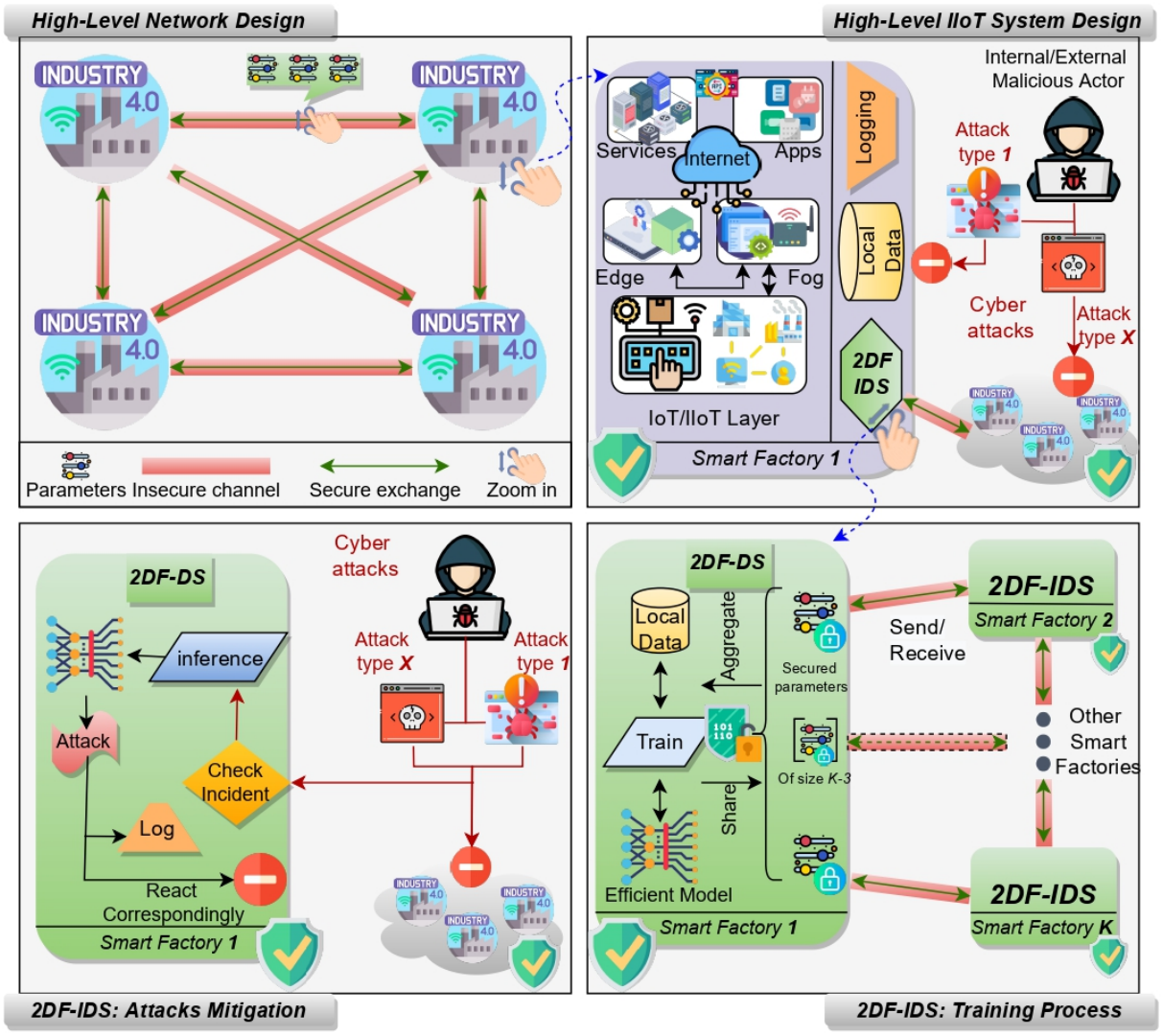


Figure 6.2: 2DF-IDS Illustration

Motivations

Currently, extensive research is being conducted on the advancement of IDSs using FL, as indicated by recent studies [66, 110]. This research is driven by the promise of achieving a certain level of privacy preservation in FL and efficient profiling of both benign and malicious patterns in IDS. The primary objectives encompass the assurance of the mentioned attributes. Additionally, our system aims to introduce supplementary security layers and efficiency-enhancing features to further enhance privacy, security, and portability. Our envisaged scheme is designed for deployment across various smart factories owned by different organizations. The goal is to collaboratively develop cybersecurity models without encountering privacy concerns. Similar to how data exchange between divisions of the same organization is anticipated to go well due to the mutual trust that has been built up. Therefore, even with a lower level of trust, incorporating cybersecurity knowledge from outside organizations under a common defense policy could potentially raise exposure and competition. Our proposition is uniquely tailored to address situations where collaborative trust-based becomes imperative within unpredictable environments. It's important to note that our system diverges from traditional FL, as it refrains from incorporating a centralized entity. FL has the potential to protect privacy while achieving accurate model training and efficient profiling of both benign and harmful behaviors. By utilizing these advantages, our suggested method aims to improve security, privacy, and dependability by adding extra protection layers and productivity-boosting features.

Objectives

The main objectives and contributions are as follows:

- *Enhanced Security and Privacy*: We aim to strengthen security by incorporating extra security layers to protect against both external threats and potential compromises from participating parties. The use of quantum-resilient R-LWE key exchange and differential privacy mechanisms adds robustness to data transmission and model aggregation, ensuring privacy preservation.
- *Efficiency and Reliability*: Our system is designed to enhance the efficiency of model training and aggregation, enabling accurate profiling of cyber threats. By eliminating the need for a centralized entity and adopting a P2P approach, we enhance the reliability and resilience of the system.
- *Collaborative Defense*: The proposed system is aimed to be deployed in various SF of different organizations. By collaboratively shaping defensive cybersecurity models, these organizations can jointly combat threats without conflicts of interest. Trust between different parts of the same organization allows data sharing without concerns, contributing to mutual trust training for unreliable environments.
- *Mutual Trust Training*: Our system fosters scenarios where sharing cybersecurity intelligence among organizations under the same defense policy is essential. By maintaining mutual trust training, we can address security challenges more effectively.

- *Decentralized Approach*: Unlike traditional FL, our system does not rely on a centralized entity for aggregating updates. Instead, it employs a peer-to-peer approach with multiple security layers, ensuring robustness against various security issues.

Threat Model

The "honest but curious" guiding principle guides the design of the 2DF-IDS system that we propose. This means that while participating clients honestly follow all system policies, they also have the option to look at their peers' parameters. We depict an opponent, indicated as *mathcalA*, as either a hostile force with malicious goals or an inquisitive system user who wants to learn more without causing harm. Even while it's still theoretically possible for an opponent to try to sabotage learning, the huge presence of many trustworthy entities significantly reduces the impact of the adversary's strategies. The system's resistance against possible threats from adversaries is strengthened by the collaborative framework. The ensuing threat model is considered for the proposed system:

- *SPOF Threat:* The server in conventional FL constitutes a solitary point of vulnerability. If this server is targeted by an adversary $\mathcal{A}$, it can disrupt the entire training, impacting the security of the system. Attacks on the aggregation server can significantly delay the implementation and deployment of the IDS model, providing more timing window for successful attacks.

- *Private Data Reconstruction:* Certain conditions in FL settings can threaten the privacy of training data, either unintentionally [122] or intentionally [121]. For instance, it has been demonstrated that the shared gradients can be used to reconstitute the private training data of specific users. Attackers can use model weight parameters w_k and biases b_k to reconstruct specific samples x_k, which compromises privacy.

- *Quantum-based Crypto-analysis Attacks:* Shor's idea demonstrates how public-key cryptography techniques now in use, which rely on challenging issues like prime number factorization and discrete logarithm, are susceptible to being broken by quantum computers [142]. This implies that future quantum-based attacks may be able to compromise the cryptosystems that are now utilized to protect the interchange of model parameters during FL training sessions.

Network and System Model

As shown in Fig. 6.2, the proposed scheme operates as follows:

- *Network model:* The network comprises of smart factories that are connected to one another via an insecure communication route. Securing this route is the main goal in order to safeguard transferred data.

- *System model:* Every factory is a combination of interoperating technologies. It provides a local private dataset, an event logging system, and a client instance of

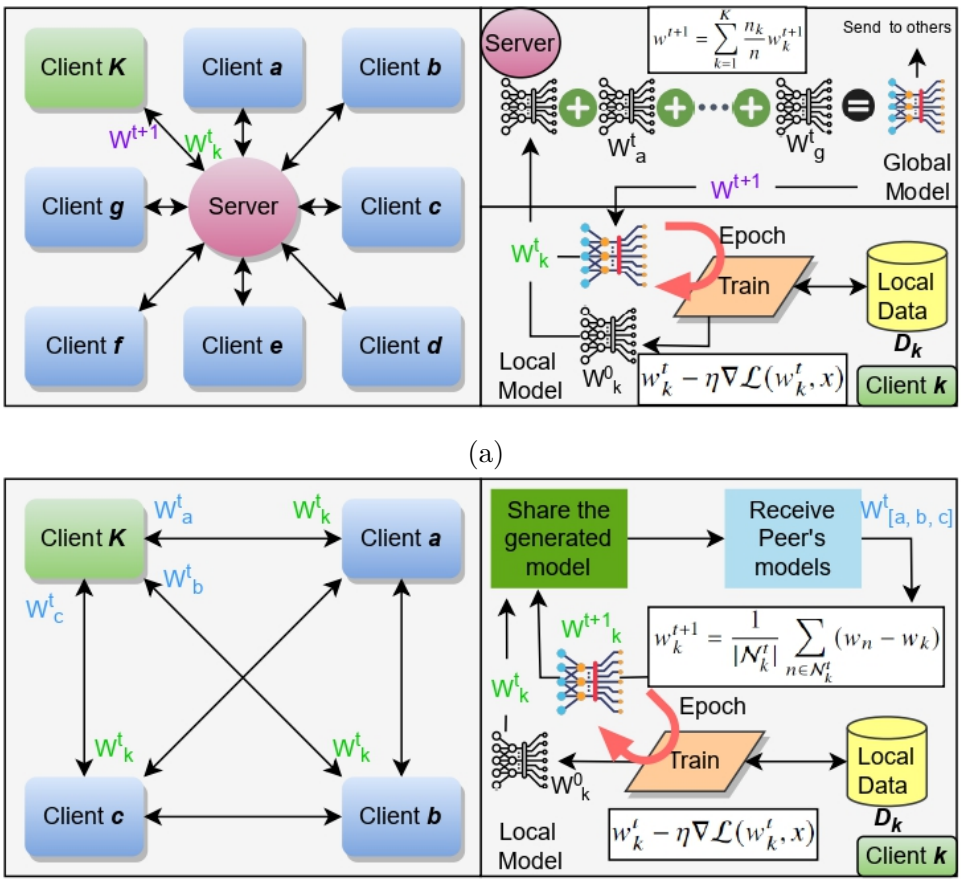
2DF-IDS. During training, clients exchange learned models based on their local data, participating in a cooperative and distributed learning. The communicated parameters need protection from both external threats (from the unsecured channel) and internal threats (within trusted zones). The collaboratively trained model effectively detects all attack types experienced by the group since all 2DF-IDS instances share collective client experiences.

Notation	Description
T	Global iterations
E	Clients local iterations
$\mathcal{P}_\mathcal{G}$	Connected graph
N_k^t	Neighborhood of the client k at t
D_k	Local Private Dataset for client k
w_k^t	Client k parameters at t
K	Total clients number
η	Learning rate
$\mathcal{L}(.)$	Loss function
$\mathfrak{g}(x_i)$	Gradient computed on x_i
C	Gradient norm bound
σ	DP Noise scale
R	Ring $(R := \mathbb{Z}[X]/(f(X)))$
R_q	Quotient Ring (R/qR)
χ	β-bounded Gaussian distribution over R_q
$a \leftarrow R_q$	Uniformly random sampling
s_k	Initial secret key for user k
P_k	Temporary public key for user k
$Sig(.)$	*Signal* Function
$Rec(.)$	*Reconciliation* function
LK_k	Initial shared session key
SK	*Ephemeral* key
$H(.)$	One-way Hash function
$KeyGen(.)$	Key Generation function
$Enc(.)$	Symmetric Encryption function
$Dec(.)$	Symmetric Decryption function
(ϵ, δ)	Privacy cost

Table 6.1: Notations used in the proposed scheme

6.2.2 2DF-IDS: Building Blocks

The following section describes the components used in our system. Tab. 6.1 shows the notations used.



(a)

(b)

Figure 6.3: Centralized vs. decentralized FL

Decentralization of FL

We adopt the algorithm of facilitates the exchange of information among the network participants from [143], ensuring continuous high connectivity. Each client can train a model using their personal data D_k. The exchanged params among connected clients are represented by W_k. Consider a graph $\mathcal{P}_\mathcal{G}$ representing P2P clients, defined as $\mathcal{P}_\mathcal{G} = (V, E)$, where V is a vector of K elements representing the clients, and E is the set of linked entities in the graph. An edge $(k, n) \in E$ indicates a connection between 2 clients (k and n). The set N_k^t, representing the neighborhood of client k at time t, encompasses all clients $1, ..., N$, with $N \leq K$, who are linked to client k and capable of information exchange with it. In *FedAvg* [54], each client computes the average gradient on its local data and updates its model weights using $w_k - \eta\nabla\mathcal{L}(w_k, x)$, where η is the learning rate, and $\mathcal{L}(.)$ is the loss function. In classic FL, every client updates its model weights using $w_k - \eta\nabla\mathcal{L}(w_k, x)$, then the server computes the new weight using $\sum_{k=1}^{K} \frac{n_k}{n} w^{t+1}k$. Returning to the network topology, we can consider the FL as a star network. In the event that client k is able to get all of the weights for the model from its connected neighbors in $N_k t$, it can update its model parameters by simply averaging the variations (Eq. 6.1 [144]). By averaging the resulting differences with its neighbors, each client in this procedure creates its own model. This enables effective cross-topology collaborative model update.

$$w_k = \frac{1}{|N_k^t|} \sum_{n \in N_k^t} (w_n - w_k) \tag{6.1}$$

There is at least one path connecting every pair of clients, hence it has been demonstrated that the aforementioned technique will eventually converge to the centroid of all client models [144]. Clients are able to train a model and come to an agreement without the help of the aggregation server as a result. In Fig. 6.3, we provide a detailed example that highlights the most important differences across aggregation server-based server-less FL. As shown, the primary distinction between the two methods is that the decentralized FL method's use of aggregation techniques is impacted by the absence of a central server managing the training process.

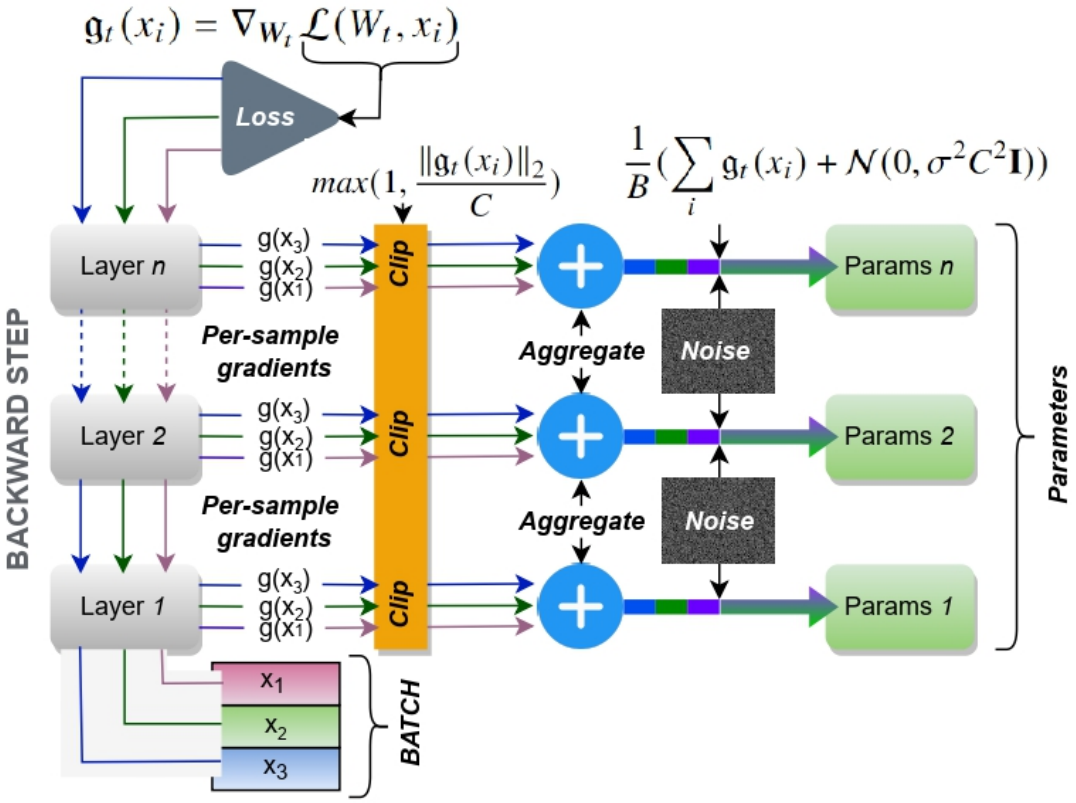


Figure 6.4: DP-SGD illustration

DP-based Privacy Preserving

It is essential to make sure that the models do not disclose any sensitive information about the trained data in a system where model parameters are exchanged amongst peers. A method for performing calculations on huge datasets while restricting the disclosure of sensitive information is the use of DP [145, 146]. To put it another way, a DP-satisfied scheme, quantified by (ϵ, δ) privacy settings, ensures that the inclusion or deletion of one statistic in the input will not materially affect the result or the statistics of the total dataset.

Formally, S is considered (ϵ, δ) private for datasets d and $d' \in \mathcal{D}$ (where $\mathcal{D}$ is the dataset space), which differ in at most one data sample. For each output subset $R \subseteq \mathcal{R}$, it is required $Pr(S(d) \in R) \leq e^{\epsilon} \cdot Pr(S(d') \in R) + \delta$, meaning that the probability of obtaining a certain output remains relatively consistent, even when a single data point is modified. This ensures strong privacy guarantees for the participants in the decentralized learning process.

Working only with the final parameters gleaned from the training process would be the initial naive approach to protecting private training data. However, adding an overly conservative amount of random data (or noise) to the parameters may severely diminish the effectiveness of the learned model [146]. A more advanced approach, as utilized by [147], is the implementation of Differentially Private Stochastic Gradient Descent (DP-SGD). DP-SGD is a privacy-preserving version of the SGD optimization algorithm that ensures differential privacy throughout the parameter updates. By adjusting the parameter gradients utilized during the model's weight updates rather than changing the data itself, this method preserves the confidentiality of the training dataset.

By doing so, the privacy of the training data is maintained while still allowing the model to learn meaningful and accurate representations from the data. This is accomplished via DP-SGD using:

$$\mathfrak{g}_t(x_i) = \nabla_{W_t} \mathcal{L}(W_t, x_i) \tag{6.2}$$

Then clip the gradient's l_2 norm:

$$max(1, \frac{\|\mathfrak{g}_t(x_i)\|_2}{C}) \tag{6.3}$$

Where C represents is the norm bound. The noise is added by:

$$\frac{1}{B}(\sum_i \mathfrak{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})) \tag{6.4}$$

The group size is denoted by B, and $\mathcal{N}$ is the Gaussian distribution. The noise scale is denoted by σ, which is used to introduce random noise, resulting in updated gradients $\mathfrak{g}t'$. The model parameters are then updated using $Wt - \eta\mathfrak{g}'_t$. Afterward, the overall privacy cost, represented by (ϵ, δ), is computed to quantify the level of differential privacy achieved by the DP-SGD optimization process. Fig. 6.4 shows the DP-SGD. The micro-batching technique employed in DP-SGD yields per-sample gradients, however, it requires a significant amount of time. *Vectorized computation* is an innovative technique introduced in [148], designed to address the speed issue in DP-SGD. By building a vectorized version of the per-sample gradient method and deriving it, this technique seeks to increase the effectiveness of DP-SGD. By utilizing vectorized operations, this approach enhances the computation speed, making it more efficient compared to the traditional per-sample gradient approach.

Secure Key Exchange Protocol

Even though DP preserves the confidentiality of training data at the client end, it might not offer extra defense across unprotected networks. To address this concern, we implement a secure key exchange protocol, enabling secure communication between peers over the transmission channels. Existing public key methods are based on hard problems like discrete logarithm problem. However, these methods are vulnerable to attacks from quantum computers, as demonstrated by [149]. Post-quantum cryptography, on the other hand, involves designing cryptographic systems that remain reliable even in the presence of adversaries with quantum computers. By adopting such post-quantum cryptographic approaches, the secure key exchange protocol ensures the resilience of the communication channels against potential quantum attacks, further enhancing the security of the 2DF-IDS system. Lattice-based cryptography has seen widespread application in practice, and one particular class of problems that has garnered significant interest in recent implementations is the LWE [150] problem and its more efficient variant, R-LWE [151], which is based on ring structures.

Let's consider the polynomial quotient ring R of degree n over the integers ($\mathbb{Z}$). It's defined as $R := \mathbb{Z}[X]/(f(X))$, where n is a power of 2, and $f(X)$ is an irreducible polynomial. Now, define R_q as $R/qR = \mathbb{Z}_q[X]/(f(X))$, where R_q represents the quotient ring with a prime integer modulus q. In this context, we have an *error distribution* denoted as χ, which characterizes short elements within the ring R. Subsequently, the *search problem* for Ring-LWE involves the retrieval of a uniformly random secret s through the acquisition of independent samples in the following manner:

$$(a_i, b_i = s \cdot a_i + e_i) \in R_q R_q \tag{6.5}$$

Where each $a_i \in R_q$, and $s_i \in R_q$ are randomly sampled uniformly and each $e_i \leftarrow \chi$, is drawn according to an error distribution χ. Furthermore, the *decision problem* involves the problem of distinguishing samples of the form $(a_i, b_i) = (s \cdot a_i + e_i) \in R_q R_q$. This decision problem is an important aspect of the Ring-LWE cryptographic scheme, as it establishes the security of the system by determining whether an adversary can differentiate between valid Ring-LWE samples and random samples from the ring R_q. Given the fact that is hard for quantum algorithms to approximate the Shortest Vector Problem (approx-SVP) in polynomial-time with the worst case on ideal lattices, Ring-LWE is known to be as hard as approx-SVP on any ideal lattice [151]. Ding et al.'s research [152] has exposed a vulnerability in the DXL-KE protocol, specifically its susceptibility to key reuse attacks. To bolster security, a set of more stringent measures has been underscored, including:

- Utilizing the Ring-LWE shared key as th input to the AES, which generates the final shared secret key. This approach ensures that the Ring-LWE shared key is utilized securely to derive the group key for communication.

- Changing both the secret/public settings for each training session. By updating these settings for each session, the system reduces the risk of potential attacks exploiting reused keys or configurations.

- Ensuring that the public and secret settings, as well as the cryptographic keys, are never reused for a new training session. This means that entirely new settings and keys should be generated for each subsequent training session, further enhancing the security of the communication.

With regard to selecting q, n, and the ring R, the final two conditions demand the usage of a randomization function. By adhering to these stricter requirements, the security of the DXL-KE protocol can be significantly improved, mitigating the risks associated with key reuse attacks and ensuring a more robust and secure communication process.

6.2.3 secure exchange, decentralized learning, and DP-enforced training

The system operates in two main phases: key exchange and collaborative training. In the first phase, the protocol from [153] is used, along with the AES cryptosystem as depicted in Algorithm 7. In the second phase, this ephemeral key will be used to protect the transmission pathway (Algorithm 8). Table 6.1 provides a list of notations used in the system.

Initialization

We look at initialization at two different levels: system-wide and clients.

- *Global system initialization:* The following public system parameters are required:
  - K: The number of clients.
  - w^0: The set of parameters used for initializing the model.
  - T: Global training epochs.
  - E: Local training epochs for each client.
  - η: The learning rate used during the training process.
  - n: An integer representing the power of 2 used as a dimension, defining the polynomial $f(x) = x^n + 1$.
  - R: The ring structure defined as $R := \mathbb{Z}[x]/(f(x))$.
  - q: A prime integer chosen to define the quotient ring $R_q = \mathbb{Z}q[x]/(f(x))$.
  - χ: The Discrete Gaussian distribution on Rq with a norm at most β.
  - a: A public sample uniformly randomly chosen from R_q.
- *Clients initialization:* During the initialization process for each client in the 2DF-IDS scheme, the following steps are performed:
  - Sets the initial parameter w_k^0 to the default parameter w^0.
  - Samples two random values s_k and e^0k from the Discrete Gaussian distribution χ, where sk represents the secret key. e^0k denotes the error distribution.

Algorithm 7: The protocol of training session key exchange for the group of clients

/* R: the used ring, χ: the discrete Gaussian distribution on R_q, and a: the uniformly random public sample from R_q. */
Input K, n, q, R, χ, a;
Output Symmetric group session key SK;
Require $n = x^2$, $f(x) = x^n + 1$, $a \leftarrow R_q$;
/* Clients initialization */
for $k = 0, .., K-1$ *in parallel* **do**
    Randomly sample. $s_k \leftarrow \chi$;
    Randomly sample. $e_k^0 \leftarrow \chi$;
    Compute. $P_k^0 = as_k + 2e_k^0$;
    Send P_k^0 to client $k+1$;
/* Group-SK negotiation */
for $k = 0, .., K-1$ **do**
    for $c = 1, .., K-2$ **do**
        Randomly sample. $e_k^c \leftarrow \chi$;
        Set. $z = (k+c)\ mod\ K$;
        Compute. $P_k^c = s_z \cdot P_k^{c-1} + 2e_k^c$;
        Send P_k^c to client $(z+1)\ mod\ K$;
/* Ephemeral key generation */
for $k = 0, .., K-1$ **do**
    Set. $nd = K-2$;
    Set. $np = (k == 0)?1 : k+1$;
    Randomly sample. $er_k \leftarrow \chi$;
    Compute. $LK_k = s_k \cdot P_{np}^{nd} + 2er_k$;
    if $k == 0$ **then**
        Compute. $\varsigma = Sig(LK_k)$;
        Compute. $TK_k = Rec(LK_k, \varsigma)$;
        Broadcast. ς ;
    else
        Compute. $TK_k = Rec(LK_k, \varsigma)$;
    /* AES ephemeral key */
    $SK = KeyGen(TK_k)$;

- Using the sampled values, k computes its public key P^0k as $P^0k = as_k + 2e_k^0$.
- Client k then sends its computed public key P^0k to the next client in the network.
- As a connected graph, the network is regarded, and denoted as $\mathcal{P}_\mathcal{G}$ consisting of P2P clients. Client positions in the graph can be managed to logically use it as a cyclic group.

Agreement on the Group-SK

This is where participants decide on the key to a joint training session.

- *Negotiating Initial group-SK:* In this stage, each client k goes through the process of negotiating the initial group secret key (Group-SK) with the other clients. The negotiation is carried out in two steps: 1. *Ring-LWE-based key exchange:* For every client k and a randomly chosen value c from the set $[1, \ldots, K-2]$, a value e_k^c sampled from χ. The client then computes $P_k^c = s_z \cdot P_k^{c-1} + 2e_k^c$, where $z = (k+c)$ modulo K. This computed value P_k^c is communicated to the next client in the network. Each client repeats this process by sampling another value e_k^c and computes a local key LK_k. 2. *Final Group-SK computation:* The first client in the network utilizes the *signal* ($Sig(.)$) and *Reconciliation* ($Rec(.)$) functions to generate TK_k (a temporary key) and broadcasts ς to the other clients. Other clients use $Rec(LK_k, \varsigma)$ to obtain the same temporary key. It has been demonstrated that with an "overwhelming probability," all clients will ultimately obtain the exact same value for TK_k with a $a\prod_{k=0}^{K-1} s_k + \psi$, provided that the condition $\frac{q}{4} - 2 \geq n^K \beta^{K+1}.4K \geq |\psi_0 - \psi_{k\in[1,..,K-1]}|$ is satisfied [153].
- *Ephemeral key generation:* After obtaining the key LK_k during the initial group-SK negotiation, all clients utilize this key to have an *ephemeral* key SK using AES. This process involves the use of four functions: 1. *One-way hash function H(.):* 2. *Key generation function KeyGen(.):* This function, defined as $KeyGen(x) = H(x||K)$, generates the ephemeral key SK by combining the value of x with the group secret key K using the one-way hash function. 3. *Symmetric encryption function Enc(.):* The symmetric encryption function *Enc(.)* takes plaintext data M and the generated ephemeral key KS to produce ciphertext C. 4. *Symmetric decryption function Dec(.):* The symmetric decryption function *Dec(.)* is used to decrypt ciphertext C using the ephemeral key KS, resulting in the original plaintext data M.

Model training

Once all clients $k \in [0, .., K-1]$ have established the group session key SK, they proceed to commence the training. This phase involves 2 main stages: local client training and decentralized aggregation, as outlined in Algorithm 8.

- *Privacy-enhanced learning:* clients starts training on D_k. Every k then performs training on mini-batches of data D_k^i. During each iteration, the client computes

Algorithm 8: 2DF-IDS algorithm

Input SK, $\mathcal{P}_{\mathcal{G}}$, $\{N_k^t\}^{k \in K}$, K, w^0, T, E, $\{D_k\}^{k \in K}$, η , C, σ;
Output Trained model w^T, and overall privacy cost (ϵ, δ) ;
Require Prior execution of Alg. 7 ;
Set. $w_k^0 = w^0$ for $k \in [0, .., K]$;
/* Local Training */
for $t = 0, .., T$ **do**
  for $k = 0, .., K-1$ *in parallel* **do**
    for $i = 1, .., E$ **do**
      Sample. $D_k^i \leftarrow D_k$, with Prob. $\frac{D_k^i}{D_k}$;
      Set. $MB = size(D_k^i)$;
      for *each* $d_k \in D_k^i$ **do**
        Compute. $\mathfrak{g}_k^t(d_k) = \nabla_{w_k^t} \mathcal{L}(w_k^t, d_k)$;
        Clip. $\mathfrak{g}_k^t(d_k) = \frac{\mathfrak{g}_k^t(d_k)}{max(1, \frac{\|\mathfrak{g}_t(d_k)\|_2}{C})}$;
      Compute. $\mathfrak{g}_k^t = \frac{1}{MB}(\sum_{d_k \in D_k^i} \mathfrak{g}_k^t(d_k) + \mathcal{N}(0, \sigma^2 C^2))$;
      Compute. $w_k^{t+1} = w_k^t - \eta \cdot \mathfrak{g}_k^t$;
      Compute. $cw_k^{t+1} = Enc(w_k^{t+1}, SK)$;
      Send. cw_k^{t+1} to all neighbors $n \in N_k^t$;
    /* Global Training */
    while *No consensus* **do**
      for $k = 0, .., K-1$ **do**
        for *each* $n \in N_k^t$ **do**
          Compute. $NW_k^t[n] = Dec(cw_n^{t+1}, SK)$;
        Compute. $w_k^{t+1} = \frac{1}{|N_k^t|} \sum_{n \in N_k^t} (NW_k^t[n] - w_k^{t+1})$;

the gradient $\mathfrak{g}k^t(dk)$ for each data sample d_k using $\nabla_{w_k^t}\mathcal{L}(w_k^t, d_k)$, and subsequently applies $l2$ norm clipping [147]. The client calculates the new local model parameters wk^{t+1} after completing the local epochs and applying Gaussian noise $\mathcal{N}(0, \sigma^2 C^2))$ to the gradients. After completing the local epochs, the client introduces Gaussian noise $\mathcal{N}(0, \sigma^2 C^2))$ to the gradients, where σ denotes the noise scale, and computes the new local model parameters wk^{t+1}.

- *Decentralized aggregation:* Upon receiving all ciphered new models from its neighbors, client k decrypts each one of them using $Dec(cw_n^{t+1}, SK)$, then saves in $NW_k^t[n]$. Subsequently, client k executes the consensus protocol to aggregate all the weights, computing $\frac{1}{|N_k^t|}\sum_{n\in N_k^t}(NW_k^t[n] - w_k^{t+1})$ to get w_k^{t+1}. Prior to starting the subsequent local update cycle, the consensus procedure makes sure that the final step is repeated until all clients have obtained identical model parameters [144].

6.3 Performance evaluation

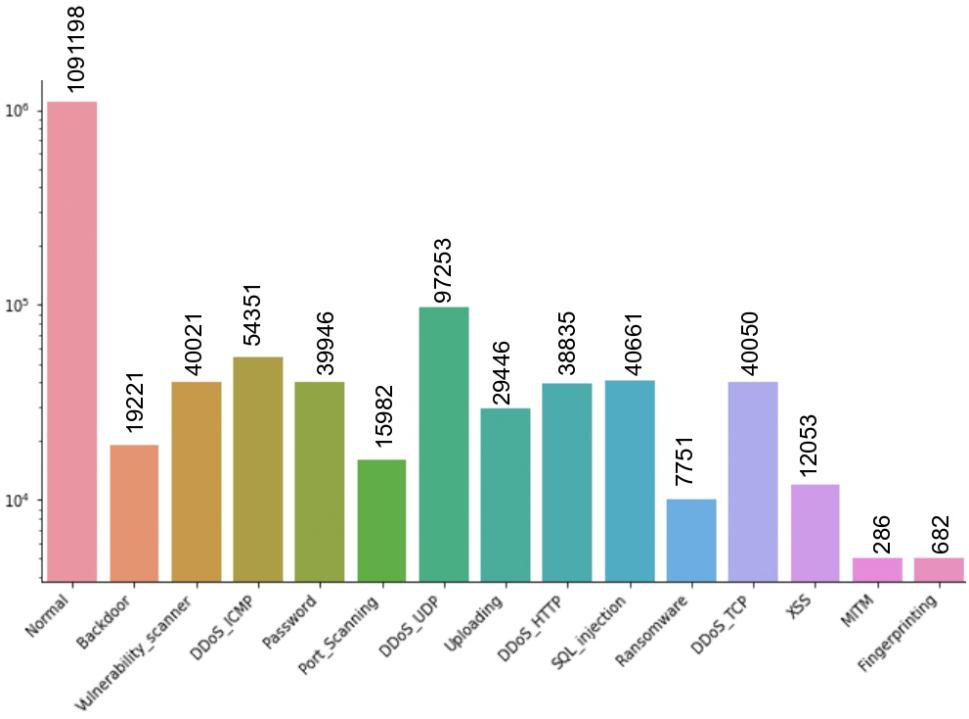
Within the present section, we first present of the employed dataset, and then evaluate our proposed 2DF-IDS scheme implementation. Furthermore, we demonstrate the efficiency of our proposed scheme by benchmarking it against various recent studies.

6.3.1 Dataset description

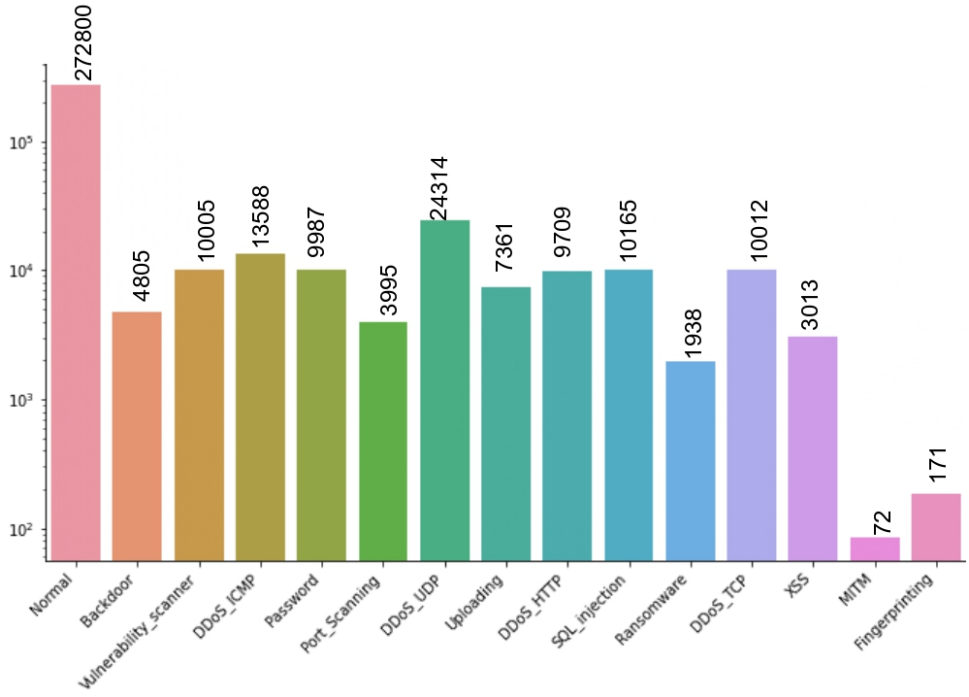
The dataset used in this study is the Edge-IIoTset dataset [65], which is specifically designed for cybersecurity purposes in IoT/IIoT-based applications.

The Edge-IIoTset dataset provide a comprehensive representation of various IoT/IIoT cyber attacks flows. These attacks are classified into the following classes:

- *DoS/DDoS attacks:* There are approximately 288,112 samples in this class, accounting for 15.09% of the total samples.
- *Information gathering attacks:* This category encompasses network port scanning, Operating Systems fingerprinting, and vulnerability scanning attacks. There are around 70,856 samples in this class, representing 3.71% of the total samples.
- *MITM attacks:* This category includes ARP and DNS Spoofing attacks. There are approximately 358 samples in this class, accounting for 0.02% of the total samples.
- *Injection attacks:* This category comprises SQL injection, XSS, and uploading attacks. There are around 102,699 samples in this class, representing 5.38% of the total samples.
- *Malware attacks:* This category includes ransomware, backdoor, and password cracking attacks. There are approximately 83,648 samples in this class, accounting for 4.38% of the total samples.



(a) Train set

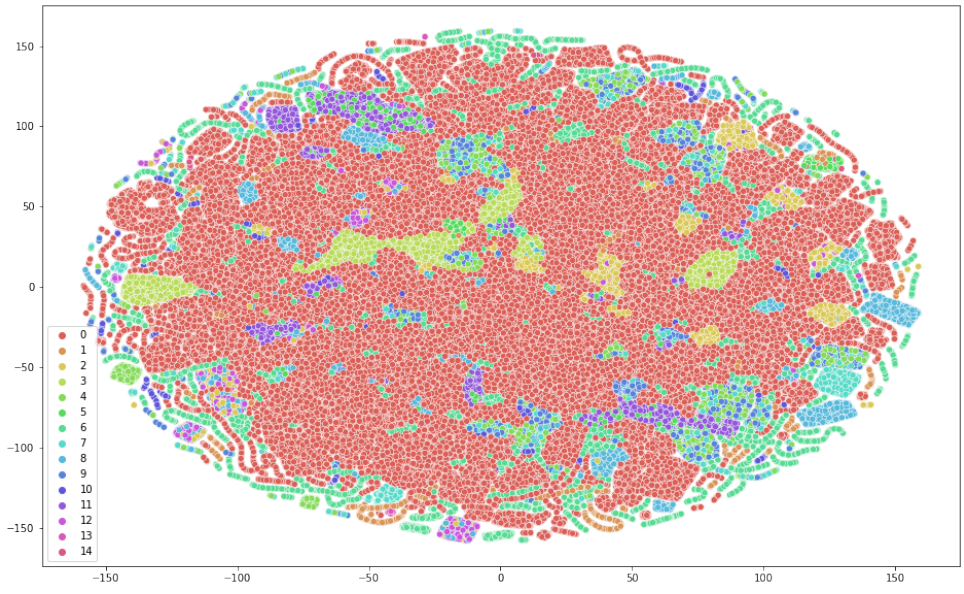


(b) Test set

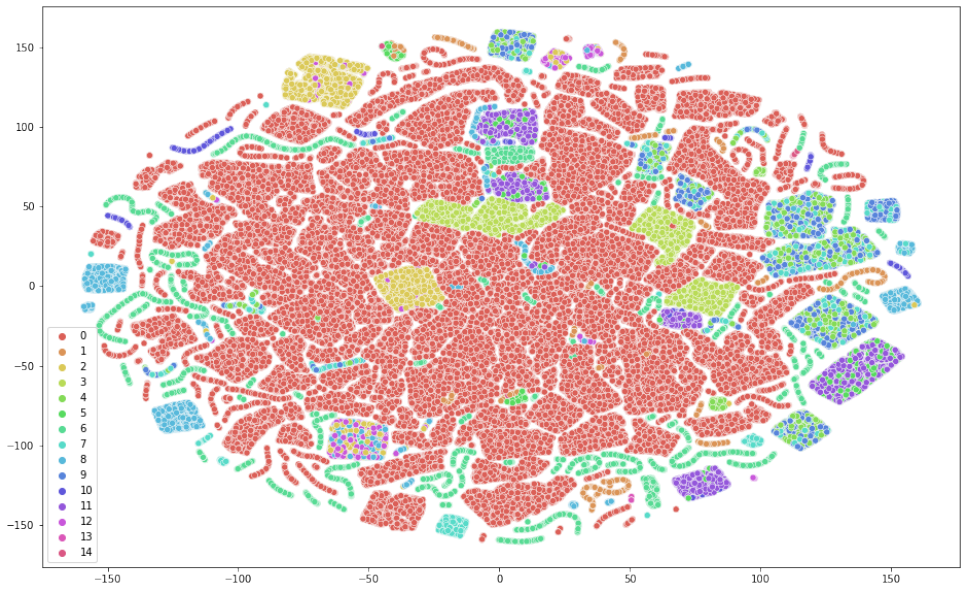
Figure 6.5: Classes distribution visualization

Having such a diverse set of attack types makes the Edge-IIoTset dataset highly suitable for evaluating and testing the performance of IDS and cybersecurity algorithms in IoT/IIoT environments.

The statistics of records are illustrated in Fig. 6.5. Additionally, Fig. 6.6 presents



(a) Training



(b) Test

Figure 6.6: t-SNE

the t-Distributed Stochastic Neighbor Embedding (t-SNE) for the training and test sets, providing a visualization of the data distribution. Where each class is represented as follows: Normal (0), Backdoor (1), Vulnerability Scanner (2), DDoS ICMP (3), Password (4), Port Scanning (5), DDoS UDP (6), Uploading (7), DDoS HTTP (8), SQL Injection (9), Ransomware (10), DDoS TCP (11), XSS (12), MITM (13), and Fingerprinting (14)

6.3.2 Experimental settings

The performance evaluation involved conducting 2 distinct groups of experiments on the G-Colab platform. Table 6.2 presents the varied settings utilized in the experiments.

Subject	Parameters	Values
Classifier	Hidden nodes	215
	Hidden layers	3
	Nodes per layer	128, 64, 32
	Learning rate	[0.1, 0.01, 0.0001]
	Regularization	L_2
	Loss function	*CrossEntropyLoss*
	Activation function	*ReLu*
	Batch size	1000
	Classification function	*SoftMax*
FL	Clients Sets	[20, 40, 80]
	Data Distribution	Non-IID
	Local epochs	1
	Global epochs	30
	Batch size	100
DP	ϵ	[0.2, 0.5, 1.0]
	δ	$2.e^{-7}$
	C	1.2

Table 6.2: Settings used in the experiments

The learning rate (η) was experimented with different values, namely 0.1, 0.001, and 0.0001. Fig. 6.7 illustrates the accuracy obtained for each learning rate. Among the different choices, the value of 0.01 demonstrated the highest accuracy. Consequently, it was adopted.

The experiments sets

The experiments conducted can be summarized as follows:

- *First set:* In this set of experiments, the 2DF-IDS performance is evaluated in comparison to two other approaches: the centralized data training version and classic FL version. DP was not used in these experiments to allow a direct comparison between the decentralized, centralized, and FL approaches. Experiments, aside from the centralized learning, was repeated using three different sets of clients: 20, 40, and 80 clients. Non-IID was used since it is more realistic for real-life settings.
- *Second set:* Similar to the first set, these experiments were repeated, but this time, DP was introduced with three different fixed ϵ values: 0.2, 0.5, and 1.0.

6.3.3 Results

Models training performances

In Fig. 6.8 the learning effectiveness of the initial set of experiments is depicted. Every learning epoch shows an increase in the validation accuracy of the models across all

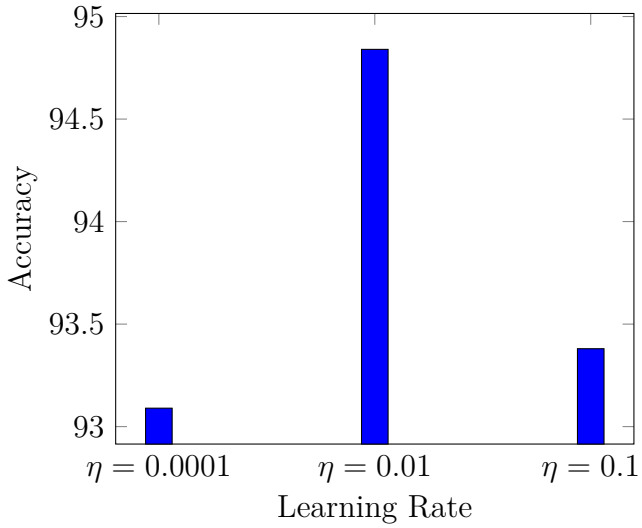


Figure 6.7: η values and related accuracy

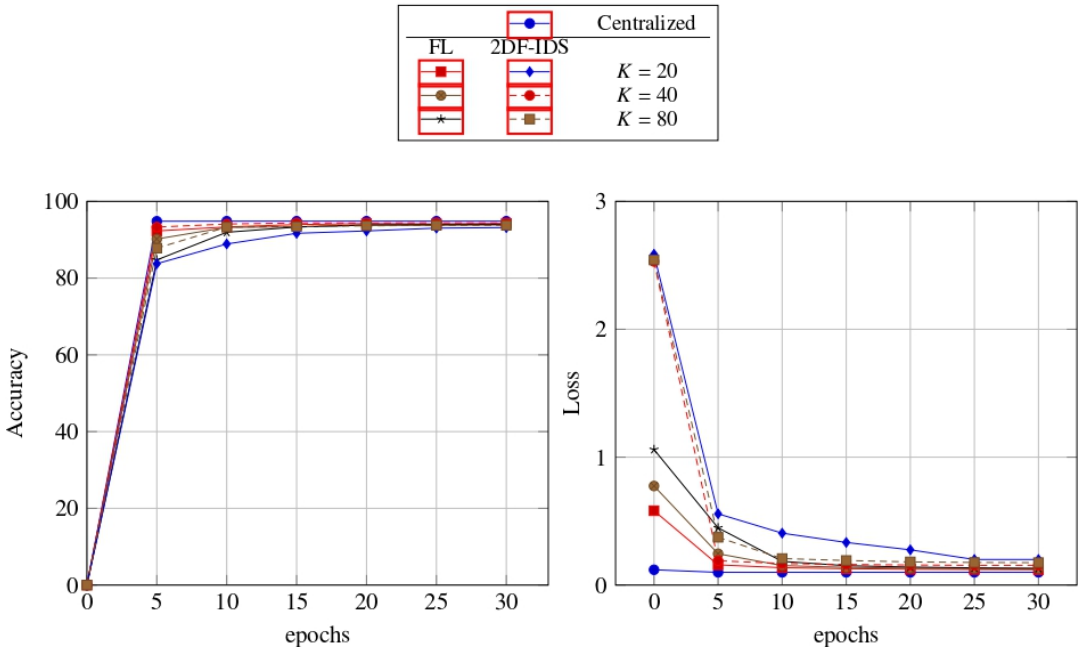


Figure 6.8: Accuracy: DP-disabled

Model	K	DP-settings			
		No-DP	$\epsilon = 1.0$	$\epsilon = 0.5$	$\epsilon = 0.2$
Centralized	-	94.84%	94.29%	94.22%	94.05%
FL	20	94.27%	90.90%	90.99%	88.87%
	40	93.91%	86.79%	87.93%	83.33%
	80	93.96%	80.94%	80.80%	79.89%
2DF-IDS	20	93.17%	91.95%	91.69%	84.64%
	40	94.37%	90.82%	90.40%	78.00%
	80	93.78%	79.13%	77.95%	77.95%

Table 6.3: Accuracy after 30 epochs

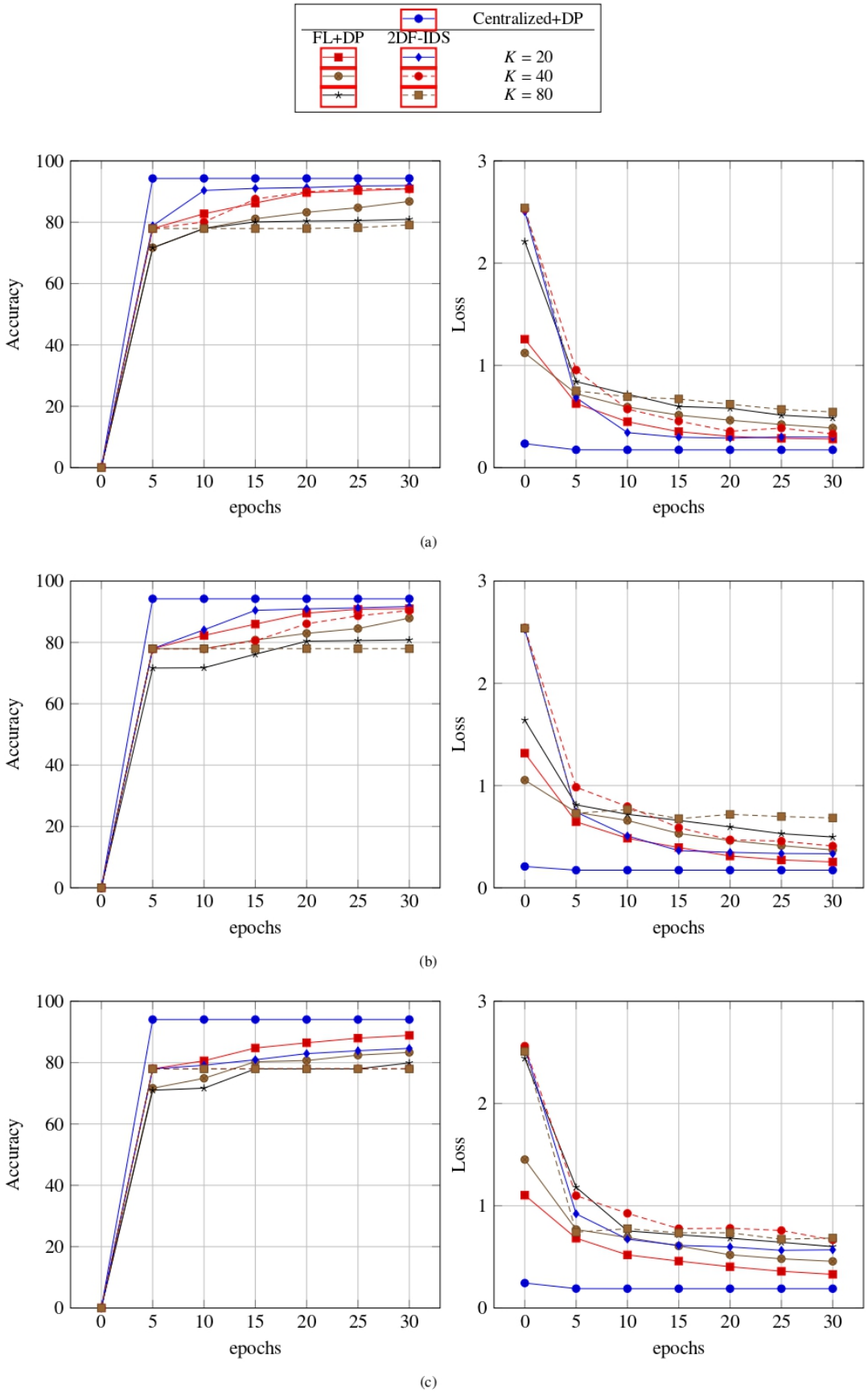


Figure 6.9: Accuracy: (a) $\epsilon = 1.0$, (b) $\epsilon = 0.5$, and (c) $\epsilon = 0.2$

situations. This finding highlights the mutual benefit that all client sets get from their peers' expertise, ultimately strengthening the shared model. In comparison to the first set of tests, there is a relative decline in performance for all training procedures when DP is included (Fig. 6.9). This result is anticipated given the added noise. However, Tab. 6.3 reports that in some specific situations, the 2DF-IDS model performs better than the FL model. The FL model's validation accuracy after 30 training epochs, for example, FL scored 86.79% whereas the 2DF-IDS model's reached 90.82% for *epsilon* = 1.0 and $K = 40$. The need of accurately identifying the Benign class within an IDS must be emphasized because false positive rates might cause unneeded confusion and undesired warnings. We consistently achieve a True Positive Rate (TPR) of 100% and a False Positive Rate (FPR) of 0% for the Benign class across all experiments for the FL and 2DF-IDS models. Tab. 6.4 presents the performance of different approaches in various settings, in terms of precision, recall, and F1 score (Macro and Weighted Avg.). The results show that the 2DF-IDS outperforms FL in different experimental settings. For example, when $\epsilon = 0.5$, the Macro F1 score for the FL model was 32%, while the 2DF-IDS reached 39%. An overview of the precision, recall, and F1 score of several techniques in various contexts is given in Table 6.4. In many experimental conditions, the results demonstrate that the 2DF-IDS is better. For instance, when $\epsilon = 0.5$ 2DF-IDS scored 39%, versus 32% for the FL model's Macro F1.

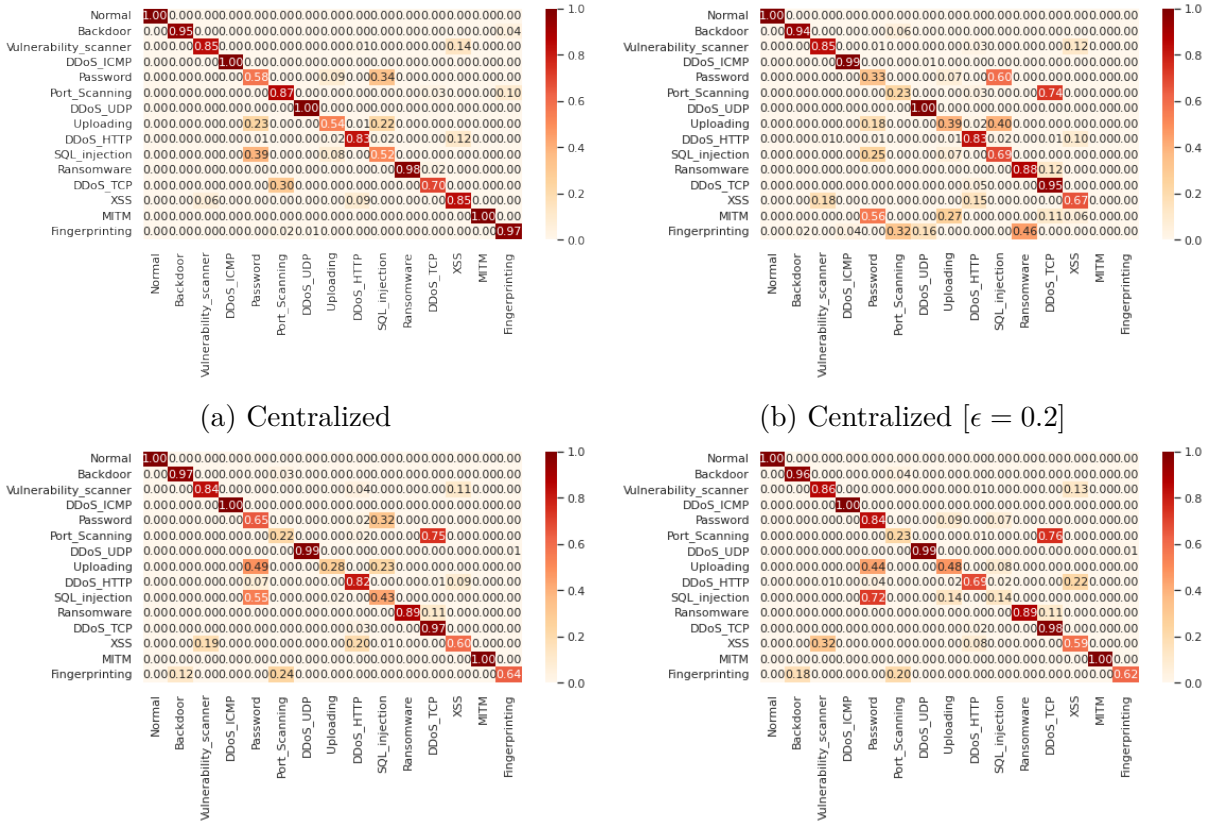
Model	DP	Macro-Avg			Weighted-Avg		
		Pr.	Re.	F1.	Pr.	Re.	F1.
Centralized	No-DP	79%	84%	79%	96%	95%	95%
	$\epsilon = 1.0$	83%	76%	76%	95%	94%	94%
	$\epsilon = 0.5$	82%	76%	77%	95%	94%	94%
	$\epsilon = 0.2$	68%	65%	65%	94%	94%	94%
FL	No-DP	79%	74%	73%	95%	94%	94%
	$\epsilon = 1.0$	32%	35%	31%	85%	87%	85%
	$\epsilon = 0.5$	38%	38%	32%	86%	88%	85%
	$\epsilon = 0.2$	19%	27%	21%	79%	83%	80%
2DF-IDS	No-DP	82%	77%	77%	95%	94%	94%
	$\epsilon = 1.0$	41%	48%	43%	88%	91%	89%
	$\epsilon = 0.5$	39%	47%	39%	88%	90%	88%
	$\epsilon = 0.2$	10%	13%	12%	75%	78%	76%

Precision (Pr.); Recall (Re.); F1-score (F1.)

Table 6.4: Precision, recall, and F1 scores

Per-class performances

We looked at how well each of the three methods performed per class. Centralized model without DP (Fig. 6.10a), versus introducing severe privacy budget ($\epsilon = 0.2$) (in Fig. 6.10b). Additionally, we compare FL (Fig. 6.10c) to 2DF-IDS (Fig. 6.10d). We can see from the first comparison that the added noise has had an impact on the centralized model's performance in each class. Furthermore, we see that, compared to the prior comparison, FL and 2DF-IDS perform similarly to each other when subjected to the same experimental settings regarding client sets. However, significant differences still exist in the detection of certain attacks. However, as shown in Table 6.4 when noise is added



(a) Centralized

(b) Centralized [$\epsilon = 0.2$]

(c) FL [$K = 80$]

(d) 2DF [$K = 80$]

Figure 6.10: Confusion matrices

to assure robust privacy preservation, 2DF-IDS performs better than FL, another angle of comparison can be found in Tab. 6.5. Which evaluate per-class performance under $K = 40$ settings. The rigorous privacy budget has the biggest effect on the performance of all models. However, the majority classes are less affected by this restriction than are the minority classes, and this can be seen throughout the benign class.

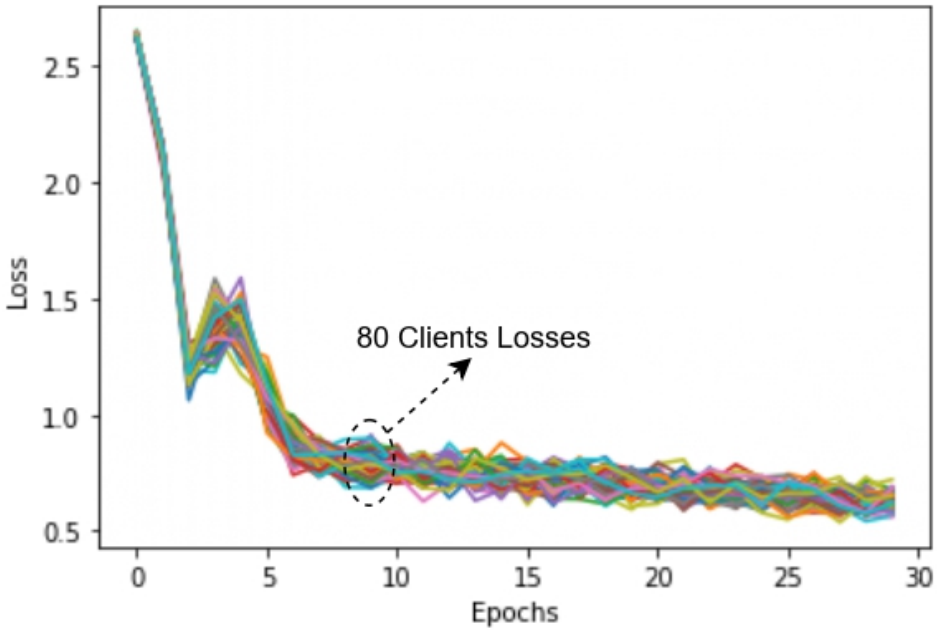
2DF-IDS vs. FL

The earlier findings show that, for all methodologies, there is an obvious trade-off between effectiveness, private budgetary constraints, and the total number of clients engaged with the training process. Particularly, performance tends to suffer whenever additional customers take part and tougher DP requirements are put in place. Furthermore, we have demonstrated in several cases the benefits of 2DF-IDS over FL. The individual customer loss throughout a training is another intriguing comparison point, as depicted in Fig. 6.11. The most important remarque here is that, in FL, all clients start out with similar loss values but eventually start to diverge as training goes on. Contrarily, in 2DF-IDS, each client starts training with a separate default model and set of local data, resulting in a range of losses for each client. The models of all clients, however, become considerably closer with each epoch as decentralized aggregation proceeds, resulting in uniform losses at the end. As a result of effectively collecting a robust representations of all customers' data without sacrificing security or privacy, the outcome of the 2DF-IDS model seems

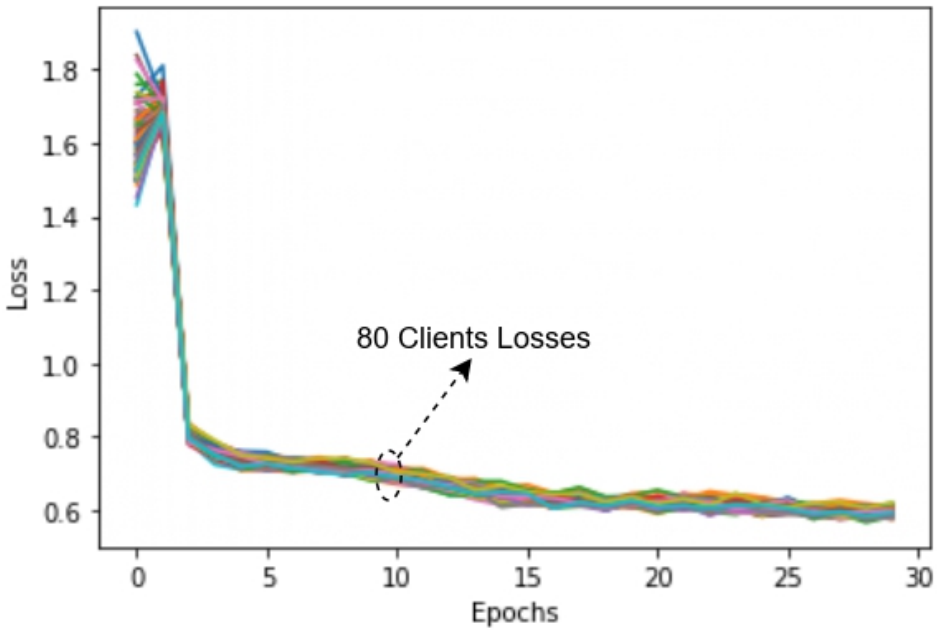
Class	DP settings	Precision			Recall			F1-score		
		CL	FL	2DF-IDS	CL	FL	2DF-IDS	CL	FL	2DF-IDS
N	No-DP	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	$\epsilon = 1.0$	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	$\epsilon = 0.5$	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	$\epsilon = 0.2$	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
B	No-DP	100 %	100 %	100 %	95 %	95 %	95 %	98 %	97 %	97 %
	$\epsilon = 1.0$	99 %	00 %	71 %	97 %	00 %	97 %	98 %	00 %	82 %
	$\epsilon = 0.5$	99 %	100 %	61 %	96 %	0.02 %	100 %	98 %	0.03 %	76 %
	$\epsilon = 0.2$	100 %	00 %	00 %	94 %	00 %	00 %	97 %	00 %	00 %
VS	No-DP	97 %	95 %	95 %	85 %	84 %	84 %	91 %	89 %	89 %
	$\epsilon = 1.0$	95 %	30 %	52 %	84 %	43 %	100 %	89 %	35 %	68 %
	$\epsilon = 0.5$	95 %	43 %	62 %	84 %	99 %	99 %	89 %	60 %	76 %
	$\epsilon = 0.2$	93 %	20 %	00 %	85 %	100 %	00 %	89 %	33 %	00 %
P	No-DP	50 %	40 %	43 %	58 %	64 %	84 %	53 %	49 %	57 %
	$\epsilon = 1.0$	43 %	00 %	22 %	77 %	00 %	0.09 %	55 %	00 %	0.13 %
	$\epsilon = 0.5$	45 %	37 %	00 %	49 %	76 %	00 %	47 %	50 %	00 %
	$\epsilon = 0.2$	45 %	00 %	00 %	33 %	00 %	00 %	38 %	00 %	00 %
DI	No-DP	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	$\epsilon = 1.0$	100 %	98 %	97 %	100 %	55 %	94 %	100 %	70 %	95 %
	$\epsilon = 0.5$	100 %	91 %	98 %	100 %	51 %	92 %	100 %	66 %	95 %
	$\epsilon = 0.2$	99 %	00 %	0.01 %	99 %	00 %	0.01 %	99 %	00 %	0.01 %
PS	No-DP	54 %	77 %	76 %	87 %	23 %	23 %	66 %	35 %	35 %
	$\epsilon = 1.0$	82 %	00 %	00 %	23 %	00 %	00 %	35 %	00 %	00 %
	$\epsilon = 0.5$	77 %	00 %	00 %	22 %	00 %	00 %	35 %	00 %	00 %
	$\epsilon = 0.2$	73 %	00 %	00 %	23 %	00 %	00 %	35 %	00 %	00 %
DU	No-DP	100 %	100 %	100 %	100 %	99 %	100 %	100 %	99 %	100 %
	$\epsilon = 1.0$	100 %	79 %	94 %	100 %	99 %	99 %	100 %	88 %	97 %
	$\epsilon = 0.5$	99 %	78 %	92 %	100 %	99 %	99 %	100 %	87 %	96 %
	$\epsilon = 0.2$	99 %	63 %	56 %	100 %	100 %	100 %	100 %	77 %	72 %
U	No-DP	68 %	83 %	62 %	54 %	30 %	44 %	60 %	44 %	52 %
	$\epsilon = 1.0$	65 %	00 %	00 %	41 %	00 %	00 %	50 %	00 %	00 %
	$\epsilon = 0.5$	62 %	00 %	00 %	42 %	00 %	00 %	50 %	00 %	00 %
	$\epsilon = 0.2$	66 %	00 %	00 %	00 %	39 %	00 %	00 %	49 %	00 %
DH	No-DP	95 %	72 %	92 %	83 %	91 %	82 %	89 %	80 %	87 %
	$\epsilon = 1.0$	89 %	92 %	70 %	83 %	43 %	43 %	86 %	58 %	53 %
	$\epsilon = 0.5$	88 %	80 %	78 %	83 %	43 %	15 %	85 %	56 %	25 %
	$\epsilon = 0.2$	85 %	43 %	00 %	83 %	31 %	00 %	84 %	36 %	00 %
SI	No-DP	51 %	46 %	56 %	52 %	44 %	19 %	51 %	45 %	29 %
	$\epsilon = 1.0$	50 %	24 %	37 %	27 %	90 %	73 %	35 %	38 %	49 %
	$\epsilon = 0.5$	45 %	00 %	36 %	53 %	00 %	98 %	49 %	00 %	53 %
	$\epsilon = 0.2$	43 %	00 %	00 %	69 %	00 %	00 %	53 %	00 %	00 %
R	No-DP	100 %	100 %	100 %	98 %	89 %	90 %	99 %	94 %	95 %
	$\epsilon = 1.0$	100 %	00 %	00 %	88 %	00 %	00 %	93 %	00 %	00 %
	$\epsilon = 0.5$	100 %	00 %	00 %	88 %	00 %	00 %	93 %	00 %	00 %
	$\epsilon = 0.2$	96 %	00 %	00 %	88 %	00 %	00 %	91 %	00 %	00 %
DT	No-DP	98 %	75 %	75 %	70 %	98 %	98 %	82 %	85 %	85 %
	$\epsilon = 1.0$	75 %	62 %	68 %	96 %	93 %	100 %	84 %	74 %	81 %
	$\epsilon = 0.5$	75 %	45 %	56 %	96 %	99 %	96 %	84 %	61 %	71 %
	$\epsilon = 0.2$	74 %	57 %	00 %	95 %	77 %	00 %	84 %	66 %	00 %
X	No-DP	50 %	65 %	47 %	85 %	21 %	78 %	63 %	32 %	59 %
	$\epsilon = 1.0$	48 %	00 %	00 %	78 %	00 %	00 %	59 %	00 %	00 %
	$\epsilon = 0.5$	48 %	00 %	00 %	77 %	00 %	00 %	59 %	00 %	00 %
	$\epsilon = 0.2$	48 %	00 %	00 %	67 %	00 %	00 %	56 %	00 %	00 %
M	No-DP	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
	$\epsilon = 1.0$	100 %	00 %	00 %	100 %	00 %	00 %	100 %	00 %	00 %
	$\epsilon = 0.5$	100 %	00 %	00 %	100 %	00 %	00 %	100 %	00 %	00 %
	$\epsilon = 0.2$	00 %	00 %	00 %	00 %	00 %	00 %	00 %	00 %	00 %
F	No-DP	21 %	27 %	88 %	97 %	67 %	60 %	35 %	39 %	71 %
	$\epsilon = 1.0$	100 %	00 %	00 %	43 %	00 %	00 %	60 %	00 %	00 %
	$\epsilon = 0.5$	93 %	00 %	00 %	54 %	00 %	00 %	68 %	00 %	00 %
	$\epsilon = 0.2$	00 %	00 %	00 %	00 %	00 %	00 %	00 %	00 %	00 %

Centralized Learning (CL); Normal (N); Backdoor (B); Vulnerability Scanning (VS); Password (P); DDOS ICMP (DI); Port Scanning (PS); DDoS UDP (DU); Uploading (U); DDoS HTTP (DH); SQL Injection (SI); Ransomware (R); DDoS TCP (DT); XSS (X); MiTM (M); Fingerprinting (F)

Table 6.5: Performance Per-class



(a) FL



(b) 2DF-IDS

Figure 6.11: Training loss [$K = 80$, $\epsilon = 0.2$]

to be more effective. In conclusion, our research shows that 2DF-IDS outperforms FL in a variety of experimental conditions, striking a good balance among model efficacy and privacy preservation. Because 2DF-IDS is decentralized, users can contribute data without disclosing private information, fostering an interdependent and privacy-preserving learning environment.

System	Dataset	Classifier	FL Settings			Privacy Settings		C.S.	
			K	D.D.	D.	DP	ϵ	E.C.	P.Q.
[97]	Collected	RNN-GRU	[5, 9, 15]	Non-IID	✗	✗	✗	✗	✗
[115]	MIMIC	MLP	[2, 4, 8, 16, 32, 64]	N/A	✗	✗	✗	✗	✗
[118]	MNIST, CIFAR10	CNN	[10, 20,..,50]	Non-IID	✗	✗	✗	✗	✗
[99]	MNIST	CNN	[20, 30,..,100]	Non-IID	✗	✓	[0.5, 2]	✓	✓
[119]	Collected	SGD	4	Non-IID	✓	✗	✗	✓	✗
[120]	TON_IoT	GRU	N/A	Non-IID	✗	✗	✗	✗	✗
[121]	Drebin, Genome, Contagio	CNN	[5, 6,..,15]	Non-IID	✗	✗	✗	✗	✗
[96]	GasPipeline	CNN-GRU	[3, 5, 7]	IID	✗	✗	✗	✓	✗
[122]	Modbus network	GRU	N/A	Non-IID	✗	✗	✗	✓	✗
Ours	Edge-IIoTset	DNN	[20, 40, 80]	Non-IID	✓	✓	[0.2, 0.5, 1.0]	✓	✓

Data Distribution (D.D.); Communication Security (C.S.); Encrypted Communications (E.C.); Post-Quantum (P.Q); Decentralized (D.)

Table 6.6: 2DF-IDS and recent works comparisons

2DF-IDS vs. related works

In Table 6.6, we present a comparative analysis between our proposed 2DF-IDS and other approaches [95, 116, 115, 97, 118, 94, 112]. Noteworthy distinctions and advantages of our system are as follows:

- Classifier Simplicity: Our approach utilizes a lightweight and efficient DNN model (as presented in Table 6.2). This choice ensures the efficient communication of model parameters, resulting in cost-effective performance. Conversely, some other studies employed more complex classifiers such as CNNs [115, 97, 118, 94], leading to increased model sizes.

- Training Params: In our experiments, we evaluate the system's performance with three distinct sets of clients (20, 40, and 80), providing a comprehensive analysis under varying scenarios. Conversely, some other studies used a single set of clients with a smaller number (e.g., 4 clients in [116]), while others worked with small-sized client sets (only 3, 5, and 7 in [94]) or larger but still relatively limited sets (15 in [118], 50 in [115], and 64 in [112]).

- Data Distribution Technique: Our system adopts the Non-IID data distribution technique, to better reflects the heterogeneity present in real-world distributed datasets. In contrast, some other studies used an IID technique [94], which may not fully capture the diversity of real-world data distribution.

- Learning Approach: While many works adopted the centralized FL approach, we have chosen a decentralized learning approach due to its various advantages, including the ability to bypass the SPOF issue.

- Communication Security: Several previous works have focused on ensuring communication security during the exchange of gradients, and some of them include [116, 94, 119]. However, only [97] took into account the resistance to quantum attacks. Nevertheless, the 2DF-IDS methodology goes beyond these approaches because we do not rely on an aggregate server or other third party. The inclusion of such central components in the system can introduce potential vulnerabilities and single points of failure. In contrast, the 2DF-IDS framework is designed to provide multiple layers of security to safeguard FL-based decentralized IDSs. By eliminating the aggregation server and key generation center, we reduce the attack surface

and enhance the overall system's security. This decentralized nature contributes to a more robust and resilient framework against potential threats.

6.3.4 Discussion

In the context of the results obtained from our experiments, 2DF-IDS has demonstrated its robustness and effectiveness. However, one potential area for improvement in the proposed system lies in enhancing its security against federated poisoning attacks. The decentralized learning session can then be infiltrated, which could eventually lead to the entire model producing unfavorable and possibly harmful results. Implementing a robust authentication system to confirm the identity of each participating client prior to allowing them to participate in the training session is one way to mitigate this kind of attack. However, this method is less dependable because even authenticated clients could be exploited. To strengthen the defense against federated poisoning attacks, future work could investigate incorporating a defensive mechanism into each client's aggregation algorithm.

The trade-off between privacy and accuracy is a well-known limitation of privacy-enhanced FL approaches [154], as demonstrated in our performance evaluations. To address this, several strategies can be employed. One approach involves increasing the number of samples available to each client, if feasible, to enhance the overall model's performance while maintaining privacy to a certain extent. Data augmentation techniques, such as generating synthetic samples for each client, can also be utilized to augment the available data and improve model performance. Adversarial ML approaches like FAug [155] and FedHome [156] can aid in generating augmented data that enhances the training process. However, it is crucial to understand that there will always be some compromise between model performance and privacy protection, regardless of the strategy used. Striking the right balance among these factors depends on the specific implementation requirements and privacy considerations.

In conclusion, managing the privacy and accuracy trade-off is a significant challenge in FL. Various techniques, such as data augmentation and adversarial approaches, can be explored to optimize this trade-off based on the specific context and implementation choices.

6.4 Conclusion

Technology has a tremendous positive impact on agriculture and industry, but it also exposes those sectors to cybersecurity risks. To address this issue, we introduced 2DF-IDS, a secure and decentralized approach indented to protect such sectors. The proposed system enhances and secures the FL training process, safeguarding private data from external parties and securing the training process itself from internal participants. The system assessment showed its trustworthy efficacy in identifying various cyberthreats. It outperformed existing related FL-based IDS, with a near comparable accuracy to the centralized learning baseline (within a 0.47% deviation). Moreover, under strict privacy settings, it showed significant improvements. 2DF-IDS offers a promising solution for securing in-

dustrial IoT systems against cyber threats, and its decentralized nature provides added resilience and security compared to traditional centralized approaches

General conclusion

This thesis research successfully met its objectives. The aim of this chapter is to conclude the thesis by summarizing the research work performed and the corresponding contributions.

The implementation of smart agriculture practices can have numerous benefits for farmers and the environment. By utilizing advanced technologies and data analytics, farmers can optimize their crop yield, reduce costs, and minimize environmental impact. Smart agriculture can also help to promote sustainable farming practices, which are increasingly important in today's world. Overall, the benefits of smart agriculture demonstrate the potential to revolutionize traditional farming practices and improve food production efficiency for a growing population.

However, several security challenges may arise. These include data security, where with the use of sensors and other smart technologies, there is an increased risk of cyber threats. Farmers and food corporations need to guarantee that their data is secured, and that proper authentication mechanisms are in place to prevent unauthorized access. Also, privacy concerns, since the use of smart technologies in agriculture may collect personal information about farmers and their practices. It is crucial to ensure that farmers' and industrial food companies' privacy rights are protected, and that data collection is conducted in compliance with relevant data protection laws and regulations. Another issue would be infrastructure security, given that smart agriculture relies heavily on networked technologies, which can be vulnerable to attacks. Farmers need to ensure that proper security is in place. In addition, physical security involves the use of expensive equipment and machinery, which can be targeted by thieves. Food manufacturers need to ensure that physical security measures such as surveillance cameras, alarms, and fencing are in place to protect their equipment.

Overall, the implementation of smart agriculture requires careful consideration of security challenges and issues to maintain the protection of data, privacy, and infrastructure. This thesis aims to conduct a comprehensive investigation into the security and privacy issues that arise from the implementation of smart industrial agriculture. The primary objective is to propose novel security mechanisms that can account for the emerging technological trends in this domain, as well as the characteristics of the technologies employed, the need for sustainability, and the resource constraints of the devices involved. The proposed approaches aim to overcome the limitations of previous methods and enhance the security of smart agriculture by addressing key challenges related to privacy, data protection, and threat mitigation. .

In the first contribution, we suggested the use of SDN to safeguard blockchain-based

agricultural systems, including private supply chain infrastructures. The proposed security framework is comprised of three main components, namely a data management system, an integrity surveillance entity, and a network management component. Our proposed security solution has been evaluated has demonstrated its feasibility. To prevent cyberattacks on Agri-IoT infrastructures, our second contribution introduces FELIDS, a multilayered deep FL-based IDS. The suggested IDS is compact and has improved detection capabilities. The feasibility and efficiency of our suggested system are further illustrated by a thorough performance evaluation and comparative study of the proposed FELIDS model, the centralized ML model, and related works.

In the third contribution, we enhanced the security of the FL process by implementing safeguards for gradients. The proposed system is designed to be secure, decentralized, and differentially private, and is aimed at enhancing the security of IoT/IIoT networks. The participating clients' data privacy and confidentiality is ensured by two mechanisms, both within the learning session and from the outside. The first mechanism involves the use of a quantum-resilient R-LWE, while the second mechanism leverages DP. Additionally, the proposal employs a decentralized aggregation, to eliminate the SPOF risk associated with conventional FL. Several experimental evaluations were carried out on the 2DF-IDS using the Edge-IIoTset. These evaluations demonstrated the system's ability to perform robustly in securing such networks, and highlighted its superiority over existing related methods.

What is next?

Due to the possible financial losses, privacy and security are major issues in the agricultural industry. As with IoT, smart agriculture confronts security risks related to privacy, authentication, and access control. But also, unique problems with information management and storage. Protecting data gathering and ensuring the physical security of equipment are security concerns at the physical layer of smart agriculture. Although IDSs possess excellent detection capabilities when trained on high-quality data, they are not sufficient to guarantee security independently. For instance, an IDS may fail to detect zero-day attacks, which can be dangerous for the agriculture sector as it deals with sensitive data. Another severe issue is that ML is susceptible to poisoning attacks, particularly in cybersecurity-related applications, if not well secured. This can create opportunities for different types of attacks, potentially causing disastrous consequences for smart agriculture. To address this issue, future research should explore innovative techniques to mitigatesuch risks. For example, using clustering algorithms to identify clusters with high susceptibility to attacks or developing robust data pre-processing techniques to identify and eliminate suspicious data points could be potential research directions. Our future objectives are summarized below:

- *AI Adversarial Defense*: implementing robust models that are less susceptible to poisoning attacks. This can involve techniques such as training models on a more diverse range of data, developing new algorithms for detecting and mitigating adversarial examples, and creating new architectures that are specifically designed to be more resilient to attacks.

- *AI Explainability and Transparency*: Another important area of research in AI security is the development of more transparent and explainable models. This is especially critical in sensitive environments such as healthcare and finance, where it is crucial to understand how a decision was made. Future research can focus on developing techniques for making AI models more interpretable, such as building models with attention mechanisms or using generative models to generate explanations for their outputs.

- *Continuous Monitoring and Model Management*: Once an AI model is deployed, it needs to be continuously monitored for any suspicious activity or potential security breaches. Future research can focus on developing techniques for real-time monitoring of AI models and detecting anomalous behavior. Additionally, research can focus on developing tools for managing AI models, such as version control and continuous integration/continuous delivery (CI/CD) pipelines, to ensure that models are always up-to-date and secure.

- *Ethical Considerations and Human-Centered Design*: As AI becomes more prevalent in our daily lives, It is vital to take the usage's ethical ramifications into account. Future research can focus on developing human-centered design principles for AI systems that prioritize user privacy, autonomy, and fairness. This can involve developing frameworks for ethical decision-making in AI, as well as exploring the societal and cultural implications of AI systems. Additionally, research can focus on developing techniques for mitigating bias and ensuring that AI systems are fair and unbiased.

Bibliography

[1] U. Desa, "World population prospects 2019: Highlights," *New York (US): United Nations Department for Economic and Social Affairs*, vol. 11, no. 1, p. 125, 2019.

[2] F. FAO, "The future of food and agriculture: alternative pathways to 2050," *Food and Agriculture Organization of the United Nations Rome*, 2018.

[3] I. FAO, "More people, more food, worse water? a global review of water pollution from agriculture," *FAO/IWMI, Rome*, 2018.

[4] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, and X. Wang, "Internet of things for the future of smart agriculture: A comprehensive survey of emerging technologies," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 718–752, 2021.

[5] "Smart agriculture market by offering (hardware, software, services), agriculture type, farm size (large, medium, small), application (precision farming, livestock monitoring) and region (america, europe, asia pacific, row) - global forecast to 2028," https://www.marketsandmarkets.com/Market-Reports/smart-agriculture-market-239736790.html, last accessed 2023-03-02.

[6] "Jbs: Cyber-attack hits world's largest meat supplier," https://www.bbc.com/news/world-us-canada-57318965, last accessed 2023-03-02.

[7] "Fbi: Private industry notification (2021)," https://www.ic3.gov/Media/News/2021/210907.pdf.

[8] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *Journal of Network and Computer Applications*, vol. 149, p. 102481, 2020.

[9] F. AQUASTAT, "Aquastat-fao's global information system on water and agriculture," 2021.

[10] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, "From industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4322–4334, 2020.

[11] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, and A. F. Skarmeta, "Smart farming iot platform based on edge and cloud computing," *Biosystems engineering*, vol. 177, pp. 4–17, 2019.

[12] R. da Rosa Righi, G. Goldschmidt, R. Kunst, C. Deon, and C. A. da Costa, "Towards combining data prediction and internet of things to manage milk production on dairy cows," *Computers and Electronics in Agriculture*, vol. 169, p. 105156, 2020.

[13] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[14] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Aggoune, "Internet-of-things (iot)-based smart agriculture: Toward making the fields talk," *IEEE access*, vol. 7, pp. 129 551–129 583, 2019.

[15] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of iot in agriculture for the implementation of smart farming," *Ieee Access*, vol. 7, pp. 156 237–156 271, 2019.

[16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[17] H.-M. Huang *et al.*, "Autonomy levels for unmanned systems (alfus) framework volume i: Terminology version 2.0," 2004.

[18] E. Symeonaki, K. Arvanitis, and D. Piromalis, "A context-aware middleware cloud approach for integrating precision farming facilities into the iot toward agriculture 4.0," *Applied Sciences*, vol. 10, no. 3, p. 813, 2020.

[19] S. Singh, I. Chana, and R. Buyya, "Agri-info: Cloud based autonomic system for delivering agriculture as a service," *Internet of Things*, vol. 9, p. 100131, 2020.

[20] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955," *AI magazine*, vol. 27, no. 4, pp. 12–12, 2006.

[21] D. R. Vincent, N. Deepa, D. Elavarasan, K. Srinivasan, S. H. Chauhdary, and C. Iwendi, "Sensors driven ai-based agriculture recommendation model for assessing land suitability," *Sensors*, vol. 19, no. 17, p. 3667, 2019.

[22] I. Salian, "Supervize me: What's the difference between supervised, unsupervised, semi-supervised and reinforcement learning?" *Nvidia, Aug*, vol. 2, 2018.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] B. A. Ashqar, B. S. Abu-Nasser, and S. S. Abu-Naser, "Plant seedlings classification using deep learning," *International Journal of Academic Information Systems Research (IJAISR)*, vol. 3, no. 1, pp. 7–14, 2019.

[25] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[26] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[27] O. Friha and M. A. Ferrag, "Blockchain technology for 6g communication networks: A vision for the future," in *Cybersecurity Issues in Emerging Technologies*. CRC Press, 2021, pp. 77–96.

[28] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in agri-food supply chain management: A practical implementation," in *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*. IEEE, 2018, pp. 1–4.

[29] D. Popescu, F. Stoican, G. Stamatescu, L. Ichim, and C. Dragana, "Advanced uav–wsn system for intelligent monitoring in precision agriculture," *Sensors*, vol. 20, no. 3, p. 817, 2020.

[30] L. Hang, I. Ullah, and D.-H. Kim, "A secure fish farm platform based on blockchain for agriculture data integrity," *Computers and Electronics in Agriculture*, vol. 170, p. 105251, 2020.

[31] Y. Zhao, L. Liu, C. Xie, R. Wang, F. Wang, Y. Bu, and S. Zhang, "An effective automatic system deployed in agricultural internet of things using multi-context fusion network towards crop disease recognition in the wild," *Applied Soft Computing*, vol. 89, p. 106128, 2020.

[32] M. Muñoz, J. D. Gil, L. Roca, F. Rodríguez, and M. Berenguel, "An iot architecture for water resource management in agroindustrial environments: A case study in almería (spain)," *Sensors*, vol. 20, no. 3, p. 596, 2020.

[33] C. Kamienski, J.-P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T. Salmon Cinotti, R. Filev Maia, and A. Torre Neto, "Smart water management platform: Iot-based precision irrigation for agriculture," *Sensors*, vol. 19, no. 2, p. 276, 2019.

[34] G. Kousiouris, S. Tsarsitalidis, E. Psomakelis, S. Koloniaris, C. Bardaki, K. Tserpes, M. Nikolaidou, and D. Anagnostopoulos, "A microservice-based framework for integrating iot management platforms, semantic and ai services for supply chain management," *ICT Express*, vol. 5, no. 2, pp. 141–145, 2019.

[35] R. S. Alonso, I. Sittón-Candanedo, Ó. García, J. Prieto, and S. Rodríguez-González, "An intelligent edge-iot platform for monitoring livestock and crops in a dairy farming scenario," *Ad Hoc Networks*, vol. 98, p. 102047, 2020.

[36] A. Goap, D. Sharma, A. Shukla, and C. R. Krishna, "An iot based smart irrigation management system using machine learning and open source technologies," *Computers and electronics in agriculture*, vol. 155, pp. 41–49, 2018.

[37] L. Barreto and A. Amaral, "Smart farming: Cyber security challenges," in *2018 International Conference on Intelligent Systems (IS)*. IEEE, 2018, pp. 870–876.

[38] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and privacy in smart farming: Challenges and opportunities," *IEEE Access*, vol. 8, pp. 34 564–34 584, 2020.

[39] K. Demestichas, N. Peppes, and T. Alexakis, "Survey on security threats in agricultural iot and smart farming," *Sensors*, vol. 20, no. 22, p. 6458, 2020.

[40] A. R. de Araujo Zanella, E. da Silva, and L. C. P. Albini, "Security challenges to smart agriculture: Current state, key issues, and future directions," *Array*, p. 100048, 2020.

[41] J. Nikander, O. Manninen, and M. Laajalahti, "Requirements for cybersecurity in agricultural communication networks," *Computers and Electronics in Agriculture*, vol. 179, p. 105776, 2020.

[42] N. Etemadi, Y. Borbon, and F. Strozzi, "Blockchain technology for cybersecurity applications in the food supply chain: A systematic literature review," *Proceedings of the XXIV Summer School "Francesco Turco"—Industrial Systems Engineering, Bergamo, Italy*, pp. 9–11, 2020.

[43] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and privacy for green iot-based agriculture: Review, blockchain solutions, and challenges," *IEEE access*, vol. 8, pp. 32 031–32 053, 2020.

[44] A. Yazdinejad, B. Zolfaghari, A. Azmoodeh, A. Dehghantanha, H. Karimipour, E. Fraser, A. G. Green, C. Russell, and E. Duncan, "A review on security of smart farming and precision agriculture: Security aspects, attacks, threats and countermeasures," *Applied Sciences*, vol. 11, no. 16, p. 7518, 2021.

[45] F. Cyber Task Force, "Cyber criminal actors targeting the food and agriculture sector with ransomware attacks," 2021.

[46] T. Drape, N. Magerkorth, A. Sen, J. Simpson, M. Seibel, R. S. Murch, and S. E. Duncan, "Assessing the role of cyberbiosecurity in agriculture: A case study," *Frontiers in Bioengineering and Biotechnology*, p. 742, 2021.

[47] O. Friha, M. A. Ferrag, L. Maglaras, and L. Shu, "Digital agriculture security: Aspects, threats, mitigation strategies, and future trends," *IEEE Internet of Things Magazine*, to appear.

[48] T. Alves and T. Morris, "Hardware-based cyber threats." in *ICISSP*, 2018, pp. 259–266.

[49] P. Ney, K. Koscher, L. Organick, L. Ceze, and T. Kohno, "Computer security, privacy, and {DNA} sequencing: compromising computers with synthesized {DNA}, privacy leaks, and more," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 765–779.

[50] M. M. Yamin, M. Ullah, H. Ullah, and B. Katt, "Weaponized ai for cyber attacks," *Journal of Information Security and Applications*, vol. 57, p. 102722, 2021.

[51] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual international cryptology conference*. Springer, 1999, pp. 388–397.

[52] Y. T. Kalai and L. Reyzin, "A survey of leakage-resilient cryptography," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, 2019, pp. 727–794.

[53] M. S. Eddine, M. A. Ferrag, O. Friha, and L. Maglaras, "Easbf: An efficient authentication scheme over blockchain for fog computing-enabled internet of vehicles," *Journal of Information Security and Applications*, vol. 59, p. 102802, 2021.

[54] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[55] M. A. Ferrag, L. Shu, O. Friha, and X. Yang, "Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 407–436, 2021.

[56] J. P. Anderson, "Computer security technology planning study. volume 2," ANDERSON (JAMES P) AND CO FORT WASHINGTON PA, Tech. Rep., 1972.

[57] ——, "Computer security threat monitoring and surveillance," *Technical Report, James P. Anderson Company*, 1980.

[58] E. Cole, *Hackers beware*. Sams Publishing, 2002.

[59] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer networks*, vol. 31, no. 8, pp. 805–822, 1999.

[60] K. Scarfone, P. Mell *et al.*, "Guide to intrusion detection and prevention systems (idps)," *NIST special publication*, vol. 800, no. 2007, p. 94, 2007.

[61] A. Milenkoski, "Evaluation of intrusion detection systems in virtualized environments," Ph.D. dissertation, Universität Würzburg, 2016.

[62] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[63] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE transactions on computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[64] F. Sadikin, T. van Deursen, and S. Kumar, "A zigbee intrusion detection system for iot using secure and efficient data collection," *Internet of Things*, vol. 12, p. 100306, 2020.

[65] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[66] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, "Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis," *IEEE Access*, vol. 9, pp. 138 509–138 542, 2021.

[67] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[68] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "Insdn: A novel sdn intrusion dataset," *IEEE Access*, vol. 8, pp. 165 263–165 284, 2020.

[69] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "Mqttset, a new dataset for machine learning techniques on mqtt," *Sensors*, vol. 20, no. 22, p. 6578, 2020.

[70] K. S. Gill, S. Saxena, and A. Sharma, "Gtm-csec: Game theoretic model for cloud security based on ids and honeypot," *Computers & Security*, vol. 92, p. 101732, 2020.

[71] M. Rabbani, Y. L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu, "A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing," *Journal of Network and Computer Applications*, vol. 151, p. 102507, 2020.

[72] G. S. Kushwah and V. Ranga, "Voting extreme learning machine based distributed denial of service attack detection in cloud computing," *Journal of Information Security and Applications*, vol. 53, p. 102532, 2020.

[73] A. Aldribi, I. Traoré, B. Moa, and O. Nwamuo, "Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking," *Computers & Security*, vol. 88, p. 101646, 2020.

[74] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2019.

[75] A. S. Almogren, "Intrusion detection in edge-of-things computing," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 259–265, 2020.

[76] W. Haider, N. Moustafa, M. Keshk, A. Fernandez, K.-K. R. Choo, and A. Wahab, "Fgmc-hads: Fuzzy gaussian mixture-based correntropy models for detecting zero-day attacks from linux systems," *Computers & Security*, p. 101906, 2020.

[77] S. Hosseini and B. M. H. Zade, "New hybrid method for attack detection using combination of evolutionary algorithms, svm, and ann," *Computer Networks*, p. 107168, 2020.

[78] I. H. Abdulqadder, S. Zhou, D. Zou, I. T. Aziz, and S. M. A. Akber, "Multi-layered intrusion detection and prevention in the sdn/nfv enabled cloud of 5g networks using ai-based defense mechanisms," *Computer Networks*, p. 107364, 2020.

[79] A. Derhab, M. Guerroumi, A. Gumaei, L. Maglaras, M. A. Ferrag, M. Mukherjee, and F. A. Khan, "Blockchain and random subspace learning-based ids for sdn-enabled industrial iot security," *Sensors*, vol. 19, no. 14, p. 3119, 2019.

[80] D.-M. Ngo, C. Pham-Quoc, and T. N. Thinh, "Heterogeneous hardware-based network intrusion detection system with multiple approaches for sdn," *Mobile Networks and Applications*, pp. 1–15, 2019.

[81] S. Sciancalepore, O. A. Ibrahim, G. Oligeri, and R. Di Pietro, "Pinch: An effective, efficient, and robust solution to drone detection via network traffic analysis," *Computer Networks*, vol. 168, p. 107044, 2020.

[82] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020.

[83] M. Zhou, L. Han, H. Lu, and C. Fu, "Distributed collaborative intrusion detection system for vehicular ad hoc networks based on invariant," *Computer Networks*, p. 107174, 2020.

[84] M. A. Ferrag, L. Shu, H. Djallel, and K.-K. R. Choo, "Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0," *Electronics*, vol. 10, no. 11, p. 1257, 2021.

[85] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks," *Future Internet*, vol. 12, no. 3, p. 44, 2020.

[86] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and grey wolf optimization for iot botnet detection," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–17, 2019.

[87] S. Murali and A. Jamalipour, "A lightweight intrusion detection for sybil attack under mobile rpl in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 379–388, 2019.

[88] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2019.

[89] J. Liu, W. Zhang, T. Ma, Z. Tang, Y. Xie, W. Gui, and J. P. Niyoyita, "Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection," *Expert Systems With Applications*, p. 113578, 2020.

[90] M. N. Kurt, O. Ogundijo, C. Li, and X. Wang, "Online cyber-attack detection in smart grid: A reinforcement learning approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5174–5185, 2018.

[91] A. Patel, H. Alhussian, J. M. Pedersen, B. Bounabat, J. C. Júnior, and S. Katsikas, "A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems," *Computers & Security*, vol. 64, pp. 92–109, 2017.

[92] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous sgd," *arXiv preprint arXiv:1604.00981*, 2016.

[93] K. Li, H. Zhou, Z. Tu, W. Wang, and H. Zhang, "Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning," *IEEE Access*, vol. 8, pp. 214 852–214 865, 2020.

[94] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, 2020.

[95] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "Dïot: A federated self-learning anomaly detection system for iot," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 756–767.

[96] I. Mohammed, S. Tabatabai, A. Al-Fuqaha, F. El Bouanani, J. Qadir, B. Qolomany, and M. Guizani, "Budgeted online selection of candidate iot clients to participate in federated learning," *IEEE Internet of Things Journal*, 2020.

[97] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

[98] X. Zhang, R. Lu, J. Shao, F. Wang, H. Zhu, and A. A. Ghorbani, "Fedsky: An efficient and privacy-preserving scheme for federated mobile crowdsensing," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[99] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, "When deep reinforcement learning meets federated learning: Intelligent multi-timescale resource management for multi-access edge computing in 5g ultra dense network," *IEEE Internet of Things Journal*, 2020.

[100] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[101] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Federated learning for data privacy preservation in vehicular cyber-physical systems," *IEEE Network*, vol. 34, no. 3, pp. 50–56, 2020.

[102] J. Yao and N. Ansari, "Secure federated learning by power control for internet of drones," *IEEE Transactions on Cognitive Communications and Networking*, 2021.

[103] C. Fang, Y. Guo, N. Wang, and A. Ju, "Highly efficient federated learning with strong privacy preservation in cloud computing," *Computers & Security*, vol. 96, p. 101889, 2020.

[104] Y. Xiao, Y. Li, G. Shi, and H. V. Poor, "Optimizing resource-efficiency for federated edge intelligence in iot networks," in *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2020, pp. 86–92.

[105] M. A. Rahman, M. S. Hossain, M. S. Islam, N. A. Alrajeh, and G. Muhammad, "Secure and provenance enhanced internet of health things framework: A blockchain managed federated learning approach," *Ieee Access*, vol. 8, pp. 205 071–205 087, 2020.

[106] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2019.

[107] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in sdn using machine learning approach," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2016, pp. 167–172.

[108] G. Cusack, O. Michel, and E. Keller, "Machine learning-based detection of ransomware using sdn," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 2018, pp. 1–6.

[109] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Applied Sciences*, vol. 8, no. 12, p. 2663, 2018.

[110] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K. R. Choo, and M. Nafaa, "Felids: Federated learning-based intrusion detection system for agricultural internet of things," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 17–31, 2022.

[111] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, 2020.

[112] W. Schneble and G. Thamilarasu, "Attack detection using federated learning in medical cyber-physical systems," in *Proceedings of the 28th International Conference on Computer Communications and Networks (ICCCN), Valencia, Spain*, vol. 29, 2019.

[113] T. T. Huong, T. P. Bac, D. M. Long, B. D. Thang, N. T. Binh, T. D. Luong, and T. K. Phuc, "Lockedge: Low-complexity cyberattack detection in iot edge computing," *IEEE Access*, vol. 9, pp. 29 696–29 710, 2021.

[114] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, "Intelligent intrusion detection based on federated learning aided long short-term memory," *Physical Communication*, vol. 42, p. 101157, 2020.

[115] Y. Song, T. Liu, T. Wei, X. Wang, Z. Tao, and M. Chen, "Fda3: Federated defense against adversarial attacks for cloud-based iiot applications," *arXiv e-prints*, pp. arXiv–2006, 2020.

[116] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial iot," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5926–5937, 2020.

[117] I. A. Khan, N. Moustafa, D. Pi, Y. Hussain, and N. A. Khan, "Dff-sc4n: A deep federated defence framework for protecting supply chain 4.0 networks," *IEEE Transactions on Industrial Informatics*, 2021.

[118] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "Fed-iiot: A robust federated malware detection architecture in industrial iot," *IEEE transactions on industrial informatics*, vol. 17, no. 12, pp. 8442–8452, 2020.

[119] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, 2021.

[120] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[121] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2512–2520.

[122] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 267–284.

[123] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[124] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the internet of things: Research issues and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2018.

[125] "Hyperledger sawtooth," https://www.hyperledger.org/use/sawtooth, last accessed 2022-10-04.

[126] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[127] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[128] M. A. Ferrag, L. Maglaras, and H. Janicke, "Blockchain and its role in the internet of things," in *Strategic Innovative Marketing and Tourism*. Springer, 2019, pp. 1029–1038.

[129] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19 293–19 304, 2017.

[130] A. Tsipis, A. Papamichail, G. Koufoudakis, G. Tsoumanis, S. E. Polykalas, and K. Oikonomou, "Latency-adjustable cloud/fog computing architecture for time-sensitive environmental monitoring in olive groves," *AgriEngineering*, vol. 2, no. 1, pp. 175–205, 2020.

[131] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow *et al.*, "Onos: towards an open, distributed sdn os," in *Proceedings of the third workshop on Hot topics in software defined networking*, 2014, pp. 1–6.

[132] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[133] G. Fortino, C. Savaglio, G. Spezzano, and M. Zhou, "Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 223–236, 2020.

[134] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[135] E. Barka, S. Dahmane, C. A. Kerrache, M. Khayat, and F. Sallabi, "Sthm: A secured and trusted healthcare monitoring architecture using sdn and blockchain," *Electronics*, vol. 10, no. 15, p. 1787, 2021.

[136] G. Rathee, F. Ahmad, R. Sandhu, C. A. Kerrache, and M. A. Azad, "On the design and implementation of a secure blockchain-based hybrid framework for industrial internet-of-things," *Information Processing & Management*, vol. 58, no. 3, p. 102526, 2021.

[137] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[138] R. Mînea, "A study on privacy-preserving federated learning and enhancement through transfer learning," 2021.

[139] X. Qiu, T. Parcollet, J. Fernandez-Marques, P. P. B. de Gusmao, D. J. Beutel, T. Topal, A. Mathur, and N. D. Lane, "A first look into the carbon footprint of federated learning," *arXiv preprint arXiv:2102.07627*, 2021.

[140] N. Rodríguez-Barroso, G. Stipcich, D. Jiménez-López, J. A. Ruiz-Millán, E. Martínez-Cámara, G. González-Seco, M. V. Luzón, M. A. Veganzones, and F. Herrera, "Federated learning and differential privacy: Software tools analysis, the sherpa. ai fl framework and methodological guidelines for preserving data privacy," *Information Fusion*, vol. 64, pp. 270–292, 2020.

[141] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," *arXiv preprint arXiv:2007.03051*, 2020.

[142] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[143] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173*, 2019.

[144] Z. Zhang, M. Zhou, K. Niu, and C. Abdallah, "The effect of training parameters and mechanisms on decentralized federated learning based on mnist dataset," *arXiv preprint arXiv:2108.03508*, 2021.

[145] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 245–248.

[146] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[147] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[148] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao *et al.*, "Opacus: User-friendly differential privacy library in pytorch," *arXiv preprint arXiv:2109.12298*, 2021.

[149] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[150] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.

[151] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2010, pp. 1–23.

[152] J. Ding, S. Fluhrer, and S. Rv, "Complete attack on rlwe key exchange with reused keys, without signal leakage," in *Australasian conference on information security and privacy*. Springer, 2018, pp. 467–486.

[153] J. Ding, X. Xie, and X. Lin, "A simple provably secure key exchange scheme based on the learning with errors problem," *Cryptology ePrint Archive*, 2012.

[154] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[155] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018.

[156] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, 2020.