Secure Monitoring System for Industrial Internet of Things using Searchable Encryption, Access Control and Machine Learning

Jawhara Bader Aljabri

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Computing Science

College of Science and Engineering
University of Glasgow

October 2022

Abstract

This thesis is an alternative format submission comprising a set of publications and a comprehensive literature review, an introduction, and a conclusion. Continuous compliance with data protection legislation on many levels in the Industrial Internet of Things (IIoT) is a significant challenge. Automated continuous compliance should also consider adaptable security compliance management for multiple users. The IIoT should automate compliance with corporate rules, regulations, and regulatory frameworks for industrial applications. Thus, this thesis aims to improve continuous compliance by introducing an edge-server architecture which incorporates searchable encryption with multi-authority access to provide access to useful data for various stakeholders in the compliance domain. In this thesis, we propose an edge lightweight searchable attribute-based encryption system (ELSA). The ELSA system leverages cloud-edge architecture to improve search time beyond a previous state-of-the-art encryption solution. The main contributions of the first paper are as follows. First, we present an untrusted cloud and trusted edge architecture that processes data efficiently and optimises decision-making in the IIoT context. Second, we enhanced the search performance over the current state-of-the-art (LSABE-MA) regarding order of magnitude. We achieved this enhancement by storing keywords only on the trusted edge server and introducing a query optimiser to achieve better-than-linear search performance. The query optimiser uses k-means clustering to improve the efficiency of range queries, removing the need for a linear search. As a result, we achieved higher performance without sacrificing result accuracy. In the second paper, we extended ELSA to illustrate the correlation between the number of keywords and ELSA performance. This extension supports annotating records with multiple keywords in trapdoor and record storage and enables the record to be returned with single-keyword queries. In addition, the experiments demonstrated the scalability and efficiency of ELSA with an increasing number of keywords and complexity. Based on the experimental results and feedback received from the publication and presentation of this work, we published our third technical paper. In this paper, we improved ELSA by minimising the lookup table size and summarising the data records by integrating machine-learning (ML) methods suitable for execution at the edge. This integration removes records of unnecessary data by evaluating added value to further processing. This process results in the minimisation of the lookup table size, the cloud storage, and the network traffic, taking full advantage of the edge architecture benefits. We demonstrated the mini-ELSA expanded method on two well-known IIoT datasets. Our results reveal a reduction of storage requirements by $> 21\%$ while improving execution time by $> 1.39\times$ and search time by $> 50\%$ and maintaining an optimal balance between prediction accuracy and space reduction. In addition, we present the computational complexity analysis that reinforces these experimental results.

Acknowledgements

First and foremost, I want to thank Allah for giving me the patience and strength to accomplish the thesis finally.

I would like to extend my sincere gratitude to Dr Lito Michala for her immense knowledge, great suggestions, and endless support during my PhD journey. Her guidance and supervision were instrumental in the success of this project. Dr Lito, thank you for teaching me to be a good researcher and for supporting me with words of encouragement during the COVID-19 pandemic. I also extend my gratitude to Dr Jeremy for his insightful feedback and valuable meetings.

Last but not least, I would also like to thank my parents, who encouraged me whenever I encountered difficulties. Without them, this work would not have been possible. I extend my appreciation to my sisters and brothers for their love and support. Enormous gratitude goes to my dearest husband, Fahd Alharbi, for his endless love and unconditional understanding during my PhD. Fahd, thank you for being proud of me for what I have accomplished. I also extend a special thank you to my children for their love and patience.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

IIoT Industrial Internet of Things

SE Searchable Encryption

AC Access Control

MA Multi Authority

PEKS Public Key Encryption with Keyword Search

ABE Attribute Based Encryption

CP-ABE Ciphertext-Policy Attribute Based Encryption

CT Ciphertext

CR user credential

MA Machine Learning

CA Central Authority

AI Artificial Intelligence

Chapter 1

Introduction

1.1 Overview

The use of intelligent sensors and actuators to improve manufacturing and industrial processes is known as the industrial internet of things (IIoT) [1]. IIoT, sometimes called Industry 4.0 or the industrial internet, uses real-time analytics and intelligent machines to make the most of the data that "dumb machines" have produced in industrial settings for years [2]. The underlying tenet of IIoT is that intelligent machines are superior to people in data collection and real-time analysis, as well as at conveying critical information that can be utilised to make business choices more quickly and correctly[3].

To illustrate the difference between the Internet of Things (IoT) applications and IIoT applications, Mathur [4] states the former focus on the consumer, whereas the latter focus on industrial applications. Thus, the accuracy, precision, and the risk impact of application failure metrics of IIoT applications should be higher than in IoT applications. The IIoT applications deal with complicated processes and affect the lives of employees inside the factory, as well as the business resources and cost [5]. Furthermore, IoT applications do not have strict real-time requirements in a wide range of cases, but IIoT systems have strict requirements of the time synchronisation and regular communication [6].

According to Vitali Kalesnik, the Director of Research for Europe at Research Affiliates, there are few cases of IoT applications that have successfully contributed to production improvements [7]. Therefore, more researchers using IIoT are needed to improve the industrial processes. It has been predicted that IoT and IIoT will have 75.44 billion deployed devices by 2025 [8], and Accenture has predicted that the IIoT will add $ 14.2 trillion US dollars to the global economy by 2030 [9].

The IIoT can benefit the industry by improving the quality, traceability and integrity of industrial processes [10]. This improvement is achieved by installing sensors within the factory and across the production lines, to monitor and collect readings for various factors that can interrupt the production and degrade the production quality. These sensors are controlled by monitoring the systems responsible for collecting the sensors' data and extracting meaningful information. Monitoring Systems can prevent or minimise manufacturing errors and waste, and improve decision making, such as, temporarily stopping or slowing down the production line. As a result, they can improve and guarantee a high production accuracy for the factory [6].

Wireless sensor and actuator networks (WSAN) are another concepts used in smart factories to provide remote monitoring and control of machines to reduce equipment failures and improve industrial productivity. However, the main difference between WSAN and IIoT is that a WSAN consists of a network of wireless sensors and actuators connected to a central controller performing a monitoring task without connecting directly to the Internet [11]. In contrast, IIoT devices can directly share the generated data via the Internet, where a data centre can process and interpret the data to take specific actions [12].

Connected sensors and actuators enhance business intelligence initiatives by allowing businesses to identify inefficiencies and problems earlier and save time and money, especially in the industrial sector[13].

The applications of IIoT in case of the deep integration of industrialization and informatisation in many fields such as manufacturing can be classified into four categories [14]:

1. The first class include production flow, quality control and energy consumption to improve producing processes.

2. The operation and management oriented is the second one , such as the supply chain and enterprise decision management.

3. The third one is allocation and collaboration of resources , which includes collaborative manufacturing and customization technology.

4. The last one is remote maintenance and product traceability to manage the product life cycle and service optimization.

A typical example IIoT application is predictive maintenance where IIoT data analytics is used to avoid any errors or shut down of the factory [15]. Another example is a monitoring system which is to monitor the customers or facilities in the factory to improve resources, needs of its clients and quality of productivity. Therefore, the IIoT help industries by providing ubiquitous connectivity, efficient data analytics tools, and better decision support systems for better market competitiveness [16, 17].

In recent years, researchers have shown increased interest in the IIoT to build and design monitoring systems and architectures [18]. This is achieved by utilising IIoT's capabilities in collecting data, such as sensor data, and analysing it on the cloud [19].

Monitoring Systems in smart factories work by collecting data from multiple locations and analysing the data on the cloud. To reduce the maintenance of cloud infrastructure, cost and overhead, factories uses Infrastructure as a Service (IaaS) for their Monitoring Systems on the cloud. The IaaS are offered and managed by service providers such as Google Cloud, Amazon AWS and Microsoft Azure. As a result, IaaS introduces several concerns related to Monitoring Systems, which are data confidentiality and integrity, searching over encrypted data, and data access control (AC) [20].

Such cases necessitate either the owner analysing the data and providing the requested answer or making the data available to the requesting agent—either approach results in obstacles. In the first case, the owner must have the expertise to perform the analysis and be trusted to provide accurate results of the analysis [7].In the second case, problems related to security, GDPR regulations or privacy of sensitive commercial information are added. Continuous compliance is a proactive strategy for continuously supporting the standards established by frameworks and laws throughout business environments [21].

The continuous compliance solution for security compliance management is highly flexible and can be automated. This approach automates maintaining compliance with enterprise policies, regulations, and regulatory frameworks for managed IT services providers [22].

However, collecting and processing data on the cloud compromises the data privacy and security which may leak sensitive information [23]. As a result, these data-at-rest must be securely stored and processed on the cloud without compromising its security and privacy [24, 25]. This goal is challenging as it requires processing the encrypted data (not the plaintext) on the cloud. Moreover, introducing AC policies on these encrypted data and allowing specific users, such as third-party contractors, to access, query and request specific adds even a more significant challenge [26].

To highlight the need for data confidentiality and integrity in monitoring systems, the authors in [20] demonstrated how collecting air-quality-related data on unsecure servers can misinform the public or even mislead policymakers. The authors showed that any modification to the sensor data could lead to false-negative emergency alert or wrong decisions [27], such as triggering the evacuation alarm or stopping a production line. As a result, compromising the data confidentiality and integrity may results in significant monetary losses for the factory. Therefore, IIoT systems must protect data by encrypting the collected sensor data before storing it on the cloud [28, 29].

The degree of severity to violate privacy on the IIoT differs from that of the IoT. In other words, unauthorised access in the IoT system may lead to privacy problems such as data theft. While violating it on the IIoT may lead to a disastrous decision that is being forced or the entire system stops working [6]. There is no difference between the security issues that target the normal IoT and IIoT, but there is a big difference between the degree of severity in the event of a security breach [30]. In other words, authenticating an illegitimate device may cause a normal IoT system some problems such as the invasion of privacy while on an IIoT system it could become a vast disaster, for example disrupting the network or forcing the network to take a hazardous action. In consequence, the IIoT requires higher-level security that considers deadline requirements, nature of devices in the system, and a recovery technique in the case of an attack [31].

In addition to the previous security and privacy requirements, factories need to share data with third parties such as insurance or/and consulting companies, customers or/and employees. To control and manage data access, IIoT systems must deploy AC Mechanisms on the encrypted data on the cloud [32]. The previous security and privacy requirement in IIoT Monitoring Systems introduces the following challenges:

1. IIoT devices have limited resources such as low computational power, low power consumption and low storage. Therefore, deploying and running encryption algorithms on these devices may add significant performance overhead.

2. Searching the encrypted data on the cloud requires adopting and enhancing Searchable Encryption (SE) algorithms to work on both the cloud and IIoT devices.

3. The IIoT Monitoring System should fulfil the critical real-time requirement for IIoT Systems, which significantly affect the decision-making process. Thus, each component in the system should be optimised to reduce the overall execution time.

The privacy term in IIoT refers to protecting the confidentiality of collected data or preserving sensor location since detecting location information could be a security and safety risk. The lack of privacy preservation may lead to a security threat such as utility monitoring which is impacting the network process [33].

The three most essential components of Security are Confidentiality, Integrity and Availability, aka the CIA triangle. Confidentiality ensures that only authorised users can read the data of a system. The integrity provides that no changes made to the data, and the availability means that each service and data are available [34].

The security issues in IIoT are as follows: impact attack, secure communication, authentication, accountability, trust management [4]. According to [35], the main security concerns are authentication and access control because the presence of users with improper access rights can severely affect these systems. Further, certain specific attacks may have devastating outcomes in IIoT scenarios.

Cloud servers, however, are not entirely trustworthy. To ensure data privacy, sensitive industrial data must be encrypted before being sent to cloud servers [36]. Unfortunately, conventional encryption techniques complicate the retrieval of data from cloud servers [37].

SE is the most common search methods retrieve files using keywords instead of retrieving all the encrypted files back [12]. The data owners usually extract keywords from the data files to create encrypted index structure and subsequently outsources both the encrypted data and this index structure to the cloud [38]. To searching over encrypted data, the cloud server combines the trapdoors of the keywords with the index information and then returns the corresponding files to the data users [39]. Thus, if keywords are misspelled, incorrect results are returned [40]. Public Key Encryption with Keyword Search (PEKS) is the one of SE method used in IIoT application [41].

In Data Security, AC mechanisms are employed to dictate who has access to the data [42]. One of the AC methods is Ciphertext-Policy Attribute Based Encryption (CP-ABE) , in this scheme, the ciphertext is encrypted with an access policy and the key is associated with user's attributes [43]. The decryption condition is that the user's attributes have to satisfy the access policy in the ciphertext.

1.2 Thesis Statement

Industrial Internet of Things (IIoT) systems, such as monitoring systems, collect and analyse sensitive IIoT data (e.g., sensor data) to detect unusual events and provide critical insights. These systems must meet the real-time requirements of IIoT while coping with the low computational power and storage of IIoT devices without compromising the security and privacy of the data. This thesis proposes a system to efficiently and securely collect IIoT data while allowing authorised users and monitoring systems to securely search and analyse the IIoT data based on their privileges.

Thus, we hypothesise that by combining cloud-based Public Key Encryption with Keyword Search (PEKS) and edge-based Ciphertext-Policy Attribute Based Encryption (CP-ABE) through an edge computing architecture using statistical techniques to reduce data transfer, we can provide lower latency, lower network traffic, higher security, and privacy, without sacrificing accuracy with respect to identified state of the art encryption approaches in IIoT applications as baseline for comparison.

1.3 Research Questions

Leading from the hypothesis we identify four research questions. The research questions in this thesis are:

- **RQ1:**. How to adopt and deploy a lightweight version of PEKS algorithm on both the IIoT devices and the cloud to achieve a near real-time performance in our framework that is suitable for time sensitive IIoT systems?
- **RQ2:** How can we introduce, investigate, and evaluate the combination of PEKS and CP-ABE mechanisms in the cloud versus edge architecture to the proposed framework while achieving sufficiently performance for time-sensitive IIoT systems?
- **RQ3:** How to investigate the performance overhead for deployment on the edge vs the cloud server on various IIoT applications, such as Power Planet Systems?
- **RQ4:** How to design and develop a framework with efficient execution time in the CP-ABE mechanism and a PEKS algorithm tailored to a suitable cloud and edge deployment for IIoT systems to provide a secure and privacy-preserving solution for IIoT systems with AC support?
- **RQ5:** How further optimise such a framework to efficiently reduce the volume of data traveling over the cloud and edge deployment?

To address both the hypothesis and the research questions we identify the following main research objectives:

1. Searchable encryption feature extraction through Machine or Statistical learning to extract features on the edge of the network and to improve the accuracy of SE, improve user profiles, and reduce latency.
2. Propose a system where the cloud execute the SE (store encrypted IoT data and search over it) and the edge to execute CP-ABE tasks (encryption/decryption) improving privacy and security through the proposed architecture and further improving latency.

1.4 Contributions

The contributions of this thesis are:

- We introduce an edge lightweight searchable attribute-based encryption system (ELSA), a cloud-edge architecture that optimises the efficiency of data processing and decision making in the IIoT context. This contribution addresses RQ1.
- Our key novelty is the introduction of keyword indices at the trusted edge alongside a query optimiser, which uses a clustering algorithm; this improves the efficiency of range queries, removing the need for linear search. This contribution addresses RQ2.
- We improve search performance relative to state-of-the-art lightweight keyword-based searchable encryption with multi-authority access (LSABE-MA) [44] by an order of magnitude. This contribution addresses RQ4.
- ELSA extension to support annotating records with multiple keywords in trapdoor and record storage and allowing the record to be returnable with single-keyword queries. This contribution addresses RQ3.
- ELSA scalability experiments demonstrate search time remains in the region of 10^2 ms for as many as 1000 keywords. This contribution addresses RQ3.
- Reduction of the memory requirements on the edge server by $> 21\%$ for two IIoT datasets represeting different uses cases.. This contribution addresses RQ5.
- Reduction of the volume and improvement of the quality of permanently stored data by evaluating the contribution of each data-point to the ML model's learning. This contribution addresses RQ5.
- Reduction of the search computational complexity by minimizing the size of encrypted records on the cloud and edge servers. This contribution addresses RQ5.

1.5 Publications

P1 Bader, J. and Michala, A. L. (2021) Searchable encryption with access control in Industrial Internet of Things (IIoT). Wireless Communications and Mobile Computing, 2021, 5555362. (doi: 10.1155/2021/5555362)

P2 Aljabri, J., Michala, A. L.,and Singer, J. (2022). ELSA: a keyword-based searchable encryption for cloud-edge assisted industrial internet of things. In proceedings of : The 22th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2022). (doi: 10.1109/CCGrid54584.2022.00035)

P3 Aljabri, J., Michala, A. L., and Singer, J. (2022). ELSA: Edge Lightweight Searchable Attribute-based encryption Multi-keyword Scalability. In proceedings of : The

2022 5th IEEE Conference on Dependable and Secure Computing (IEEE DSC 2022), **Recipient of the Best Poster First Place.**

P4 Aljabri, J., Michala, A. L., Singer, J. , and Vourganas, I. (2022). mini-ELSA: using Machine Learning to improve space efficiency in Edge Lightweight Searchable Attribute-based encryption for Industry 4.0. Submitted as pre-print at arxiv (http://arxiv.org/abs/2209.10896) and for review at ACM/IEEE IPSN 2023.

1.6 Organisation of the Thesis

The remainder of this thesis is structured as follows:

- **Chapter 2** presents a technical background of the studied area and a review of the related literature presenting [P1] and extending beyond it. There are a number of ways that IIoT has been addressed with respect to security, privacy, and efficiency. Next, the chapter presents ways that can be proposed to further improve security and privacy in the future of smart factories. It continues to discuss the challenges and open gaps in the latest secure IIoT technologies. Finally, it proposes a set of standards for security and privacy-preserving IIoT systems to enhance efficiency and deliver better IIoT applications.

- **Chapter 3** presents the cloud edge architecture IIoT, which uses a keyword-based searchable encryption scheme for multi-authority (MA) access control (AC) for IIoT devices supported by the 3-tier edge computing architecture called ELSA. Our main novelty is introducing a keyword index to the trusted edge and a query optimizer that uses a clustering algorithm. As a result, ELSA makes range queries more efficient and eliminates the need for linear searches. In addition, search performance is higher than state-of-the-art lightweight keyword-based searchable encryption with multi-authority access (LSABE-MA). This proposed system was presented in [P2].

- **Chapter 4** extends Chapter 3 by improving ELSA and presenting [P3]. This extension improves the ELSA search engine so it can more easily retrieve results that partially match the keyword set specified by the user. In addition, we have conducted further experiments to investigate scalability.

- **Chapter 5** presents machine learning pipeline integration at the edge server to reduce memory requirements on the edge server in ELSA. Moreover, minimise cloud storage costs, optimise network traffic over the entire stack and maintain high-quality and optimal quantity datasets. This system, namely mini-ELSA, was presented in [P4].

- **Chapter 6** gives a summary of the contributions and findings of this work. It revisits the thesis statement and explores potential research directions and future work. Finally, it summarises and highlights important concluding remarks.

Chapter 2

Searchable Encryption with Access Control in Industrial Internet of Things (IIoT)

This chapter discusses the previous studies regarding using common AC mechanisms with SE. Then, a review of state-of-the-art IIoT systems is presented in the context of SE and AC. This review establishes the gap in the literature and addresses the RQ. This chapter then extends by discussing (1) further work related to IIoT applications and cloud systems; (2) related work that discusses the data in IIoT context while highlighting the difference between this thesis and the related work; and (3) how IIoT systems adopt machine learning to improve the performance.

2.1 Overview

IoT has gained enormous popularity in the last decade, which consists of interconnected devices such as mobile phones, computers, sensors, and many more. These devices helped to develop and improve many sectors, such as Smart Cities, Smart Homes, and Healthcare [45].The significant improvement added to these sectors encouraged the industrial sector to introduce IoT into the manufacturing paradigm. As a result, this led to a new industrial revolution; Industry 4.0. A new term IIoT has been used to collectively refer to proposed IoT solutions in this space [14].

The applications of IIoT can be classified into four categories (Figure 2.1). The first class includes production flow, quality control, and energy consumption. This class aims to improve production processes. The second class is operation-oriented management, which includes supply chain and enterprise decision management. The third class focuses on the allocation and collaboration of resources. This class includes collaborative manufacturing and customization technology. Finally, the last class mainly focuses on product life cycle management. Additionally, it focuses on service optimization, such as remote maintenance and product traceability [14].

Operation and Management Oriented
Production Processes
IIoT application based Factories
Allocation and Collaboration of Resources
Remote Maintenance and Produc Traceability

Figure 2.1: IIoT application based Factories.

The growing population has led to an increasing demand for products, which has saturated the manufacturing industry; and even more so in the recent COVID crisis. As a result, the manufacturing segment is expected to have the highest and fastest-growing market segment by end-user at a compound annual growth rate (CAGR) of 27.94 %. To meet the growing demand, an efficient manufacturing system has become mandatory. This demand can only be achieved by the integration of the latest technologies, such as IoT within the manufacturing process [46]. However, the stringent regulatory requirements (COMAH, IEC, SIL) for safety must be satisfied for this shift to become usable in real-world applications.

To demonstrate the relevance of IIoT and its ability to meet regulatory requirements, we present two examples of manufacturers currently using IIoT schemes [47]. The first one is Airbus, the European aircraft manufacturer. Airbus currently integrates IoT technologies into its products and its workers' tools in the manufacturing process. Also, Airbus is harnessing IoT technologies to clear a backlog of orders and boost revenues. It is clear that IoT is transforming the aviation industry by enabling a more seamless passenger journey, increasing operational efficiency and driving a new age of 'connected aviation'. The second example is the global tech firm Client Global Insights (CGI). CGI has teamed up with Microsoft to deliver a predictive maintenance solution for elevators by leveraging IoT. CGI claims that it has developed a solution which securely connects thousands of sensors and systems within elevators, and monitors everything from motor temperature to shaft alignment. The data are collected and processed on the cloud using Microsoft's cloud-based Azure Intelligent Systems Service. These elevators enable technicians to use real-time IIoT data to spot defects and repair them before a breakdown occurs.

Thus IIoT, such as monitoring systems, help industries improve their resources and meet their clients' needs while ensuring high-quality production. IIoT achieves this by providing ubiquitous connectivity, efficient data analytics tools, better decision support systems and applications [48, 4]. As IIoT applications deal with complicated processes, these applications have a critical impact on several parties. For example, a failure in an IIoT application may put the employees' lives in the factory at severe risk. Similarly, the business resources may be at risk which has cost implications [6]. Therefore, the accuracy, precision, and risk impact of application-failure metrics of IIoT applications should be higher than in IoT applications. Moreover, IIoT applications must fulfil the stringent requirements of real-time processing and feedback, time synchronisation, and regular communication [7].

The three essential components of security are confidentiality, integrity, and availability, which are known as the CIA triangle. Confidentiality ensures that only authorised users can read the data of a system. Integrity ensures that no changes are made to the data, and availability means that all services and data are available [49]. Availability and integrity of data are considered more essential than confidentiality for industrial environments. This, however, does not diminish the need for confidentiality. With the internet-connected systems of IIoT, all three aspects should be brought up to an acceptable level. Thus, in the development of new IIoT and Industry 4.0 systems that leverage the existing network and cloud infrastructure, confidentiality and integrity should be weighed equally to availability [50].

IIoT applications, such as monitoring systems in smart factories, work by collecting data from multiple locations and analysing the data on the cloud. However, collecting and processing data on the cloud compromises the data privacy and security, leading to sensitive information leakage [24, 23]. Data-at-rest must be securely stored and processed on the cloud without compromising its security and privacy [24, 25]. This goal is challenging as it requires processing the encrypted data (not the plaintext) on the cloud. Moreover, some IIoT applications require AC policies to allow specific users, such as a manager or a third-party contractor to access and query the data. This AC requirements adds a more significant challenge [1, 26]. To highlight the need for data confidentiality and integrity in IIoT applications, the authors in [20] demonstrated how collecting Air-quality-related data on unsecure servers can misinform the public or mislead policymakers. The authors showed that any modification to the sensors' data can lead to false-negative emergency alerts or wrong decisions. An example of wrong decisions is triggering the evacuation alarm or stopping a production line. Therefore, it is crucial for IIoT applications to secure the collected data. This can be achieved by encrypting the collected data during transmission and at-rest [28].

The degree of severity in relation to violating privacy on the IIoT differs from that of the IoT. In IoT, unauthorised access may lead to privacy problems such as data theft. On the other hand, violating privacy on IIoT may lead to a disastrous decision that can cause the entire system to fail [6]. IoT and IIoT may share similar security threats. Yet, there is a substantial difference between the degree of severity in the event of a security breach in both IoT and IIoT [30]. In other words, authenticating an illegitimate device may cause a normal IoT system to experience some problems, such as privacy invasion. On the other hand, a similar scenario in an IIoT system could cause serious consequences. For example, disrupting the network or forcing the network to take hazardous actions. Thus, IIoT requires a higher-level of security. To do so, there are several factors to consider, such as the applications' requirements, the type of IIoT devices, and a recovery technique in the case of cybersecurity attack [31].

In addition to the previous security and privacy requirements, factories need to share the data with other parties, such as insurance or/and consulting companies, customers and/or employees. To control and manage data access, IIoT systems must deploy AC Mechanisms on the encrypted data on the cloud [32].

Several recent studies have addressed the security and privacy issues for IIoT, from different perspectives. For example, the authors of [28] categorised the security challenges for both IoT and IIoT. The authors also specified whether these challenges applicable to IoT or IIoT or both. On the other hand, the study demonstrates the security challenges in the IIoT and stresses the need to design practical solutions. Also, the study shows that various IIoT scenarios require application-specific designs. However, solutions to the challenge of appropriate designs are not suggested by the authors.

In [51], the authors analysed the security challenges of IIoT and provided a comparative analysis of the available solutions. This study set out to identify some open research problems related to system integration, communication, energy factor, preventive and detective measures, authorisation, and architecture of IIoT. However, the study does not suggest feasible and practical solutions. Similarity, Tange et al.[50] provide a systematic literature review of IIoT security requirements. The authors demonstrate how fog computing can address these requirements. Additionally, the authors identified some research opportunity to use secure fog computing for IIoT.

Building on existing findings from [28, 51, 50], in this article, we examined the practical considerations of embedding security and privacy solutions to IIoT system architectures moving away from the cloud paradigm to minimise exposure to threats. Thus, we focus on combining SE and AC methods in a cloud-Edge architecture to assess their suitability and efficiency from the privacy, security, and response time perspectives.

2.2 Literature Review

We focused on several topics, including security in IIoT, enhanced SE algorithms, a combination of SE, and AC methods, and ML in IIoT applications. The search keywords alongside the number of results are presented in Table 2.1. The following subsections discuss the related topics to this thesis.

Table 2.1: Results of the Literature Search

Topic	Online library	Number of Results
Industrial Internet of Things Applications	IEEE	2,845
	Springer	44,937
	Google search	53,000
	Google Scholar	100,000
Searchable Encryption for IIoT	IEEE	9
	Springer	8
	Google search	124
	Google Scholar	120
Access Control for IIoT	IEEE	101
	Springer	898
	Google search	5,600
	Google Scholar	5,000
Privacy Preserving for IIoT Applications	IEEE	19
	Springer	114
	Google search	1,000
	Google Scholar	1,110
Edge Computing in IIoT Application	IEEE	89
	Springer	483
	Google search	3,150
	Google Scholar	3,000
Requirements of IIoT	IEEE	238
	Springer	1,057
	Google search	5,000
	Google Scholar	4,580
Searchable Encryption with Access Control	IEEE	105
	Springer	1,168
	Google search	2,300
	Google Scholar	10,000
Monitoring System using IIoT	IEEE	125
	Springer	967
	Google search	5,200
	Google Scholar	15,100
Security of IIoT Applications	IEEE	197
	Springer	853
	Google search	4,600

	Google Scholar	5,030
Machine learning in IIoT	IEEE	121
	Springer	654
	Google search	2,450
	Google Scholar	15,030

2.2.1 State-of-The-Art in IIoT with Embedded Security Mechanisms

IIoT systems can benefit from the massive amount of collected data to generate a useful approach. This approach can improve the performance of the system and minimise unplanned downtime [52]. IIoT systems utilise cloud servers to store and process the generated massive data [52]. However, the data need time to be transferred to centralized data centres, which degrades the IIoT system efficiency. This implies that processing data on an Edge server could help the IIoT system meet real-time requirements and reduce the decision-making latency [53]. The survey presented in[28], identified two constraints when protecting data confidentiality in IIoT systems through data encryption. One of these constraints is related to the limited resources of IIoT devices.

Gebremichael et al. [31] described the privacy challenges in IIoT based on the levels of the architecture as follows: device, platform, and application layers. The solutions provided AC methods, authentication mechanisms, data encryption, and secure channels to ensure the privacy at the device layer. For example, they protected Edge nodes against a fake node insertion attack. They also describe several points that developers need to consider when designing privacy solutions for the IIoT. These points can be described as follows:

- Cryptographic mechanisms are generally employed to enforce privacy policies. The challenge is to design a lightweight privacy-enhancing cryptosystem suitable for IIoT devices. These IIoT devices have limited resources. Thus, it is crucial to prevent heavy computations to meet the IIoT real-time requirement.

- Further research is needed to provide lightweight cryptosystem solutions with anonymised data methods. Also, advanced data analytics tools to process the collected data.

- Reducing the amount of data collected by Edge devices to the minimum data points that are required for system operations while continuing to provide anonymisation techniques on user data.

- Illustrating data access policies and implementing appropriate AC methods that are capable of identifying authorised users that have access rights to Edge node data.

Several solutions can protect IIoT systems' privacy, such as encryption, AC, processing data on the Edge, and anonymisation. Privacy in IIoT systems is challenging as these systems usually store and process data in third-party cloud services.

Yu et.al. state that the data generated by IoT devices has increased dramatically. At the same time, Cisco predicted that the volume of data generated by IoT devices would reach 500 ZB by 2019 [54]. However, massive data need time to be transferred directly to the cloud for processing, which adds computation overhead. This computation overhead increases the latency, bandwidth and may even lead to the unavailability of IIoT applications [14]. To address this issue, the concept of Edge/fog computing has been defined, and data can now be processed much closer to the source. This is because some cloud services are brought to the Edge of the network. In this context, fog computing differs from Edge computing in that it uses the interconnection between end-points. Edge computing, on the other hand, focuses on isolated end-points [54]. This implies that processing data on the Edge server helps the IIoT system to meet the real-time requirement and reduce decision-making latency, especially for delay-sensitive applications [14]. Edge computing is applied to manufacturing based on IoT to meet these requirements [33].

Many researchers introduced improvements to SE algorithms that would make them lightweight for IIoT, such as [55]. Yet, this method is not tested for its applicability in industrial plants. Wazid et al. [34] review the access control in IIoT such as a monitoring system of an industrial plant. They state that authentication is the most important security requirement in cloud based IIoT while this requirement is still needed to improve the proposed solution.

The following subsections will discuss data analtyics, SE, and AC state-of-the-art methods that have the potential to address the challenges identified in this subsection.

2.2.2 Edge Data Analytics for IIoT Applications

Data analytics is the most important step in the monitoring system's life cycle. IoT data analytics improves fault detection, disaster forecasting, service, and smart decision-making [35]. Moreover, they help the smart factory extract the knowledge from raw data with the support of IIoT applications. For example, to better understand technological enabler behaviour, or to relate issues derived from combined and statistical data processing [12]. The usage of feature extraction methods provides more accurate data analysis results. Besides, meeting the real-time requirement for IIoT manufacturing applications. For instance, a robust incremental feature extraction method based on PCA (Principal Component Analysis) is proposed to meet the real-time requirement [35]. Extracting data features from the data by applying such techniques allows Edge servers to take smart decisions for delay-sensitive applications [33].

Applying Edge analytics directly reduces the volume of data to be transmitted to the cloud. This, in turn, reduces the information that must be encrypted, which makes the encryption overhead minor. However, this reduction introduces other challenges in terms of accuracy and traceablity, especially in regard to the route cause fault finding capabilities. Thus, appropriate Edge data analytics methods must be identified to optimise the trade off between benefit and side-effects.

2.2.3 Searchable Encryption

SE is a cryptographic technique that allows secure searching over encrypted data [38]. SE allows a user (or an automated program) to perform a secure query for a specific event without compromising the data confidentiality. For example, using SE to encrypt data on the cloud, prevents the cloud provider or any unauthorised person (including the system administrator) from accessing or querying the encrypted data. There are two SE schemes [39]; one of these schemes is symmetric searchable encryption (SSE). SSE requires a private key to be distributed between users, which is not suitable for multiple user scenarios [40]. The other scheme is PEKS [56]. PEKS is a public-key cryptosystem that allows search over encrypted data using a public key instead of private keys; allowing multiple parties to query the data without compromising the data owner's private key.

2.2.4 Access Control Methods

There are several known AC mechanisms, including but not limited to Attribute-based, Key-Policy Based, Role-based and Trust-based [57]. However, the most commonly used AC mechanisms with PEKS are Role- and Attributes-based access control. Table 2.2 summarises the difference between these two AC mechanisms, based on two recent publications [58],[59]. The following subsections will further discuss those approaches and their capacity to be combined with SE and critically compare them in the context of IIoT.

2.2.4.1 Role Based Access Control (RBAC) with PEKS

RBAC is a security mechanism that allows users to access data based on their roles within an organisation [42]. The authors in [60], introduced RBAC to PEKS using free bi-linear, as bi-linears have high computational cost. The authors used the RBAC mechanism to simplify the frequent user's permission assignment within a large organisation. However, using RBAC with PEKS makes it hard to manage third parties' access policies (users outsides the organisation), which is an essential requirement for a monitoring system in the IIoT. Besides, using RBAC with PEKS is inflexible as it must be painstakingly managed.

Table 2.2: Comparing Role and Attributes-based access control

	Access Control Mechanism	
Feature	Role Based Access Control (RBAC)	Attribute-Based Access Control (ABAC)
Access control granularity	Coarse-grain access control	fine-grain access controls
User addition mechanism	Creating access control groups defined as roles with pre-setup privileges. Users can be added into the group for their desired access privileges.	Users are assigned attributes to describe their properties. The access control system needs to focus on the required access control policies that are described by a set of attributes to check the user's privileges to decide if the access should be granted or not.
Structure of access policy	Policies are assigned (operation/object pairs) to groups before the access request is made.	Using Boolean rule structure to express the policies.
The input of authorization decisions.	Users are assigned to roles and inherit the permissions assigned to the roles they have. Roles are often organised in a role hierarchy, which defines the inheritance of permissions between roles.	They are used as input for Authorization decisions with many criteria, such as department, job code, time of day, IP address, and user location.
Decision level	Only related to functionality.	Relate to access in both the data level and the field level, but also to functionality.
Access level	Do not allow access for non-employees to organization assets.	Allow limited access for third parties to organizational assets.
Model status	One of the main problems is that it is not an automatic model, needs to be painstakingly managed, and often involves significant manual intervention. The role-based mechanism, by itself, is inadequate to address the dynamic requirements of Cloud-Based IoT.	The ABAC model is a dynamic model. The system dynamically deploys access control by using attributes, i.e., a flexible access control approach.

2.2.5 Attribute-Based Encryption (ABE)

Attribute-Based Access Control (ABAC) is a security mechanism that allows organisations to grant access to users based on some attributes, such as their division or title [61]. In addition, Attribute-Based Encryption (ABE) combines searchable encryption with the ABAC approach [62]. In ABE, a message is encrypted for a specific receiver using a set of attributes. Thus, only the person who holds a key for the matching attributes can decrypt the message [42]. ABE has two paradigms: Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). In KP-ABE, the user's private key is associated with a specified access policy, and the ciphertext is encrypted under a set of attributes. The user can decrypt the ciphertext if the attributes in ciphertext satisfy the access policy in the user's key. Thus, KP-ABE mechanism answers the following question, "what type of data should the user access?". Differently, the CP-ABE answers the question, "What attributes must a user have to access the encrypted data?". Typically, CP-ABE is considered an adjustable scheme because it guarantees more control to the user over the encrypted data [63].

Rasori [63] improved ABE and reduced the communication overhead by 35 per cent compared with existing ABE for medical applications. This novel CP-ABE is more efficient and could be a suitable solution for low-power communication protocols in IIoT. Sathya and Kumar [64] proposed a medical system that collects patient's data during emergencies and shares the data with the doctors. The authors' proposed system combines blowfish encryption and an ABE scheme. The authors evaluated their proposed system using several symmetric encryption algorithms, encryption time, decryption time and total computation time. Their evaluation shows that the blowfish algorithm has better performance to encrypt data when used with CP-ABE to grant the authorised users' access to medical data. The main advantage of this work is the fast transmission of medical data, while the main disadvantage of using the blowfish algorithm is the linear relationship between the size of ciphertext and the number of attributes. When the number of attributes increases, so does the size of ciphertext.

Miao et al. [65] proposed a higher security level PEKS with CP-ABE approach, that supports access control with multiple permissions as well as hidden access policies. Also, the authors employed traceability techniques to prevent dishonest data users from leaking their private key to others. Their evaluations show that the computation costs for encryption, and decryption increase linearly as the number of user attributes does.

Yang et al. [43] proposed a system to monitor the patient's status with two AC modes. The first mode is for normal situations where the doctors, nurses and technical staff have access under an access policy. The second mode is for emergencies where the first-aider needs access to the patient's historical data. To achieve these controlled access modes, the authors applied ABE for normal access and break-glass algorithm for emergency access. However, their approach provides data security but does not provide a revocation mechanism to the emergency access policy, once the situation is resolved.

2.2.6 Attribute-Based Keyword Search (ABKS)

In the Attribute-Based Keyword Search (ABKS) scheme, the keywords are encrypted by an AC policy and the data with attributes. The user can generate a trapdoor that can be used to search over encrypted data [66]. The ABSE (Attribute based searchable encryption) scheme have exactly the contrary where the owner transmits the valid search query to the user and allow them to decrypt the data when its attributes satisfy the access policy [67]. However, ABKS schemes provide efficient search operations which allow retrieving encrypted data for multiple authorised users with flexible access policy [68].

Guo et al. [69] proposed a new ABKS to support encryption for both keyword and messages where most existing ABKS encrypt the keyword. In their proposed ABKS, there is no need for a secure channel to transmit the search tokens to the cloud. Also, it is a robust scheme against resisting offline keyword guessing attacks by inside attackers (i.e., the honest-but-curious servers). This scheme is evaluated and applied to a telemedicine system that is used to support health care services at multiple locations. However, the communication time in this scheme is high and is not suitable for time-sensitive applications.

2.2.7 Combining Searchable Encryption with Access Control

To achieve strong confidentiality, SE must be combined with AC [70, 71]: if a ciphertext appears as a search result, we learn something about the underlying document, even if the access control does not allow us to access the document. This illustrates the need for a linked search and AC, so that search results present to users only data to be accessed by the users [72]. Thus, the SE protects data confidentiality and AC schemes protect user access privileges [73].

It is essential to protect data that travels through the IIoT network. Thus, SE covers cryptographic protection across all networks by **(1)** protecting the Edge and cloud networking and **(2)** protecting endpoint connectivity [49]. Encryption techniques protect the privacy of big data in the data storage phase. Confidentiality, the first consideration when the encrypted data is stored in cloud servers, can be secured by efficient encryption techniques. However, when the data user sends the request to retrieve the data from the cloud, the cloud server cannot reply to the user's request, because it cannot decrypt the encrypted data or search over encrypted data. SE schemes could address these challenges.

While the Attribute-based Encryption (ABE) methods might secure information transmission and the fine-grained sharing of encrypted IIoT data, they additionally need to overcome new application deterrents in IIoT-cloud frameworks: **(1)** restricted resources IoT devices; **(2)** difficulty in encrypted data recovery at cloud servers. The encrypted records limit the adaptability and accuracy of information recovery, leading to unessential or incorrect outcomes; **(3)** lack of successful key administration: once CA is compromised, all previously encrypted files can be leaked because of the keys generated by a central authority (CA). To address the above difficulties, a novel lightweight searchable encryption method is needed for IIoT-cloud frameworks [44].

2.2.8 Searchable Encryption with Access Control in IIoT Applications

The literature survey of Zhou et al. [39], which spanned 2014 to 2019, identified schemes that combined PEKS with Attribute-Based Encryption (PEKS-ABE) for cloud-based applications. Moreover, this survey demonstrated that the PEKS-ABE provides efficient data sharing and searching ability, but it needs to improve the privacy of user keys. However, they do not also apply it to IIoT wherein to improve the privacy of the user keys an Edge processing and storage approach could be utilised.

The following two works focus on improving either SE or AC for IIoT environments, but they do not combine them. Chen et al. [55] proposed lightweight searchable encryption for cloud-based IIoT applications with security improvements. In [74], published in 2020, they improve CP-ABE in many aspects:

1. Using a hybrid cloud infrastructure. Public cloud to store encrypted IoT data and the private cloud to execute CP-ABE tasks over the data.
2. Guaranteeing data-privacy at the user level against the private cloud. The author achieved this by proposing two encryption techniques. These techniques work by protecting IoT data privacy at the item level and preventing the user-key leakage problem.
3. Enabling the private cloud to execute CP-ABE encryption/decryption tasks in batches. Also, executing the CP-ABE re-encryption tasks regardless of the size of IoT data. Thus, improving the performance of IIoT applications.

Chen et al. [55] proposed lightweight searchable encryption for cloud-based IIoT applications with security improvements. To achieve more precise data retrieval, Miao et al. [75] proposed an improved ABE scheme with multi-keyword search to support simultaneous numeric attribute comparison, thereby greatly enhancing the flexibility of ABE encryption in a dynamic IoT environment. Furthermore, attribute-based multi-keyword search schemes were also investigated in [76]. Nevertheless, this CP-ABE scheme inevitably concentrates on the single authority environment in which a CA essentially controls all attributes' authorisation. The single authorisation cannot effectively generate and manage the public/secret keys in the IIoT.

However, these studies did not improve the bandwidth of data that is outsourced to the cloud, which is important to minimise the computational cost.

Zhang et al. [44] proposed a lightweight SE-AC scheme by providing lower computational complexity. Moreover, their framework enhanced privacy by preventing leakage during data outsourcing to a cloud server. In summary, they provide fine-grained AC, multi-keyword search, lightweight decryption, and a multi authority environment. They provide low latency as well as improved security against the chosen-keyword attack and the chosen plaintext attack. Their LSABE and LSABE-MA schemes can support single keyword and multi-keyword searching while maintaining the lightweight decryption on many practical testing platforms (PC, mobile phone, Raspberry Pi models). Moreover, their schemes meet the low-latency requirement of IIoT applications. Therefore, their schemes are suitable for practical IIoT environments. However, their work did not consider the accuracy and data bandwidth, which is regarded as requirements of IIoT applications. In addition, the encryption time for their schemes is 24 seconds. Simultaneously, latency is an important metric in the encryption phase for the real-world IIoT environment. Thus, encrypted privacy-sensitive data must upload to the cloud immediately. Hence, we identify a gap in extracting useful information from the raw data before encrypting them to minimise the encryption time and the bandwidth and to improve the overall performance to meet IIoT requirements.

2.2.9 Latest Developments in IIoT and Cloud systems

One of the most developed and influential sectors today is the industrial IoT [77], which changes how industrial businesses operate. The industrial IoT runs and manages production processes, automates workflows, and gathers information about assembly-related activities. The monitoring IIoT system uses real-time sensor data to display cycle times, the number of components produced, downtime, and other beneficial indications [78]. The quality of the manufacturing process is monitored by examining the characteristics of calibrated equipment, machine settings, and ambient variables to check whether they deviate from typical values [79]. An IoT solution notifies a responsible party if such a situation is detected. This approach ensures the proper use of resources, prolongs equipment life, improves reliability, provides the maximum return on investment, and increases the effectiveness of production processes [80]. Although IIoT applications can adopt cloud platforms to process their data [81], the primary challenge when adopting cloud computing is preventing data leakage and ensuring IIoT data privacy [82]. Existing solutions process data in plaintext [83]; however, IIoT data must be managed and stored securely in the cloud without compromising their security, and privacy [24].

2.2.10 Data in IIoT Applications

To protect IIoT data on the cloud, several solutions have adopted SE with AC [84]. Applications based on the IIoT that use the cloud can employ encryption and AC. The survey [85] on using searchable encryption on the cloud illustrates the need to increase efficiency, query expressiveness, and scalability for access control-based searchable encryption. Efficiency is the most crucial issue that must be addressed in all searchable encryption systems, as no current encryption scheme can deliver the performance needed by practical applications. Another concern is scalability, because most searchable encryption solutions are inappropriate for large-scale databases.

Since the first paper was published [86], the work in this domain has grown and several solutions have been proposed. For instance, [87] introduced an access policy hiding attribute-based keyword search (ABKS) and a data-sharing scheme in cloud-assisted IoT. However, their scheme considers a static data set and provides no a mechanism to update data securely and dynamically. A certificateless public key authenticated encryption with keyword search model was developed by He et al [88] to address the security requirements of IIoT systems effectively. The researchers intended to lower the security scheme's computational and storage costs to promote industrial output growth. However, their system has low performance and ineffective searches.

Venkatesan et al. [89] implemented a lightweight cryptography model to eliminate the key management problem and utilise the forward privacy property to strengthen data security. Our review paper motivated the researchers to conduct this work, which is under review and has not been published yet. The novel contribution of their work is to reduce the high storage cost consumption and improve performance. However, their scheme considers the cloud a trusted environment for storing and retrieving data in IIoT systems and provides a linear search when searching all the data stored in the cloud.

2.2.11 Machine Learning in IIoT Applications

Many ML methods help industrial IoT devices analyse data more effectively, such as the K-nearest neighbours algorithm and linear regression [90].

The primary focus of supervised and unsupervised learning approaches is on data analysis issues [91]. The data type determines this categorisation and the ML approach to be used. Supervised learning is utilised when the desired outputs (labels) and the type of input data are known. In this case, the system is trained solely to map inputs to desired outputs. Regression and classification are two instances of supervised learning approaches, with classification using discrete outputs and regression using continuous outputs [92]. Several regression methods include support vector regression (SVR), linear regression, and polynomial regression [93]. ML algorithms employ mathematical approaches and large datasets to create behaviour models. ML makes it possible for computers to learn without being explicitly programmed. Based on recently input data, these models are utilised as foundations for creating forecasts [94]. Models are developed using ML to design, test, and train datasets. In addition, these ML algorithms can generate predictions based on the newly acquired data and are used to find potential patterns and similarities in massive datasets. The IoT and AI can combine to improve system analysis, accuracy rates, and operational effectiveness [95].

In the IIoT paradigm, ML techniques have been widely employed to evaluate massive data produced by many devices [96]. Manufacturing facilities have been transformed into highly optimised “smart” facilities by utilising the enhanced analytics offered by ML [96]. Categorising the collected data and identifying major critical events that should be scheduled as high priority for regular reviews are automated and based on extracting predictive patterns from historical data. We need to integrate ML methods in IIoT systems [3]. The IIoT may benefit from ML algorithms by saving costs and improving real-time response and performance [97]. For example, Bellavista et al. [98] applied ML techniques to extract valuable information and achieve real-time analytics for IIoT systems.

However, the collected IIoT data may provide a wide range of sensor readings. These readings can be domain-specific, which can be irrelevant or useless for some applications. This nondeterministic behaviour forces IIoT applications to waste computational time in unnecessary data pipelines. In addition, the collected data can contain noise and outliers, which can affect the availability and quality of manufacturing [99]. Moreover, the performance and accuracy of ML data pipelines are significantly affected by noise and outliers, jeopardising the quality and availability of manufacturing.

In this thesis, we develop a framework that will allow data miners to process the data collected from machines on the Edge. Through the use of ML, we can eliminate the unnecessary data points and improve the efficiency of the mining process.

2.3 Chapter Summary

Several studies have combined SE with AC to query encrypted data with different AC policies. However, studies that combined PEKS and AC mechanisms, such as CP-ABE, still suffer from low privacy for user keys, high volumes of data transmission, or a high ratio of error for returned data (reduced accuracy). Some studies combined these algorithms in the medical domain to improve the privacy of medical data. Also, to improve the security level against external and internal attacks. Furthermore, some systems still have a high computational cost, which is not practical for a computationally restricted environment such as IIoT. This high computational cost prevents studies from meeting the real-time requirement for the time sensitive IIoT applications. Therefore, IIoT applications must minimise the computational cost and improve performance to meet the near real-time requirements. Gebremichael et al. [31] discussed the further research that needs to be consider in the IIoT applications. The authors argue that using SE or homomorphic encryption (HE) can maintain security and privacy for systems that rely on cloud providers. Besides, SE provides fast and secure data delivery from the cloud for time-critical applications.

Chapter 3

ELSA: a Keyword-Based Searchable Encryption for Cloud-Edge Assisted Industrial Internet of Things

Aljabri, J., Michala, A. L., and Singer, J. (2022). ELSA: a keyword-based searchable encryption for cloud-edge assisted industrial internet of things. In the proceedings of : CCGrid '22.

Aljabri contributed the literature review, conceptualisation, implementation, results discussion, and writing of the first draft of the paper. Michala contributed to the conceptualisation, reviewed and edited, rewrote the contributions and contributed the threats-to-validity section. Singer reviewed and edited the draft and final versions.

In this chapter, we present an encryption method based on edge lightweight searchable attributes (ELSA). We offer an untrusted-cloud/trusted-edge architecture that maximises data processing and decision-making effectiveness in the context of the IIoT. As a result, we significantly improved search performance compared with the state-of-the-art (LSABE-MA). We employed the edge server to introduce a query optimiser and cluster data indices by keyword. Range queries are more practical for the query optimiser's use of k-means clustering, which eliminates the requirement for a linear search. We accomplished this improvement without compromising the results' accuracy.

3.1 Abstract

The Industrial Internet of Things (IIoT) plays a powerful role in smart manufacturing by performing real-time analysis for large volumes of data. In addition, IIoT systems can monitor several factors, such as data accuracy, network bandwidth and operations latency. To perform these operations securely and in a privacy-preserving manner, one solution is to use

cryptographic primitives. However, most cryptographic solutions add performance overhead causing latency. In this chapter, we propose an ELSA. ELSA leverages the cloud-edge architecture to improve search time beyond the state-of-the-art. The main contributions of this chapter are as follows. First, we present an untrusted-cloud/trusted-edge architecture, which optimises the efficiency of data processing and decision making in the IIoT context. Second, we enhance search performance over current state-of-the-art (LSABE-MA) by an order of magnitude. We achieve this by improving the organisation of the data to provide better than linear search performance. We leverage the edge server to cluster data indices by keyword and introduce a query optimiser. The query optimiser uses k-means clustering to improve the efficiency of range queries, removing the need for linear search. In addition, we achieve this without sacrificing accuracy over the results.

3.2 Introduction

The transformation of industrial manufacturing needs the support of data and systems technology to enhance the efficiency of manufacturing operations, improve product quality, and support smart decisions. This transformation through the Internet of Things has demonstrated significant improvements in sectors such as smart cities, smart homes and healthcare [100]. As a result, introducing IoT to the industrial sector led to a new industrial revolution; Industry 4.0. A term IIoT has been used to collectively refer to proposed IoT solutions in this space [45, 4]. Khan et al. [8] have predicted that IIoT will have 75 billion deployed devices by 2025 and Accenture has predicted that the IIoT would add $14.2 trillion US dollars to the global economy by 2030 [9].

A typical use case for IIoT is remote maintenance, involving product traceability to manage the product life cycle and service optimization [14]. In this case, factories may need to share data with third parties such as insurance, consulting companies, customers and employees, requiring different users to see different levels of information. Thus, accessing all the data might compromise the factory while differential access to some higher level of information might be beneficial [7]. To provide accessibility, data from multiple locations are collected and analysed on the cloud, often provided by third-party Infrastructure as-a-Service (IaaS) providers. Relying on IaaS providers raises concerns of data confidentiality, integrity, privacy and security. Cryptographic primitives such as AES and RSA are used in various implementations [101]. Often efforts concentrate on protecting the data-at-rest [24],[25]. This goal is challenging as it requires processing the encrypted data (not the plaintext) remotely in the cloud often through the use of SE [102]. However, intermediate steps of partial decryption often execute in the untrusted cloud environment. Thus, implementing an SE method with minimal leakage of unencrypted information is desirable. Additionally the search requires data to be in a predefined (partial) order. Moreover, to control and manage data access, IIoT systems must deploy AC policies [32]. This adds an even more significant challenge [1, 26]. Furthermore the keywords may not relate to the order the data is stored and are embedded in the ciphertext. Finally, all these challenges relate to the computation and power availability of IIoT devices, the real-time requirements imposed by the industrial processes, and the complexity of SE and AC algorithms.

3.2.1 Contributions

We introduce a cloud-edge architecture IIoT that utilises keyword-based searchable encryption multi-authority (MA) access control (AC) scheme for IIoT devices assisted by a three-tier edge computing architecture. As accidents of data leakage in cloud storage happen frequently and have been considered as one of the security issues in cloud storage [103], we expect the devices in the cloud to execute in an untrusted environment and only allow them to process encrypted data. We consider the devices executing in the edge 'tier' to execute in a trusted environment. Any processing that involves the raw data or partial decryption executes on the edge server [104],[105]. Our contributions are as follows:

- We introduce an edge lightweight searchable attribute-based encryption system (ELSA), a cloud-edge architecture that optimises the efficiency of data processing and decision making in the IIoT context.
- Our key novelty is the introduction of keyword indices at the trusted edge alongside a query optimiser, which uses a clustering algorithm; this improves the efficiency of range queries, removing the need for linear search.
- We improve search performance relative to state-of-the-art lightweight keyword-based searchable encryption with multi-authority access (LSABE-MA) [44] by an order of magnitude.

3.3 Related Work

The first SE scheme was introduced by Song et al. [106] and the first asymmetrical SE scheme was presented by Boneh et al. [107]. SE is the most common search method to retrieve files using keywords instead of retrieving all the encrypted files back. The subsequent SE schemes were designed to support a range of properties such as single keyword search [108] and multi-keyword search [109].

Zhou et al. [39] identified schemes that combined PEKS with Attribute-Based Encryption (PEKS-ABE) for cloud-based applications. They showed that PEKS-ABE provides efficient data sharing and search capability, though the privacy of user keys needed improvement. However, they do not also apply the work to IIoT. In this case an edge server could provide a more trusted and privacy-preserving method for processing and storage of transactions that involve the use of private user keys.

The following works focus on improving either SE or AC for IIoT environments, but they do not combine them. Chen et al. [55] proposed lightweight searchable encryption for cloud-based IIoT applications with security improvements. Qi et al. [74] improve Ciphertext-Policy Attribute-Based Encryption (CP-ABE) in the following ways:

1. They employ a hybrid cloud infrastructure. They propose a public cloud to store encrypted IoT data and a private cloud to execute CP-ABE tasks over the data.

2. They guarantee data-privacy at the user level against the private cloud. The authors achieved this by proposing two encryption techniques. These techniques work by protecting IoT data privacy at the item level and preventing the user-key leakage problem.

3. They enable the private cloud to execute CP-ABE encryption/decryption tasks in batches, while executing the CP-ABE re-encryption tasks regardless of the size of IoT data. Thus, they improve the performance of IIoT applications.

Chen et al. [55] proposed lightweight searchable encryption for cloud-based IIoT applications with security improvements. To achieve more precise data retrieval, Miao et al. [75] proposed an improved ABE scheme with multi-keyword search to support simultaneous numeric attribute comparison, thereby greatly enhancing the flexibility of ABE encryption in a dynamic IoT environment. Furthermore, attribute-based multi-keyword search schemes were also investigated in [76]. Nevertheless, this CP-ABE scheme inevitably concentrates on the single authority environment in which a central authority (CA) essentially controls all attributes' authorisation. The single authorisation point cannot effectively generate and manage the public/secret keys in the IIoT. However, these studies did not improve the bandwidth of data that is outsourced to the cloud, which is important to minimise the computational cost.

Moreover, many extensions of keyword-based searchable encryption work have emerged in recent years such as [110], [111]. They achieved decentralization by eliminating the central authority. However, their schemes suffer from high computational overhead and use expensive bilinear pairing operations. Given the storage space constraint of lightweight IIoT devices, it is a serious issue.

Zhang et al. [44] proposed a lightweight SE-AC scheme by providing lower computational complexity. Moreover, their framework enhanced privacy by preventing leakage during data outsourcing to a cloud server. This scheme provides fine-grained AC, multi-keyword search, lightweight decryption, and a multi authority environment. They provide low latency as well as improved security against the chosen-keyword attack and the chosen plaintext attack. Their LSABE and LSABE-MA schemes can support single keyword and multi-keyword searching while maintaining lightweight decryption on many practical testing platforms (PC, cell phone, single-board computers). Moreover, their schemes meet the low-latency requirement of IIoT applications. Therefore, their schemes are suitable for practical IIoT environments. However, their work did not consider the accuracy and data bandwidth, which is regarded as requirements of IIoT applications. In addition, the encryption time for their scheme may not be suitable for real-time application requirements. Simultaneously, latency is an important metric in the encryption phase for the real-world IIoT environment. This work is based on sequential search of all the encrypted data records. Thus, searching over the encrypted privacy-sensitive data uploaded to the cloud can introduce latency that does not adhere to the real-time requirements of IIoT. Hence, we identify a deficiency in the searching method and associated time as well as the bandwidth utilised and we aim to improve the overall performance to meet IIoT requirements.

3.4 Background

We first conduct a reproduction study of LSABE-MA [44] involving two components, a server and a client app. We then enhance this approach through use of an edge device that runs sketch algorithms and a query optimiser.

For the LSABE-MA scheme, the client app encrypts and transmits the encrypted data to the cloud by performing the following steps:

- Initialisation: The system is set up by the admin by taking as an input a security parameter κ. The outputs are the master secret key (MSK) and public parameters (PP)

  $$Setup(\kappa) \rightarrow (MSK, PP)$$

  The Global Setup generates the global identity GID beside the PP and MSK

- Secret Key Generation for AC: The authority setup algorithm for each authority A_j generates an authority attribute public key $APK_{i,j}$ and attribute secret key $ASK_{i,j}$ for each attribute i. The secret key generation utilizes the master secret key, public parameter, global identity, and attribute secret key as parameters to generate the secret key for each specific authority.

  $$SecKeyGene(MSK, i, GID, ASK_{i,j}) \rightarrow SK_{i,GID}$$

- Data Encryption: The data owner extracts the keyword set KW from file M to produce the ciphertext CT, containing the IIoT devices's reading and the encrypted keyword. The encryption process takes the following input: dataset M, access policy (A,q), keyword set KW, PP and the set of attribute public keys $APK_{i,j}$ for relevant authorities, to produce the ciphertext CT, which contains the encrypted secure index I and the encrypted file C_M.

$$Encrypt(M, (A,_P), KW, PP, APK_{i,j}) \rightarrow CT$$

- Data Transmission: the client sends the ciphertext to the cloud (server)
- Searching: the data user generates a trapdoor $T_{KW'}$ by using a set of keywords, PP and SK.
- Transformation Key: The data user also runs the transform key generation function, which takes the SK and a blind value z to generate the transformation key TK. The user then sends the TK with trapdoor query to the cloud.

The second part executes on the server. The server performs the following steps:

- Receive the encrypted data from the user
- Store the data in a database
- Receive a search trapdoor from the user and perform the search. The search function takes the trapdoor $T_{KW'}$ and the ciphertext as input. If the output of this function is ”0”, then the data was not found in the database. If the output of this function is ”1”, the cloud runs the transformation algorithm.

$$Search(CT, T_{KW'}) \rightarrow 0/1$$

- If the attributes included in transformation key TK satisfy the access policy in the ciphertext then the server runs the partial decryption on the result of the query using the transformation key.
- Return the result to the user.

When the user (the client) receives the data, they will decrypt it using the blind value z and the partially decrypted ciphertext to display it as an output.

$$Decrypt(z, CT_{out}) \rightarrow M$$

For these two parts the following features of the LSABE-MA were implemented: setup, key generation, encryption, trapdoor, transform key generation, search, transformation (partially decryption) and decryption.

3.5 ELSA

3.5.1 Architecture

As illustrated in Figure 3.1, our novel ELSA system leverages a cloud-edge architecture. We propose a lightweight searchable attribute-based encryption method on the edge.

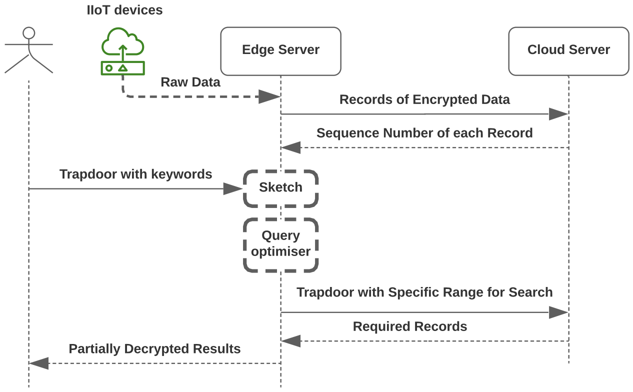
IIoT devices
Edge Server
Cloud Server
Raw Data
Records of Encrypted Data
Sequence Number of each Record
Trapdoor with keywords
Sketch
Query optimiser
Trapdoor with Specific Range for Search
Required Records
Partially Decrypted Results

Figure 3.1: Sequence diagram for ELSA which demonstrates the interactions between IIoT, Edge, Cloud, and the User.

In the original cloud-based search model, query time increases with the number of ciphertexts. If a large number of users access the ciphertext at the same time, the server can be blocked or even crash [112]. In LSABE-MA, this is induced by the linear search performed on the cloud server over the entire database. Our proposed optimisations remove the need for linear search through the introduction of clusters of data points relevant to existing keywords.

Further, a cloud-edge architecture is the most suitable platform for industrial applications. An edge computing entity can optimize the overall system computation, as it can be closely associated with IIoT objects while it can optimise the use of cloud services [113].

The ELSA method presented in Figure 3.2 utilises the cloud-edge architecture to process the query over the encrypted data where the edge is the trusted environment. The cloud server is responsible for interacting with the full encrypted dataset and only handles encrypted data (Figure 3.2 left). Moreover, the use of the edge server can reduce the load on cloud communication and meet IIoT privacy requirements, while reducing the latency for the user to acquire the requested data overall reducing core traffic. The edge (Figure 3.2 middle) is responsible for handling incoming requests from the user, eliminating queries that would yield no result (Bloom filter) and optimise queries that are propagated to the cloud. It also communicates directly with the IIoT devices and the data owner to establish the AC policies. Any partial decryption happens on the trusted edge server while user keys are only handled in this environment.

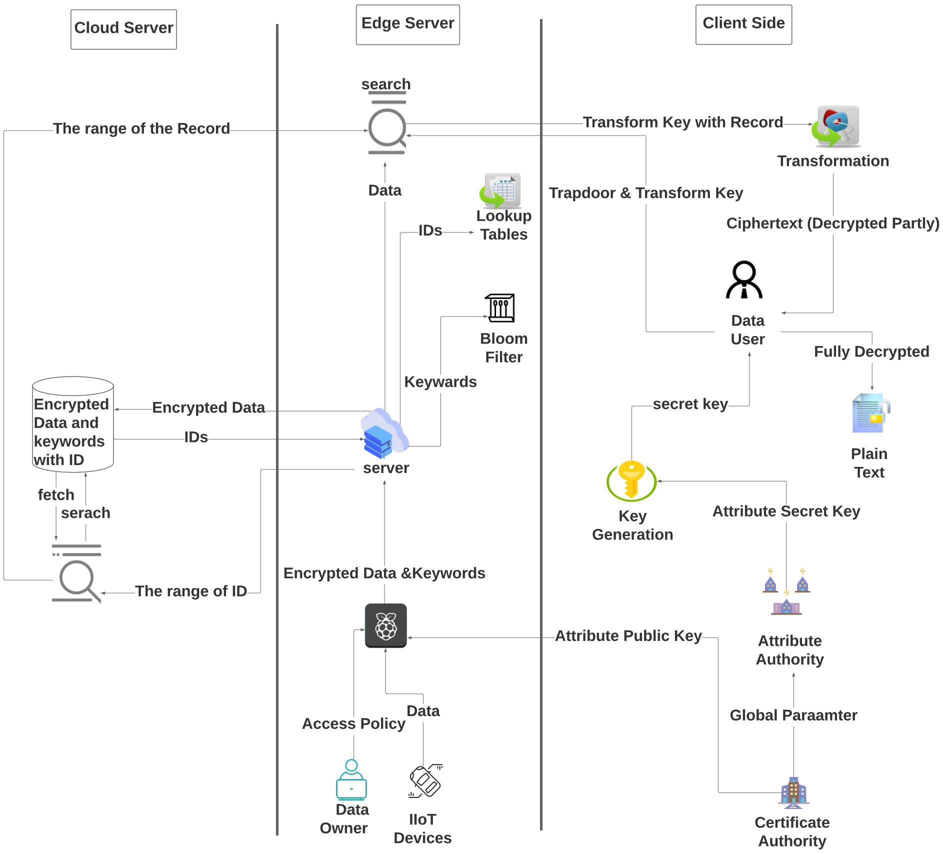

Figure 3.2: Proposed ELSA architecture separating domains of trust coinciding with data residence.

3.5.2 Generating Lookup Tables

To improve the performance, we enhance the architecture by introducing the edge server. The edge server is located within the smart-factory and introduces two significant optimisations to reduce latency of the encrypted data search. These are the sketch algorithms (probabilistic data structures), and the query optimiser which we describe further in this section. The workflow of ELSA is divided into 3 phases, as follows:

Phase 1: Generating keywords and ciphertext. The proposed architecture can collect raw data from various IIoT devices. The IIoT devices are connected to the edge Server. Regardless of the keyword generation mechanism (e.g., using feature extraction or thresholds), the edge generates the appropriate keyword set (KW) for each reading. The number of keywords associated with each reading is assumed to be predetermined at this point. For example in this paper we use the following: $\{CO_{2_{high}}$, $CO_{2_{low}}$,$CO_{2_{normal}}$, $humidity_{normal}$, $humidity_{high}$, $humidity_{low}$, $temperature_{normal}$, $temperature_{high}$, $temperature_{low}\}$. Then the edge encrypts the raw data with the keywords and appropriate user credential (CR) and generates a ciphertext (CT) as follows:

$$Encrypt(\{value, KW, sensorID, factoryID, CR\}) \rightarrow CT$$

Phase 2: Building and updating the lookup tables. The edge forwards the encrypted records to the cloud. The cloud then stores the encrypted record and generates a unique ID for each record. A unique ID for each encrypted record is sent back to the edge server. The edge is now responsible for maintaining the lookup tables which are updated whenever a new ID is received. The lookup tables store the ID and the associated keywords. The edge will also store the keywords in a sketch table which is a Bloom filter [114]. The sketch algorithm optimises access over summarised data through the use of estimated or probabilistic methods, in this case a Bloom filter [115]. The Bloom filter [116] is based on a membership approach to test if an item exists in the stored data or no.The Bloom filter simply answers the question of whether a specified item exists in the sketch or not. Thus, the sketch is used to determine whether the specified keyword exists in the database or not, which reduces the searching time. Finally the end user (Figure 3.2 right) is responsible for generating the initial queries bu generating a trapdoor, executes decryption based to generate plain text if and only if the AC allows them access.

Phase 3: Searching. To initiate a search over the encrypted data the user (or client) will need to creates a request through a trapdoor function parameterised by a keyword and credentials. This process is the trapdoor generation. When a client generates a trapdoor, it must first send the trapdoor along with the required keywords to the edge server. For each keyword, the sketch algorithm is executed to confirm if the keywords were found in the lookup tables. If not found, the edge responds to the client with zero results and indicates that no data is available for the given keyword. If found, the query optimiser is executed. The optimiser decides if a query should be handled as a range or point query. When a client sends a query to the edge, the edge will fetch all the IDs associated with the query (keywords). The optimiser will then decide if the query should be executed as a point or a range query. The IDs are then sent to the cloud and the cloud will only search within the specified IDs.

Range query. The optimiser uses the k-means clustering algorithm [117],[118] to cluster the IDs. It is more efficient to search over clustered data. When using non-clustered data, the server needs to do a *sequential* search over the entire data. On the other hand when the data is clustered, searching by a cluster index eliminates the sequential search.

In ELSA, data clusters are generated via the sklearn kmeans++ algorithm with default hyper-parameter settings. It is the simplest way to implement a lightweight clustering algorithm to deal with massive data reduction to centroids. K-means clustering is a popular unsupervised ML algorithm that groups n data items into k clusters. The user specifies the value of k. The centroid of each cluster is used to symbolise it. The first k records are normally utilised as the centroids in the first iteration. The remaining records are clustered based on the smallest ”distance” between each record and each centroid. The centroids are recalculated at the end of the first iteration which using the mean values of the attribute values for the records in each cluster. This step executes after each user query. Once the optimiser clusters the IDs, it sends the lower and upper bound of the IDs (for each cluster) to the cloud. For example, a cluster that contains IDs 1,2,5,7 will send the values 1 and 7 to the cloud. This reduces the query size. Note that the number of clusters has to be predefined, unless methods such as finding the optimal number of clusters can be adopted.

Point query. In some cases sending the lower and upper bound to the cloud uses more data than sending the IDs themselves. In this case, the optimiser will send the IDs to the cloud. For example, assume we have a query that requires the following IDs, 1,6,20,30 and the optimiser is required to find two clusters, i.e., cluster1 (1-20) and cluster 2 (30). This means that in a range query, the cloud will search in records from 1-20, and record 30. That is a total of 21 records. In a point query, the server will only search the exact records, i.e., 1,6,20,30 (total of 4 records).

After the cloud searches for the specified IDs (using the LSABE-MA search function), the matched required records are then sent to the edge. Note that in this case, the cloud will search within the specified IDs while ensuring that the user has the appropriate permission (which is embedded in the trapdoor) to access these records. Finally, the edge will receive the records from the cloud, partially decrypt the records and forward them to the authorized users. We discuss the sketch and optimiser components of the search phase in the next section.

3.5.3 Searching

As mentioned in the previous section, the edge is using the record IDs to build the lookup tables. Using the IDs provides several advantages. It allows the data to be sorted, it reflects the insertion order, and groups data together into clusters. So, the goal here is to identify a subset of data to search within instead of performing sequential search on the entire data volume.

The query optimiser algorithm 1 enhances the search process on the edge by clustering the data utilising their ID and their associated keywords using the k-means algorithm.

Based on the density of the data around the keyword centroid the optimiser selects if a range of data points or if a single data point corresponds to the user query. A range query is executed when there are several data points in a dense cluster around the requested keyword. A point query (or a set of point queries) is executed when there are few and space data points corresponding to the requested keyword. The point query is used to return unique records of required data and avoid returning unrequired records from the cloud within a set range. The range query is used to particularly when dense data is to be return that were inserted in the database in a sequential range of. This avoids sequentially searching through all saved records.

Algorithm 1 Query Optimiser

```
Input: $numOfClusters$, $IDs$ - The IDs to cluster
Output: Clusters ranges or IDs
function OPTIMISE($numOfClusters, IDs$)
    $clusters \leftarrow kmeanCluster(numOfClusters, IDs)$
    $itemsCount \leftarrow count(numOfClusters)$
    $idsCount \leftarrow count(IDs)$
    if $itemsCount \geq idsCount$ then:
        return $upperAndLowerBounds(clusters)$
    else
        return $IDs$
    end if
end function
```

Moreover, as illustrated in Figure 3.2, the edge server uses a sketch algorithm. The sketch algorithm in this case is the Bloom filter [116]. The Bloom filter works as follows: hash the key to be searched, then check all of the resulting array's index places. Return false if any bit is zero, indicating that the key is 'missing'. Return true if any bit is one, indicating that the key is 'found'. This algorithm reduces the overhead on cloud, in particular when the user runs a query that would return no results, while the size of this sketch on the edge is 158 bytes. Thus being very efficient in both execution time and resource use. However, the Bloom filter can introduce imprecision with the probability of a false positive being 0.01 in this case. If the number of unique keywords increase then the number of hash functions and the size of Bloom filter will also increase. To calculate the Bloom filter in ELSA, the number of unique keywords is nine, the probability of false positive is 0.01 and the number of hash functions is one. So, the size of Bloom filter is 100 bytes. This step can eliminate queries that would potentially return zero results. Thus, ELSA eliminates redundant queries from being propagated to the cloud and causing a linear search over the full database only to return an empty set of results. Thus a false positive would be initiating such a search which is not detrimental to returning the correct response to the user. It would only add some amount of unnecessary processing.

3.6 Evaluation

This section describes the evaluation setup, evaluation criteria, and presents the evaluation results. We evaluate ELSA (proposed method) by comparing its performance to the LSABE-MA scheme through reproducing the work presented in [44].

3.6.1 Evaluation Setup

3.6.1.1 Implementation and Dataset

In our evaluation, we first reproduced the state-of-the-art LSABE-MA [44]. The LSABE-ME [44] scheme and ELSA were implemented in Python 3.7. For the dataset, we used three different cases of data based on the percentage of representation for each keyword. These are the sparse dataset case, the medium density case and the dense case. The data consists of 200,000 unique temperature, CO_2, and humidity values. For evaluation purposes, we considered the $\{CO2_{normal}$, $CO2_{high}$,$CO2_{low}$, $humidity_{normal}$, $humidity_{high}$, $humidity_{low}$, $temperature_{normal}$, $temperature_{high}$, $temperature_{low}\}$ values as the keywords. This data is categorical, based on threshold values for ranges. The results are calculated by taking the average of 1000 runs for the three different cases. The dataset was synthetically generated as presented in Algorithm 2. For sparse data a probability of appearance was used to generate more data points associated with one keyword and very few data points associated with another; this is a more cumbersome task for the linear search approach in the LSABE-MA scheme.

Algorithm 2 Dataset generator

```
Input: $len$, $keywords$ - The keywords to associate with $len$ data points, $probabilities$ - percentage of representation for each keyword
Output: dataset
function GENERATE($len, keywords, probabilities$ )
    while $len - - \neq 0$ do
        $value \leftarrow uniform\_rand()$
        $keyword \leftarrow get\_next(keywords, probabilities)$
        $datapoint \leftarrow \{value, keyword\}$
    end while
    return $dataset$
end function
```

The specific trapdoor generated for the evaluation in each case was constant and used in both the LSABE-MA and our proposed method ELSA.

3.6.1.2 Architecture

Figure 3.3 illustrates the experimental setup used for LSABE-MA. We separated this scheme into two parts client and server. We run the client application on an edge device with Intel 2.3 GHz Core i9 processor and 16GB RAM. We deployed the server code on a docker container hosted on a DigitalOcean cloud provider (located in the UK). The plan for the cloud provider was CPU-Optimised, with 1 dedicated CPU, 2-32 vCPUs, 50 GB Memory, 2GB RAM/CPU and 2TB Bandwidth.

The following describes the procedure involved in evaluating LSABE-MA and our system (ELSA):

Setup and secret key generation - Client (1-2). The client setup and generates secret keys.

Encryption - Edge server (3). The client encrypts the sensor readings along with the appropriate keywords. The client then sends the encrypted data to the cloud, storing it on the cloud (step 4).

Searching - Client (5-6). The client generates trapdoors for the required keywords (query). Also, the client generates a transformation key used by the cloud to transform (partially decrypt) the encrypted results before sending them to the client.

Searching - Server (7-8). The server receives the trapdoor and transformation key and performs a search on the encrypted data. The results are then transformed (step 8) and returned to the client.

Decryption- Client (9). The client decrypts the data using the secret key only.

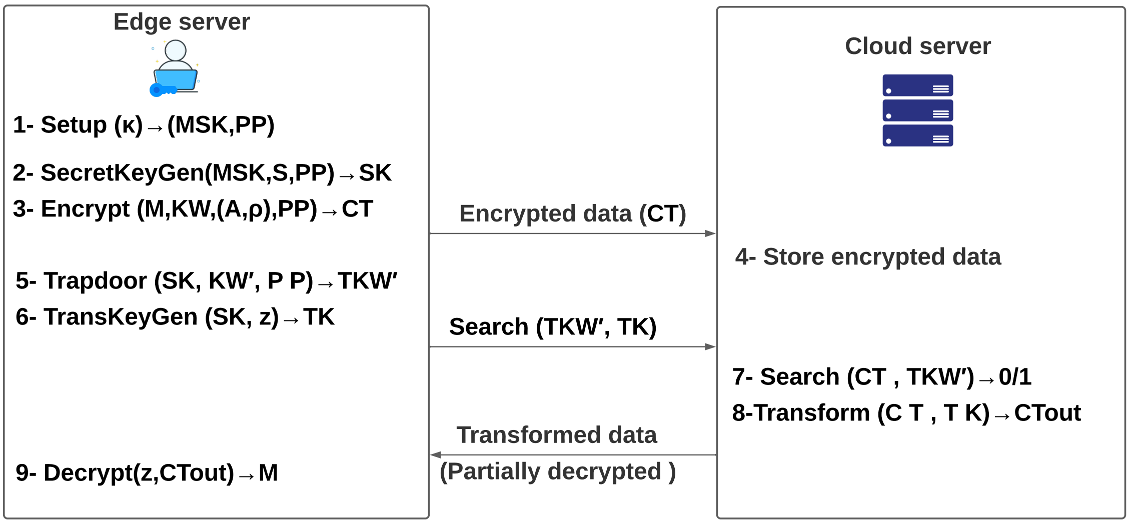
Edge server
1- Setup (κ) → (MSK,PP)
2- SecretKeyGen(MSK,S,PP) → SK
3- Encrypt (M,KW,(A,ρ),PP) → CT
5- Trapdoor (SK, KW′, P P) → TKW′
6- TransKeyGen (SK, z) → TK
9- Decrypt(z,CTout) → M
Encrypted data (CT)
Search (TKW′, TK)
Transformed data
(Partially decrypted)
Cloud server
4- Store encrypted data
7- Search (CT , TKW′) → 0/1
8-Transform (C T , T K) → CTout

Figure 3.3: Evaluation architecture of LSABE-MA where the user is directly requesting information from the cloud using the ELSA edge server as their end-point machine.

3.6.2 Criteria

Since our goal is to develop a secure, fast and accurate solution for IIoT data, we focus on the following evaluation criteria:

Performance. We evaluate the performance of LSABE-MA and our method ELSA by measuring the execution time for various functions. Specifically we measure the execution of the key generation, encryption, search and decryption.

Accuracy. We measure the accuracy of the query by comparing the encrypted results of ELSA method and LSABE-MA against a plaintext version. We use the well defined Precision and Recall metrics [119] to measure the accuracy of the search. *Precision* is the percentage of relevant records among the total retrieved records, which is defined as:

$$Precision = \frac{RRT}{RRT - IRT} \times 100\%$$

Recall on the other hand is the percentage of relevant records in relation to the correct records (in the database), which is defined as:

$$Recall = \frac{RRT}{RRT - RRNT} \times 100\%$$

where:

- RRT = Number Of Relevant Records Retrieved
- IRT = Number Of Irrelevant Records
- $RRNT$ = Number Of Relevant Records NOT Retrieved

3.7 Results

3.7.1 Secret Key Generation

Figure 3.4 demonstrates the key generation time for ELSA system and LSABE-MA by measuring the execution time in seconds. For both schemes, we measured the execution time for the function $(\mathrm{MSK}, i, \mathrm{PP}, \mathrm{GID}, \mathrm{ASK}_{i,j}) \quad \rightarrow \mathrm{SK}_{i,\mathrm{GID}}$. The results shows that there is no significant difference between ELSA system and LSABE-MA demonstrating that adding the edge server has no detrimental effect to the overall performance.

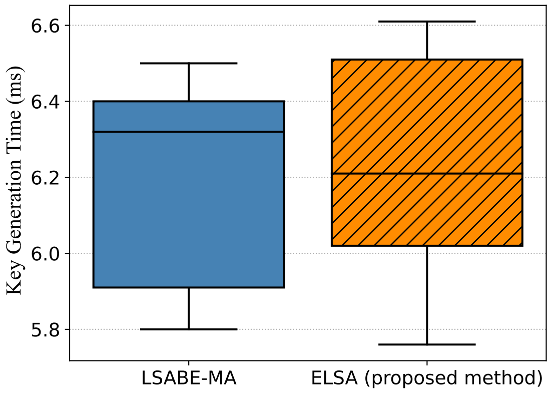

Figure 3.4: Key generation time measured over 1000 repetitions for each case, reported collectively.

3.7.2 Encryption and Decryption

Figure 3.5 and Figure 3.6 show the encryption and decryption time respectively for both our ELSA and LSABE-MA. We measure the execution time for both cases in terms of the function $(M, (A, \rho), \mathrm{KW}, \mathrm{PP}, \{\mathrm{APK}_{i,j}\}) \rightarrow \mathrm{CT}$. Decryption time for both approaches were measured using the same decryption function $(z, \mathrm{CT}_{\mathrm{out}}) \rightarrow M$. Again the introduction of the edge server has caused no detriment to the overall performance.

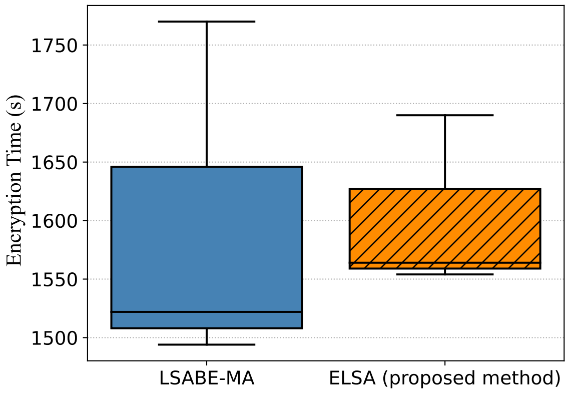

Figure 3.5: Encryption time for the full dataset measured over 1000 repetitions for each case, reported collectively.

3.7.3 Searching

To further evaluate the scalability of our approach, we measure the search time using various density datasets (shown in Figure 3.7). The searched keyword represents 50% of the data size in the Dense Data, 11% in the medium density and 5% in the sparse data. As shown in Figure 3.7, the searching time for LSABE-MA is consistent in all three cases, i.e., $386s$. ELSA attains $214s$ search time for the dense data, which is $1.8\times$ faster than LSABE-MA.

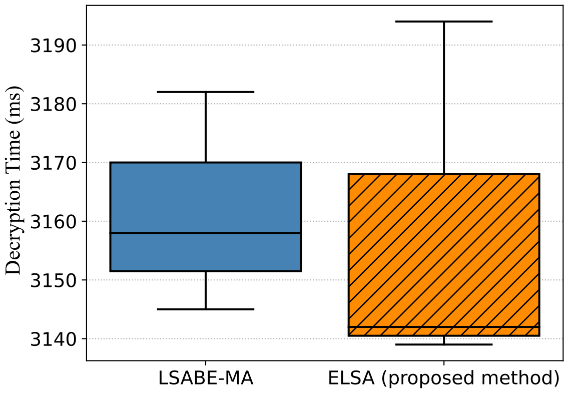

Figure 3.6: Decryption time of the returned query results measured over 1000 repetitions for each case, reported collectively. The results contain 100,000 datapoints in the dense case, 22,000 datapoints in the medium sparsity case and 10,000 in the sparse case.

ELSA has a $43s$ and $28s$ search time for the medium and sparse data, respectively. To conclude, ELSA achieves $1.8 - 14\times$ better performance than LSABE-MA. As the sparsity of the data increases the benefit of non-linear search is becoming apparent. In a dense dataset where the keyword repeats very often the search optimisation still returns a large range to be searched over. However, as the keyword appears less frequently in the sparse case, the smaller range queries combined with point queries significantly boost performance.

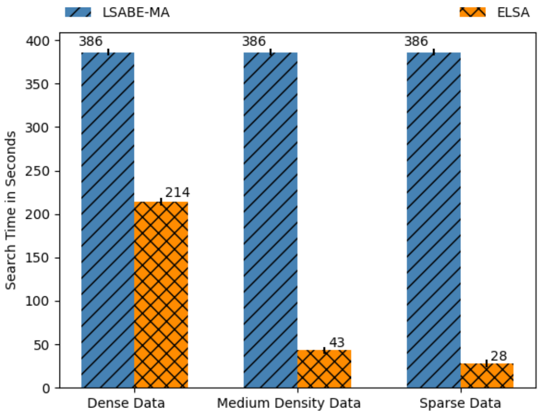

Figure 3.7: Search time for three different cases measured at 1000 repetitions per case.

3.7.4 Edge and Server Operations and Overhead Times for ELSA

In our proposed method ELSA, the encryption process has one additional step. This additional step is to store data in the lookup table on the edge. So, the overhead of this step is illustrated in Table 3.1 as load data time. It is evident that this time is insignificant compared to the overall execution time and did not affect the user because of the computational capacity available to our edge server.

In addition, we have two operations in the search process: sketch and clustering algorithms. The sum of the execution time for both these operations are less than 0.1 ms as shown in Table 3.1 once again being insignificant compared to the overall execution time.

Table 3.1: Mean overhead times on the edge server for three different cases.

Factors	**Overhead times for ELSA**		
	clustering(ms)	***data load(ms)***	***sketch(ms)***
Execution Time	0.05	32.71	0.00007

However, as ELSA system depends on several devices communicating we have further evaluated the execution time of each step in the process to identify bottlenecks and isolate the effect of networking or cloud operations that are beyond our control. In the Figure 3.8, we illustrate the overall execution time as well as details of the execution time for the various operations which take place on the edge and Cloud servers. These operations are the following: the total operations time as overall operation, encryption time, the overall operation of the cloud server, edge storage time, cloud server - communication time, and cloud server - storage time. It is evident that the encryption process is one of the main contributors to the overall latency.

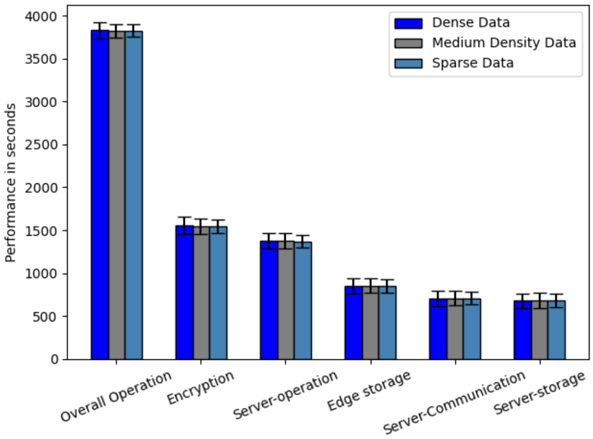

Figure 3.8: Time of perations in edge and server for ELSA measured at 1000 repetitions for each case, reported collectively.

3.7.5 Overall Execution Time

Figure 3.9 compares our ELSA system with LSABE-MA scheme in overall execution time. This graph aims to determine the overall effect of search time improvements to the latency experienced by the user from initial query to final result. As demonstrated, the ELSA system reduced the overall execution time by $1.21\times$.

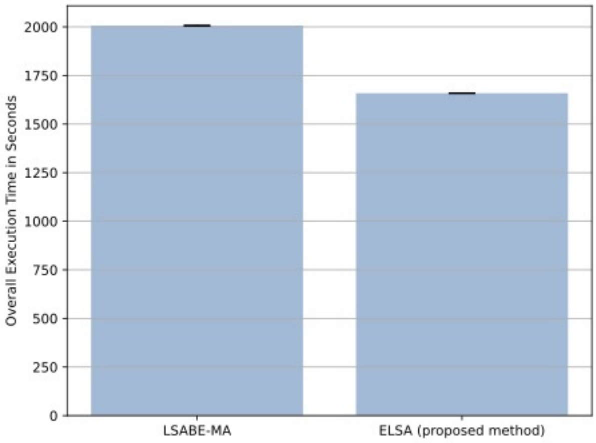

Figure 3.9: Overall execution time measured over 1000 repetitions for each case, reported collectively.

3.7.6 Lookup Table Size

To improve search time we have traded-off space used on the edge server to store the lookup tables. To investigate the potential limitation of ELSA we have measured the lookup tables size for our three cases. As presented in Figure 3.10, the size of the lookup tables for dense data, medium density data and sparse data are 2.39, 2.15 and 2.25 Megabytes respectively. This did not prove to be a limitation for our use cases but might need to be further investigated in future work to identify any adverse effect on scalability of ELSA for big datasets in the IIoT context.

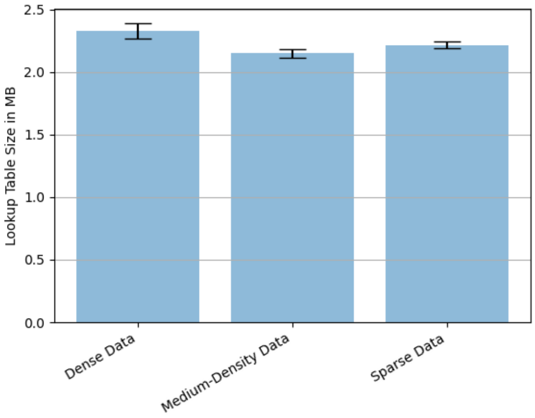

Figure 3.10: Lookup table size for three different data cases measured at 1000 repetitions per case.

3.7.7 Accuracy

To evaluate the accuracy, we performed two queries. The first query contains one keyword: (co2-high), while the second query contains two keywords: (co2-low,co2-normal). Both queries use different keywords. As shown in Table 5.1, both ELSA and LSABE-MA offer 100% precision and recall for both cases. As a result our additional operations and optimisations did not introduce a detrimental effect on the accuracy.

Table 3.2: Search precision and recall (accuracy) for our ELSA method and LSABE-MA demonstrating lossless database performance.

	Proposed ELSA		**LSABE-MA**	
Number of keywords	***1***	***2***	***1***	***2***
No. retrieved records	44,000	22,000	44,000	22,000
Precision	100%	100%	100%	100%
Recall	100%	100%	100%	100%

3.7.8 Comparison to Original LSABE-MA Results

In our implementation, we use one attribute, so we compare the results from our re-implementation with the original results from [44] for the one attribute case, which is the first point in their reported graphs. It is worth mentioning that their number of keywords is smaller (set to 5) compared to our re-implementation study.

In terms of encryption time, we find the LSABE-MA scheme takes 9.85s to encrypt the data, while in our Figure 3.5 we reported 1550s on average. However in our case we measure the encryption of the full dataset while from the original paper it is unclear if they report the encryption of a single entry (one datapoint being added). If a single datapoint requires 9.85s then our implementation of the encryption for our 200,000 points is orders of magnitude faster. However, we are unable to verify this assumption.

Figure 3.6 shows the decryption time of our replicated LSABE-MA, which took 3160ms on average. In [44], their scheme requires 400ms to decrypt the ciphertext, which, as they claim, does not significantly increase with the increased number of attributes. However, again it is unclear how many datapoints are returned in each case as it is dependent on the dataset, the keyword appearance frequency and the combination of keywords used for the query. As a result we cannot conduct any reasonable comparison between these two numbers. This is because, the authors of the original study did not explain the complexity of the dataset, the hardware configuration on the cloud and did not disclose the software they used.

From this analysis we can conclude that the large differences come from the number of records encrypted or decrypted in each case. Overall, it is unclear what is the size of the record encrypted in the original study, how large is the database that is searched over, and how many records are returned during the measured decryption step. Further it is unclear how long is the search time in the original study. Thus, no meaningful conclusions can be extrapolated from this comparison.

3.8 Threats to Validity

Because of lack of access to the original source code for LSABE-MA, we have developed a clean re-implementation of the scheme. This means that there might be subtle differences to the implementation as initially presented [44]. However, we followed the description presented in the paper and replicated algorithms and formalism to the best of our capacity.

The evaluation is performed on a single fixed deployment scenario. We feel this scenario is sufficiently representative to allow us to hypothesise that the trends identified in the results would generalise.

Further our evaluation focused on a limited set of queries and respective searches. We have tried to cover different data characteristics within those to identify trends in the performance of ELSA. In further work we aim to implement a wider variety of queries to identify potential scalability thresholds and limitations.

Finally, ELSA uses k-means as the method to restrict the scope of the query search on the cloud. However, other clustering methods may apply in this scenario. We intend to investigate these in future work.

3.9 Conclusion

In this chapter, we presented a new cloud-edge architecture for keyword-based searchable encryption with an optimised query process. Compared with the state of the art, our ELSA system is advantageous with respect to maintaining the encryption and decryption process and achieving more efficient data sharing and data searching. In terms of performance, the experimental results show that ELSA effectively reduces the search time by up to $14\times$ and overall performance by $1.21\times$. Compared to LSABE-MA, the ELSA system significantly reduces execution time and communication overhead by clustering the required data without sacrificing accuracy. This is achieved through the use of optimisations on the edge server that provide better than linear search performance with a trade-off in utilised storage space.

Chapter 4

ELSA: Edge Lightweight Searchable Attribute-Based Encryption Multi-Keyword Scalability

Aljabri, J., Michala, A. L., and Singer, J. (2022). ELSA: Edge Lightweight Searchable Attribute-based encryption Multi-keyword Scalability. In the proceedings of : IEEE DSC '22.

Aljabri contributed the literature review, implementation, results discussion, and writing of the first draft of the paper. Michala contributed to conceptualisation and reviewed, edited, and rewrote the contributions and draft and final versions of the paper. Singer reviewed and edited the final version.

In this chapter, we discuss how we enhanced ELSA to demonstrate how the number of keywords and ELSA performance are correlated. With this extension, records can be annotated with multiple keywords in trapdoor and record storage, and the record is made returnable via single-keyword searches. Additionally, the experiments revealed that ELSA is scalable and effective as the number of keywords and level of complexity increase.

4.1 Abstract

The digitalisation of industrial manufacturing requires the support of systems technology to enhance the efficiency of manufacturing operations, product quality, and smart decisions. This digitalisation can be achieved by IIoT. IIoT has played a powerful role in smart manufacturing by performing real-time analysis for a large volumes of data. One possible approach to performing these operations in a secure and privacy-preserving manner is to utilise cryptographic solutions. In previous work, we proposed SE with an AC algorithm for IIoT

based on an edge-cloud architecture, namely ELSA. This paper extends ELSA to illustrate the correlation between the number of keywords and ELSA performance. This extension supports annotating records with multiple keywords in trapdoor and record storage and allows the record to be returnable with single-keyword queries. In addition, the experiments demonstrate the scalability and efficiency of ELSA with an increasing number of keywords and complexity.

4.2 Introduction

Industry 4.0 is this century's revolution of the sector which started with the introduction of IoT, broadly refer to as IIoT [45, 4]. IIoT solutions are proposed for remote maintenance, quality control, product traceability, product life-cycle management and service optimization [14]. These aspects enforce a requirement for multi-actor access to the collected data such as insurers, customers, employees, and consultants. On the other hand, as with individuals, businesses also have privacy considerations often translating to competitive advantage or security. Thus, data could compromise the factory while controlled access to higher level information could be advantageous [7]. In this scenario data is often processed on the cloud. In this work we make the assumption that the cloud is not trusted [101].

Protecting data at rest has been a significant research domain in recent years utilising cryptographic primitives, AC policies [32], and SE [102]. The SE method however requires partial decryption on the cloud; it makes assumptions regarding the ordering of the data, and associates data with sets of keywords. To address these challenges, we recently published ELSA [120] (Figure 3.2); a keyword-based searchable encryption multi-authority (MA) access control (AC) for IIoT devices assisted by a three-tier edge computing architecture.

4.2.1 Contributions

We present an extension to ELSA to make the search queries more flexible allowing the retrieval of results that partially match the assigned keyword set provided by the user. Further, we present additional experiments investigating scalability. Our contributions are as follows:

- ELSA extension to support annotating records with multiple keywords in trapdoor and record storage and allowing the record to be returnable with single-keyword queries

- ELSA scalability experiments demonstrating search time remaining in the region of 10^2 ms for as many as 1000 keywords.

4.3 Related Work

SE allows the user to perform secure searches over encrypted data without compromising the data confidentiality. AC mechanisms are employed to dictate who has access to the data through access policies. A full review of both for the application domain of IIoT was previously presented in [86]. Since the early 2000's, SE [106] and asymmetrical SE [107] have been investigated as a method to allow retrieving only the required records of encrypted data. The method associates keywords to data records which can be retrieved through as single or multi-keyword search [108, 109].

With the rising use of the cloud the PEKS method has been combined with Attribute-Based Encryption (PEKS-ABE) [39]. However, protecting the privacy of user keys is an open challenge. The use of private cloud has also been investigated to improve data-privacy and performance in Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [74]. The limitation of CP-ABE methods is the use of a central authority for attribute authorisation.

Extensions of keyword-based searchable encryption work [110, 111] have eliminated the central authority without considering the computational overhead. Both approaches depend on bilinear pairing operations which are expensive in terms of memory requirements, making them unsuitable for IIoT applications.

For IIoT the complexity of the challenge increases as real-time execution and processing capabilities impose new requirements. SE and AC algorithms were not originally designed for embedded processors. Thus, the community has focused on improving efficiency of SE or AC in isolation. Lightweight SE with security improvements has been proposed in [55]. Attribute-based multi-keyword search schemes with CP-ABE were investigated in [76] to improve the accuracy of returned records to support dynamic IoT applications.

Most recently a low computational complexity SE-AC scheme was proposed for applications in IoT [44]. Their fine-grained AC, multi-keyword search, lightweight decryption, and LSABE-MA can support single keyword and multi-keyword searching while maintaining lightweight decryption. It also improves privacy preventing leakage in transit. The method meets the low latency requirements of IoT and supports improved security against chosen-keyword and the chosen-plaintext attacks. However, LSABE-MA does not guarantee real-time interactions, nor does it investigate the impact on returned record accuracy or data bandwidth. Accuracy, bandwidth, and real-time guarantees are considered important criteria for industrial applications [120, 86, 121].

LSABE-MA is based on sequential search of all the encrypted data records. Depending on the location of the record the search time can introduce latency larger than the required real-time guarantees. Hence, ELSA [120] expanded LSABE-MA to address this deficiency in the searching method and associated time as well as the bandwidth utilised. ELSA improved performance by an order of magnitude. This was achieved through suggesting an improved organisation of the data and an edge-cloud architecture. An edge server was proposed to cluster data indices by keyword leading to better than linear search performance while maintaining accuracy over the results.

Specifically for accuracy in LSABE-MA, when a record is annotated with a set of keywords the record can only be returned if a user requests records with the exact matching set of keywords. For example, a data record stored in the database annotated with keyword(k) set: k_1, k_2, k_3, will not be retrieved by a search query that requests records matching k_1. In plain words, if a user needs to see all the encrypted images of 'cats' we could assume that they would want to see those that simultaneously have 'cats and dogs' as well. Unfortunately, LSABE-MA will not return any images with both 'cats and dogs'. Thus, the existing searching approach limits the extent of the returned records and demands users to know the exact set of keywords which might be unsustainable for scalability. This is a deficiency in terms of the LSABE-MA scheme. ELSA on the other hand uses lookup tables on the edge server that could be used to identify unique keywords within a set associated with a record. This extension is presented in the following section.

4.4 Methods

The cloud-edge architecture of the ELSA method is presented in Figure 3.2. ELSA process queries over the encrypted data on the trusted edge to improve privacy. Partial decryption required for SE takes place on the trusted edge server. Also ELSA improves user key privacy protection by handling keys on the edge. The cloud server, not being trusted, is responsible for storage of the encrypted dataset (Figure 3.2 left). The added benefit of the edge server is the potential reduction of load on cloud communication bandwidth, while reducing the latency for query results. Thus, ELSA reduces overall core traffic. The edge (Figure 3.2 middle) is responsible for handling incoming requests from the user. Using a Bloom filter the edge can eliminate queries that would yield no result. Additionally, a query optimiser reduces the scope of the search for the cloud server. The edge is also responsible for establishing and enforcing the access policies directly linking to the IIoT devices and the data owner.

ELSA improves the search process by creating a lookup table in the edge to store the keywords with the unique number for each encrypted record before sending these records to the cloud. This lookup table process the search query (trapdoor) using a clustering algorithm to accelerate the search process.

The extended ELSA system uses this lookup table to support the multi-keywords scenario. It implements the multi keywords support so that system users can access the required data by issuing a query with any number of keywords. The key differences between multi-keyword architecture in the proposed system and single-keyword architecture are presented in Figure 4.1. The single-keyword architecture allows data users to find data corresponding with the identical sequences of keywords encrypted with the required data record. ELSA supports multi-keyword association as separate entries in the lookup table for each keyword against the same unique identifier of the record. This association can be in trapdoor and record storage, with the record being returnable. Supporting multi-keyword search in the system will avoid returning a reduced subset of results to the system user. Specifically, the lookup table in ELSA stores each keyword with a unique record number. Therefore, the multi-keyword trapdoor will be processed efficiently. The output of this will be all the records containing any of the requested keywords.

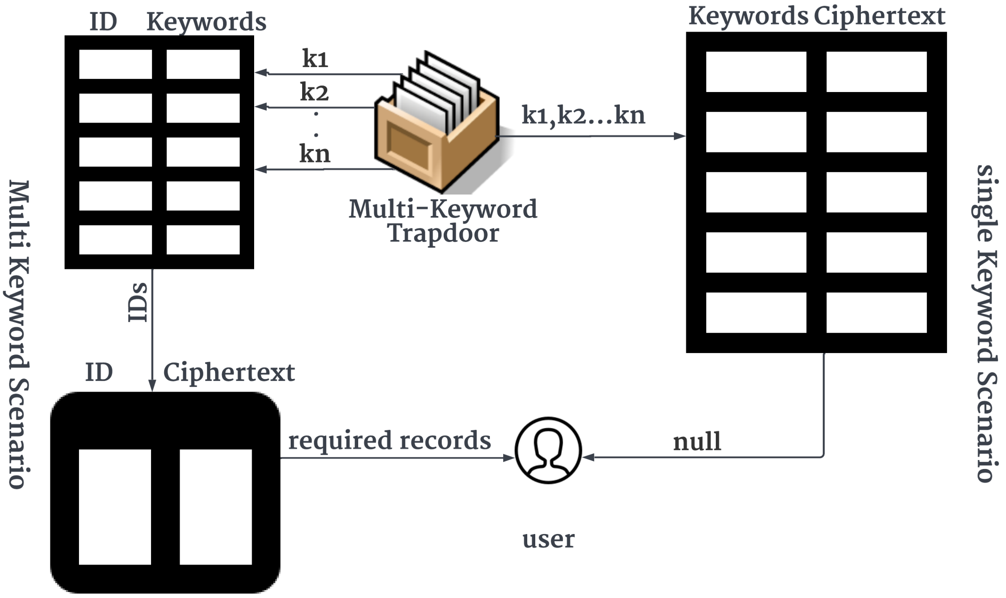

Figure 4.1: Single VS Multi-keyword Scenario.

The following experiments consider to the extended ELSA system based on the multi-keyword scenario from two aspects:

- Multi-keywords in lookup table.
- Multi-keywords in trapdoor (query).

4.5 Evaluation

This section outlines the experiments and results by discussing the influence of number of keywords in the lookup table and the trapdoor on the ELSA performance. Let $N_{_{KL}}$ be the number of keywords in the lookup table , $N_{_{KT}}$ be the number of keywords in the trapdoor.

We run the client application on an edge device with Intel 2.3 GHz Core i9 processor and 16GB RAM for evaluation setup. In addition, deployed the server code on a docker container hosted on a DigitalOcean cloud provider located in the UK. The plan for the cloud provider was CPU-Optimised, with one dedicated CPU, 2-32 vCPUs, 50 GB backing storage, 2GB RAM/CPU and 2TB Bandwidth.

For the experiment we used a synthetic dataset. The dataset consists of temperature, CO$_2$, and humidity values. For evaluation purposes, we considered the $\{CO2_{normal}$, $CO2_{high}$,$CO2_{low}$, $humidity_{normal}$, $humidity_{high}$, $humidity_{low}$, $temperature_{normal}$, $temperature_{high}$, $temperature_{low}\}$ keywords. We used two different cases of data based on the following criteria:

- percentage of representation of one of keywords (in this experiment is $CO2_{high}$),
- and number of keywords.

The two cases are (i) the sparse dataset case where the $CO2_{high}$ represent 5% of keywords, and (ii) the dense case where the $CO2_{high}$ represent 40%.

4.5.1 Multi-Keyword in Lookup Table

The specific trapdoor generated for this evaluation in each case was constant and contained one keyword, which is $CO2_{high}$. The lookup table generated by frequency of $N_{_{KL}}$ starts from 100 to 1000 keywords.

In Figure 4.2, we present the effect of the number of keywords in the lookup table $N_{_{KL}}$ on ELSA performance. The search time increases linearly as the $N_{_{KL}}$ increases. The extended ELSA system makes better use of the multi-keywords in the sparse case (5%) than dense case (40%). However, the computational cost of searching on the lookup table with 1000 keywords does not exceed 132 ms in the worst case, while the computational cost of the sparse case requires as little as 20 ms. As expected, the keyword representation ratio will directly affect the computational cost in this phase, but there is only a time gap of 132 to 20 ms between the above two cases, which could be acceptable but should be evaluated on a real-world dataset for validation.

Further to investigate scalability in terms of memory used, we measured the lookup table size for both the sparse case (5%) and dense case (40%) with different values of $N_{_{KL}}$. As presented in Figure 4.3, both cases remain below 14KB even for the very unrealistic scenario of 1000 keywords. Obviously, the lookup table size has an approximately linear relationship with $N_{_{KT}}$ values. However, it does not grow to an unsupported size for the edge server.

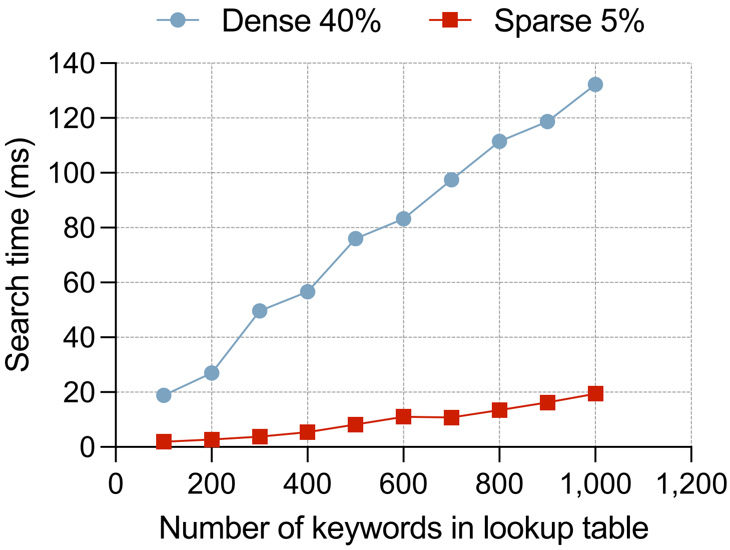

Figure 4.2: Search Time linearly increasing with no. keywords.

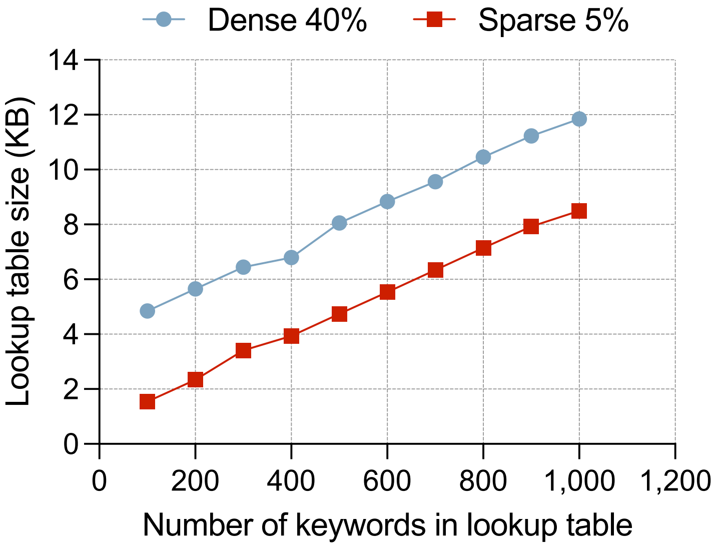

Figure 4.3: Lookup Table Size linearly increasing with no. keywords.

4.5.2 Multi-Keyword in Trapdoor

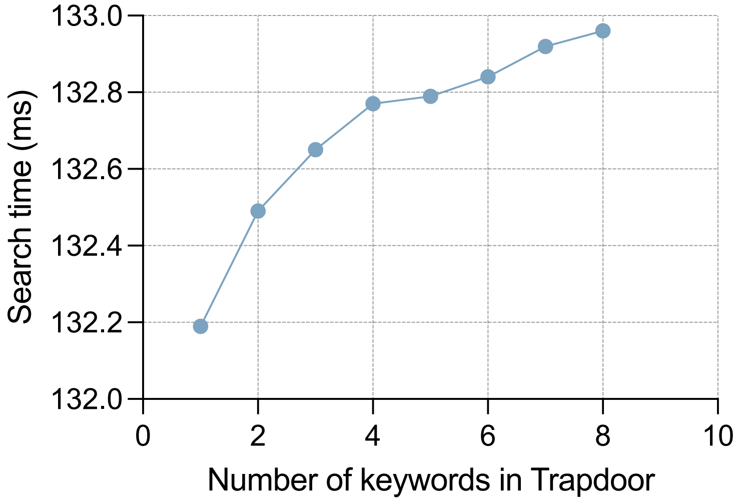

Figure 4.4: Search Time reaching a steady state below 3s regardless of no. keywords.

In this experiment, N_{KT} take a value from one keyword to eight keywords, and the lookup table contains 1000 keywords (worst case in previous experiment). Figure 4.4 presents the experimental result of search time under different values of N_{KT}. When the value of N_{KT} is 1 and 8, the computational cost of searching is 132.19 ms and 132.92 ms respectively. However, it is not linear and saturates below 133 ms, which is a promising result for scalability, and remains below 3 seconds which is marginally noticeable in terms of user experience.

4.6 Conclusion

First, we extend the ELSA system, which uses lookup tables on the edge server to identify unique keywords within sets of keywords. This extension can provide better accuracy of returned records with partial query searches. The experimental results for scalability with varying numbers of keywords demonstrate acceptable trade-off for query performance. The results show that ELSA's performance remains within acceptable limits from the perspective of user experience and memory utilisation when increasing the number of keywords. However, the lookup table size increases linearly. Therefore, in future work we will aim to minimise the lookup table size and summarise the data records by integrating ELSA with ML methods. This integration will eliminate records of unnecessary data that do not add value to further processing. The result would minimize all of the lookup table size, the cloud storage and the bandwidth utilisation taking full advantage of the edge architecture benefits.

Chapter 5

mini-ELSA: using Machine Learning to improve space efficiency in Edge Lightweight Searchable Attribute-based encryption for Industry 4.0

Aljabri, J., Michala, A. L., Singer, J. and Vourganas, I. (2022). mini-ELSA: using Machine Learning to improve space efficiency in Edge Lightweight Searchable Attribute-based encryption for Industry 4.0.

Aljabri contributed the literature review, implementation, results, discussion, and writing the first draft of the paper. Michala contributed to the conceptualisation, reviewed and edited drafts and the final version, and rewrote the methodology and contributions. Singer contributed to the experiment design and reviewed and edited the final version. Vourganas contributed to the conceptualisation, the methodology section, and reviewed the discussion and results.

In this chapter on using ML techniques for edge execution, we enhance ELSA by reducing the lookup table size and summarising the data entries. This integration eliminates redundant data records by analysing additional value for subsequent processing. Finally, this integration entirely employs the advantages of edge architecture that decrease network traffic, cloud storage, and search table size.

5.1 Abstract

Previous work has proposed a novel ELSA method to support Industry 4.0, and specifically Industrial Internet of Things applications. This paper aims to improve ELSA by minimising the lookup table size and summarising the data records by integrating ML methods suitable for execution at the edge. This integration eliminates records of *unnecessary* data by evaluating added value to further processing. This approach minimises the lookup table size, the cloud storage, and the network traffic by taking full advantage of the edge architecture benefits. We demonstrate our mini-ELSA expanded method using two well-known IIoT datasets. Our results indicated a reduction in storage requirements by $> 21\%$, an improvement of execution time by $> 1.39x$ and search time by $> 50\%$, while maintaining state-of-the-art predictive accuracy in both cases. We also present the computational complexity analysis supporting these experimental findings.

5.2 Introduction

As emerging research in the Industry 4.0 domain is progressing, IIoT applications have been proposed to improve the automation of factories and enable higher levels of control over the quality and quantity of production [14]. The industry has traditionally collected data to support maintenance and operations [121, 122, 123]. However, with the constant collection of more diverse data new opportunities arise. Multiple stakeholders can request access to data or analysis of the data to ensure regulatory compliance, improve insurance premiums, or examine the quality of ordered products.

Such use cases, however, demand that either the owner analyses the data and provides the requested answers, or the data are released to the requesting agent. Both approaches introduce barriers. In the first case, the owner must have the expertise to perform the analysis and must be trusted to provide the true results of the analysis [7]. In the second case, issues relating to security, GDPR regulation, and sensitive commercial information arise [21]. Continuous compliance has become a vivid new research domain as a result of these concerns. The continuous compliance solution for security compliance management is highly flexible and automated. This approach automates maintaining compliance with enterprise policies, regulations, and regulatory frameworks for managed IT services providers [124]. Previous works [120, 125] have proposed ELSA as a solution to these issues. This approach enables multiple stakeholders to run their own queries (and hence analysis) regarding the data while providing cyber-security and privacy preservation by design. In this approach, the process is automated, alleviating the burden of data analysis expertise and automatically complying with regulations, provided the owner has correctly described the access rules for each stakeholder. In this context, continuous compliance can be verified by the regulator at any point without the direct involvement of the owner, which improves trust in the results of the analysis.

However, ELSA requires the use of intermediate storage on an edge server. This intermediate storage is analogous to the recorded data introducing scaling constraints [125]. To address this limitation, in this paper we examine the combination of data preprocessing with our ELSA previously proposed method. Data preprocessing is well established in the domain of big data and can enable improved performance for data analytics, reducing storage needs while extracting the required meta-data for further processing [126].

Additionally, preprocessing can address issues relating to the quality of the collected data. This task is a fundamental challenge for IIoT applications in which actionable information must be extracted from high-quality data [127]. It is well known that data scientists invest 90% of a project's time in data preparation for ML or Artificial Intelligence (AI) [128]. This process is most often manual and provides results in non-real time [127].

We propose the integration of an ML pipeline at the edge server to automate and address the aforementioned challenges and lead to the following contributions:

- reduction of the memory requirements on the edge server by $> 21\%$ for two IIoT datasets representing different uses cases.
- reduction of the volume and improvement of the quality of permanently stored data by evaluating the contribution of each data-point to the ML model's learning.
- reduction of the search computational complexity by minimizing the size of encrypted records on the cloud and edge servers.

Through these contributions, our methodology achieves the following benefits over and above the state of the art: (i) to minimise the ELSA system storage requirements; (ii) to minimise cloud storage costs; (iii) to optimise network traffic over the full stack; and (iv) to maintain high-quality and optimal quantity datasets.

5.3 Related Work

Researchers need to propose new or optimised preprocessing techniques for historical large datasets and data streams to provide dynamic preprocessing of big streaming data, according to [126].

The data collected from different sources need to be processed for data quality, missing values, and outlier detection [129]. Accelerating the processing of data, which is unbounded, prevents any delay in further processing, as well as being important for making quick and intelligent decisions [130].

The transformation of raw data, collected through different devices, is a prerequisite for data processing. The raw data have valuable and useful information but contain a large amount of noise, duplicate values, missing values, and inconsistency, depending upon the architecture. Therefore, improving the raw data quality increases the efficiency and ease of data analysis. This process is also called data munging, which commonly includes removing the unnecessary or invalid data not required for identifying the underlying trends [131, 132].

The main task in data preprocessing is to eliminate noise and non-informative values and bring the reference parameters into a standard form. Therefore, training the model on raw experimental data produces unexpected results [133].

Surveys, such as [134], conducted by Dogan et al., have revealed that data selection is one of the main data-preparation problems in Industry 4.0. A widespread challenge of ML application in manufacturing is selecting the data relevant to the analysis from the available database. The manufacturing data obtained from machine measurements may not always be relevant for the data-mining process to solve the targeted problem. They can be related to different problems and be useless for a particular aspect. It is not apparent what part of the manufacturing dataset will be utilised at each point. This non-deterministic behaviour forces data miners to waste time in the non-beneficial data pile.

Factories leverage AI to transform information from various aspects of the manufacturing system into actionable insights. However, the data can contain a high degree of irrelevant and redundant information, and the relevant part may be missing altogether. These data-curation issues present a challenge for the application of ML algorithms, as the availability and quality of the manufacturing data strongly influence the performance and suitability of AI algorithms relative to the expected results. Therefore, ensuring local data quality is the key to enabling a causal analysis of the manufacturing system [135]. One approach with promising results in automatically identifying anomalies in data is random isolation forest [136].

Finally, in recent work [137], a new method was proposed to evaluate the contribution of each individual datapoint towards the output of an ML model. This approach is not ideal for cleaning the data of anomalies; however, it can be used to evaluate which datapoints are useful for further analysis and which datapoints might not be needed in the future as the knowledge they bring has already been observed by the trained model. We believe this idea, combined with anomaly detection, can automatically reduce the volume of data. This approach can automate the mining process, separating out unnecessary datapoints. We argue this approach will improve the scalability of the proposed ELSA method while improving automation for a variety of analytics performed on the collected data in Industry 4.0 applications that benefit from an edge architecture.

5.4 Methods

Our **mini-ELSA** method combines two techniques used widely in data preprocessing but often performed only on the training dataset manually and once at the beginning of the data pipeline. We aim to screen data as they come in from IIoT sensors to maintain high quality and minimise the quantity. For this purpose, we propose combining an automated anomaly detection method and a well-established data-valuation method in this new IIoT context. Our data pipeline methodology is presented in Figure 5.1.

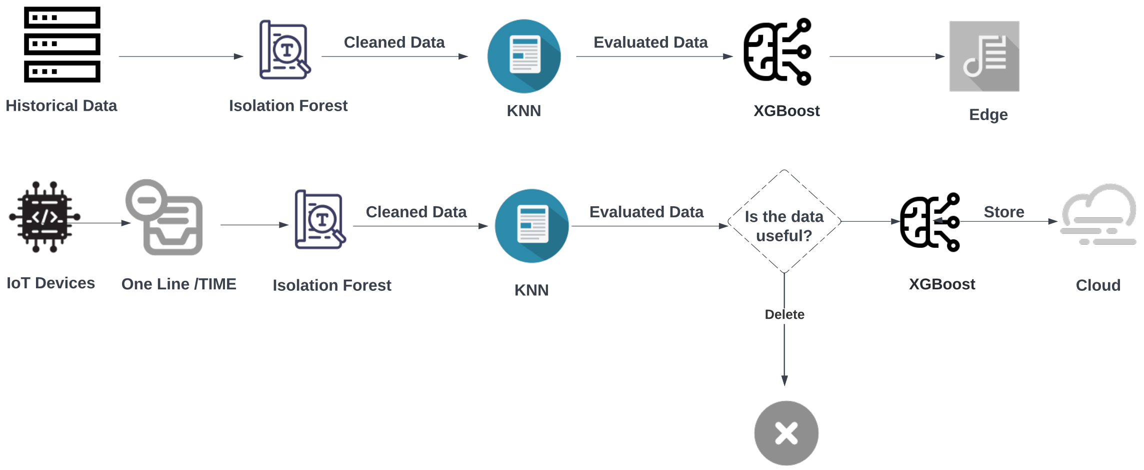

Figure 5.1: mini-ELSA.

We begin by dividing our dataset into a training set (90%) which in our IIoT context can be considered our *historical* dataset. This action is taken to maintain comparability with previous results reported in the literature, as presented in the following sections. At this stage, our data are processed through our anomaly detection module, which implements an isolation forest approach. We selected this approach because our dataset has been proven to provide better performance with random forests, according to [138]. For this paper, the authors evaluated five ML algorithms, namely, the k-nearest neighbours, linear regression, gradient-boosted regression tree (GBRT), artificial neural network, and deep neural network of the Rapid Miner software suite. Keeping the default parameters, the researchers evaluate the most crucial parameters of each algorithm to determine which best achieved the minimum root mean squared error (RMSE) and the mean absolute error (AE). The researchers also evaluate the effect of training set size and number of features on the achieved results. The researchers found GBRT outperformed the other algorithms by achieving the least RMSE and AE with 450 trees while training on 90% of the dataset. Interestingly, GBRT also exceeded in performance all the proposed methods on the same dataset by achieving the least RMSE and AE.

As a second step, our *historical* data are valuated. For each datapoint, a Shapley value is calculated as a measure of the datapoint's contribution to the ML model that our exemplar industrial setup requires. This method of datapoint valuation was proposed by [137]. This approach is generalised for any ML application but requires retraining when our mini-ELSA system is first deployed in a new setup or a new ML context. In this approach, the value of a datapoint is estimated based on its potential contribution to improving the model's predictive capacity. This approach uses the KNN method to implement efficient and performant Shapley value calculations.

Once the data are cleaned and valuated, they can be used to train any ML model. In this case, we utilise the proposed GBRT approach from [138] and implement it via the `xgboost` library provided in Python. This model aims to predict the output of the industrial setup as a regression problem (e.g. power generated in a power station, or the quality of a product produced in a plastics factory). However, any appropriate model can be used, depending on the use case.

Following the training phase, the trained models (Isolation Forest, KNN, XGBoost) are deployed on the edge server. Each new incoming datapoint arriving from any IIoT sensor can be screened in the same order for anomalies and valuation. If no anomaly and a high valuation are provided, then the datapoint can be stored on the ELSA system, populating the edge server lookup tables and, eventually, the cloud server encrypted database.

5.4.1 Implementation

We created an ML model pipeline and integrated it with the ELSA method presented in [120]. The ML pipeline was generated based on our methodology, as presented in Algorithm 3.

In this model, we extract the following features from Combined Cycle Power Plant (CCPP) dataset – temperature (T), ambient pressure (AP), relative humidity (RH), and exhaust vacuum (V) – and save them in the ”X” object. Then, we extracted the parameter we wanted to predict (PE) and saved it in the ”Y” object. After that, we split the data into the training and test subsets and and then scaled them to normalise them before running the isolation forest. Next, we created the isolation forest model with the number of isolation trees as 20, the number of samples as 50, and the contamination value as 0.1, and predicted the anomalies to remove, including outliers. Then, we ran the KNN regression and Shapley values calculation with testing and validation data, as presented in [137]. Finally, we created an XGBoost model named ‘reg’ with the best parameters (the learning rate was 0.0075, and the number of the tree was 9000) and fit it with the training dataset.

Algorithm 3 ML model

```
Load $df \leftarrow data$
Extract $X \leftarrow (AT, V, AP, RH)$
Extract $Y \leftarrow PE$
Split $X_{train}$ $\leftarrow$90% , $X_{test}$ $\leftarrow$10%
Split $Y_{train}$ $\leftarrow$90% , $Y_{test}$ $\leftarrow$10%
Scale $X_{train}$ , $X_{test}$
function RANDOM ISOLATION($Xtrain$ , $iforest$)
    $y_{pred} \leftarrow anomalies$
    Remove $y_{pred}$ from $X_{train}, Y_{train}$
    $X_{trainiforest} \leftarrow X_{train}$
    $Y_{trainiforest} \leftarrow Y_{train}$
end function
Create $x_{tst}, y_{tst}$ subsets
$(x_{tst}, y_{tst})$ $\leftarrow$first 450 lines of $X_{test}, y_{test}$
Create $(x_{val}, y_{val})$ subsets
$(x_{val}, y_{val})$ $\leftarrow$remaining of $X_{test}, y_{test}$
function GET TRUE KNN($X_{trainiforest}$, $x_{tst}$)
    $x_{tst\_knn\_gt} \leftarrow results$
end function
function GET TRUE KNN($X_{trainiforest}$, $x_{val}$)
    $x_{val\_knn\_gt} \leftarrow results$
end function
function UNWEIGHTED-KNNREGSHAPLEY($TrainingData$)
    $x_{valsp_{gt}} \leftarrow results$
    $X_{trainKNN} \leftarrow X_{trainiforest}$
    $Y_{trainKNN} \leftarrow Y_{trainiforest}$
end function
Create $idxs$object by sorting $g_{values}$
$keep_{idxs} \leftarrow idxs$
function XGB REGRESSION
    Remove Shapley value from Training Data
    if length $keep_{idxs} = X_{trainiforest}$ then
        $X_{trainkeep} \leftarrow X_{trainiforest}$
        $Y_{trainkeep} \leftarrow Y_{trainiforest}$
    else
        $keep_{idxs} \leftarrow X_{trainiforest}$
        $keep_{idxs} \leftarrow Y_{trainiforest}$
    end if
end function
Create xgboost name REG model
Train REG on $(x_{trainkeep}, Y_{trainkeep})$
Fit REG on the best $(X_{trainKNN}, Y_{trainKNN}$
Perform 5-fold cross-validation
Calculate MSE, RMSE, AE
```

5.4.2 Computational Complexity

5.4.2.1 ELSA

To calculate the computation complexity for ELSA, it is necessary to calculate the complexity for each main step as follows:

Let $|E|$, $|E_T|$, and P denote the bit-length of an element in groups G, G_T, and bilinear pairing, respectively. $|S|$ is the size of the attribute set S, E is exponentiations on Group G, $|U|$ is the size of the universe attribute set U, l_1 is the size of the keyword set KW, and l is the number of rows in the matrix of the access structure.

Key Generation $O(4|S||E|) = O(|S||E|)$ which is the same as LSABE-MA. Multi-authority needs to work together to generate the user's key, which takes time based on the size of the attributes.

Encryption is the same as LSABE-MA $O(3E_T{+}(2l{+}1)P{+}(2l{+}4)E) = O(E_T{+}l(P{+}E))$, in which the complexity is described as $2l + 4$ exponentiations on Group G. In this step, the latency is affected by the number of data to be encrypted and uploaded to the cloud.

Trapdoor $O(|S| + 1)E + P)) = O(|S|E + P)$, which is $|S| + 1$,requires exponentiations on Group G and no bilinear operations, which is the same as LSABE-MA. The user needs to encrypt the query keywords in this step. Thus, multiple keywords queries cause more computational overhead.

Range Generation the edge in our proposed method also contains the following methods:

1. K-mean: O(kn), in which k is the number of clusters, and n is the number of points
2. bloom filter: O(k_h), in which k_h is the number of hash functions.

Thus, the overall range-generation complexity is O($kn + k_h$).

Search the complexity will following the be LSABE-MA complexity but increased by the k-mean and bloom filter complexity, whereas the critical sizes are all reduced. In greater detail, assume x is the total number of encrypted records on the cloud, y is the total number of search records in the generated range by the edge server, and S_C is the search complexity of LSABE-MA (i.e., $S_C = O(3P + 3E_T + (2l + 1)E) = O(P + E_T + lE)$). Therefore, the complexity of LSABE-MA is $2l + 1$ exponentiations on Group G and three bilinear operations. The, the cloud search complexity for LSABE-MA is $x \cdot S_C$, whereas for ELSA it is $y \cdot S_C$. Therefore, the search complexity is reduced by $\frac{x}{y}\times$.

Decryption is the same as for LSABE-MA (i.e. $O(2E_T) = O(E_T)$). The process requires only two exponentiations on Group G_T. The latency in this step is based on the number of returned data returned after the search step.

5.4.2.2 Mini-ELSA

We added the cost of the mini-ELSA to the complexity of ELSA: $O(n \log n) + O(n \log n) = O(n \log n)$, in which n is the total number of training data points used for the ML pipeline before the encryption/search/decryption cycles, in which $O(n \log n)$ is the time complexity for the random forest and $O(n \log n)$ for the knn method.

The difference between this approach and LSABE-MA is that the sizes of x,y, and z will be smaller, so the worst-case complexity of mini-ELSA is always smaller than the LSABE-MA complexity.

The time complexity for the XGBoost of $O(td)$, in which t is the number of trees and d is the height of the trees, was further added to the ML pipeline for the predictive step. However, XGBoost was part of the use case, not part of the mini-ELSA method, so it was not added to the terms compared with LSABE-MA.

5.5 Evaluation Method

The goal of our methodology is to provide the following benefits over and above the state-of-the-art solution (i.e., ELSA): (i) to minimise the edge storage requirements; (ii) to minimise cloud storage costs and network traffic; (iii) to achieve a higher performance; and (iv)) do not sacrifice search or ML accuracy.

To evaluate the contributions presented in Section 5.2, we began by measuring the space requirements on the edge and in the cloud for mini-ELSA versus the original ELSA implementation presented in [120, 125] and baseline LSABE-MA. Furthermore, we derived the prediction accuracy of all three models to evaluate the effect of volume reduction towards predictive performance. We compared the predictive accuracy of the previously reported GBRT method in [138], the XGBoost approach without anomaly detection and data valuation, and our proposed XGBoost approach with the mini-ELSA implementation. We did not evaluate the security and privacy preservation benefits of ELSA but further presented execution times to evaluate our proposed mini-ELSA overheads over and above the original ELSA method.

5.5.1 Dataset and Use Case

We used the well known Combined Cycle Power Plant (CCPP) dataset from the UCI machine learning repository [139]. The dataset contains 9,568 data points collected over six years (2006-2011). The dataset's features consist of hourly average ambient variables of temperature (T), ambient pressure (AP), relative humidity (RH) and exhaust vacuum (V) to predict the net hourly electrical energy output (EP) of the plant (see Table 5.1).

Table 5.1: Feature descriptions of CCPP dataset sourced from [138]

Features	Min	Max	Variance	Std
Temperature	1.81°C	37.11	55.54	7.45
Ambient Pressure	992.89 milibar	1033.30	35.27	5.93
Relative Humidity	25.56%	100.16	213.17	14.6
Exhaust Vacuum	25.36 cm Hg	81.56	161.49	12.70
Power	420.26 MW	495.76	291.28	17.06

We also evaluated our proposed model by applying the Air Quality dataset [140]. We used this dataset, which consists of 14 attributes and 9,358 recorded instances in a year, to predict absolute humidity (AH). We extracted the following features in this model from the Air Quality (AQ) dataset (see Table 5.2):

Table 5.2: Feature descriptions of the AQ dataset

Features	unit
Date	MM/YYYY
Time	HH.MM.SS
Concentration CO	mg/m^3
PT08.S1 (tin oxide)	hourly averaged
Non-Metanic HydroCarbons concentration	$microg/m^3$
Benzene concentration	$microg/m^3$
Titania	hourly averaged
NO_x concentration in ppb	True hourly averaged
PT08.S3 (tungsten oxide)	hourly averaged
NO_2 concentration	$microg/m^3$
PT08.S4 (tungsten oxide)	hourly averaged
PT08.S5 (indium oxide)	hourly averaged
Temperature	$^\circ C$
Relative Humidity	$\%$

5.5.2 Experimental Setup

We ran the ML model and client application of ELSA on an edge device with an Intel 2.3 GHz Core i9 processor and 16GB RAM for the experimental setup. In addition, the server code was deployed on a docker container hosted on a DigitalOcean cloud provider in the UK. The plan for the cloud provider was CPU-Optimised, with one dedicated CPU, 2-32 vCPUs, 2TB bandwidth, 2GB RAM/CPU and 50 GB backing storage. The results were calculated by taking the average of 100 runs for the two systems, namely the original ELSA and mini-ELSA.

5.5.2.1 XGBoost Model

To further confirm the reliability, robustness, and validity of the developed model, the five-fold cross-validation method of 90% of randomly selected data was used to develop the model, and the remaining 10% of the data were used for model testing. Cross-validation is widely used for model selection, in which the training set is repeatedly split into training and validation sets, with each split of the training set is used for training and at least once for validation [141]. This process was repeated five times until each fold of the split data was tested once. The following Table 5.3 showed the average performance for all of the five folds.

Table 5.3: Five-cross validation average performance. The results reveal the variation between the RMSE performance and the AE.

MSE	RMSE	AE
12.38	3.51	2.47

For the ML model evaluation, we report three metrics appropriate for regression problems:

1. Mean squared error (MSE): This metric reports on the relationship between predictions and predicted values and can highlight significant prediction errors and punish models that do not predict well. A perfect MSE is 0.

2. Root Mean squared error (RMSE): This metric helps in presenting the real predicted value and not the squared. The perfect value is 0.

3. Mean Absolute error (AE): The metric can display a linear relationship between the AE and the predicted value.

In addition, we present the residuals plots, which can expose any bias that the model might have. The residual, by definition, is equal to: $Residual = Observed - Predicted$ value. Moreover, the residuals are sum to zero in a simple linear regression, and they have a mean of zero positive values for residuals (on the y-axis). The plot reveals whether the prediction was too low (negative) or too high (positive), with 0 meaning the prediction was correct.

5.5.2.2 mini-ELSA

It was necessary to categorise the data, as our system is a keyword-based search. The data were divided into equal sizes based on the dataset's minimum and maximum values of the EP. We classified the predicted EP in this dataset into four classifications based on the EP value: low, normal, high, and severe.

1. if the value of EP : between 420.26 and 439 then low
2. else if between 439 and 458 then normal
3. else if between 458 and 477 then high
4. else if between 477 and 495.76 then severe

We also classified the predicted AH for the AQ dataset into four classifications based on the value of AH: low, normal, high, and severe.

1. if the value of AH : between -200 and 0 then low
2. else if between 0.01 and 0.4 then normal
3. else if between 0.5 and 0.9 then high
4. else if between 1 and 2 then severe

For evaluation purposes, we considered the low, normal, high and severe values as the keywords.

To evaluate further the performance of our approach, we measured the search time using various keywords. We also evaluated the overall execution time for the original ELSA and the proposed mini-ELSA implementation. In addition, we measured the lookup table size at the edge.

5.6 Results

5.6.1 Edge Storage

The ELSA approach deploys a lookup table at the edge server to improve its search performance. The lookup table contains keywords that the query builder uses to construct optimal cloud search queries. To address Goal I, mini-ELSA minimisd the edge server storage by minimising the lookup table size (discussed in 5.4).

In Figure 5.2, we measured and compared the lookup table size for ELSA and mini-ELSA using the CCPP and AQ datasets. In ELSA, the number of keywords stored in the lookup table is 9,568 for CCPP and 9,358 for AQ, whereas in mini-ELSA it is 7,624 for CCPP and 7,117 for AQ. We also measured the storage size of the lookup table. The lookup table in ELSA for CCPP is 83.6 KB and for AQ is 55.7 KB, whereas in mini-ELSA it is 66.04 KB for CCPP and 41.22 KB for AQ . Therefore, mini-ELSA minimsed the edge storage for CCPP and AQ dataset by 21% and 26%, respectively. This result is the average of 100 repetitions, and it is a reliable result, as demonstrated by the error bars in Fig. 5.2.

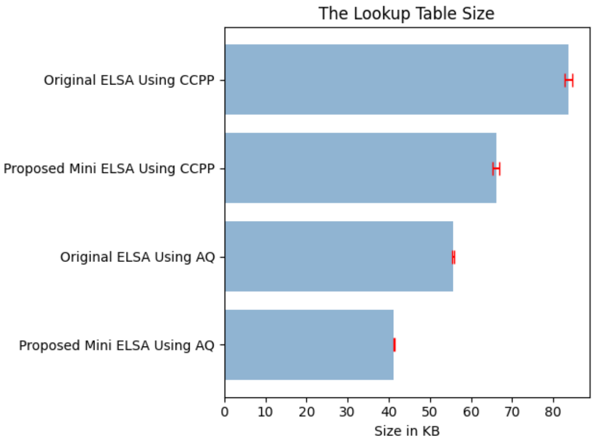

Figure 5.2: Lookup Table Size averaged over 100 repetitions using CCPP and AQ Dataset.

5.6.2 Cloud Storage

To address Goal II, we measured the ciphertext size on the cloud. For the original ELSA, it is 229.37 KB for CCPP and 198.8 KB for AQ, whereas for mini- ELSA it is 201.85 KB for CCPP and 171.28 KB for AQ. Therefore, mini-ELSA reduced the cloud storage by 12% for CCPP and 14% for AQ, again sustained over 100 repetitions.

5.6.3 Overall Execution

To address Goal III we examined the overall performance of the LSABE-MA baseline which is presented in [44], ELSA and mini-ELSA. Fig. 5.3 and Fig. 5.4 compares our mini-ELSA system with the original ELSA system and LSABE-MA regarding the overall execution time for the CCPP and AQ datasets. These graphs illustrate the overall effect of search time and lookup table size improvements on the latency experienced by the data user from initial query to final result. As demonstrated, the mini-ELSA system reduced the overall execution time for CCPP and AQ by 21.98% and 26.62 % for ELSA and 32.43% and 37.02%for LSABE-

MA, respectively. Our mini-ELSA improved execution time by $1.48\times$ for CCPP and $1.39\times$ for AQ.

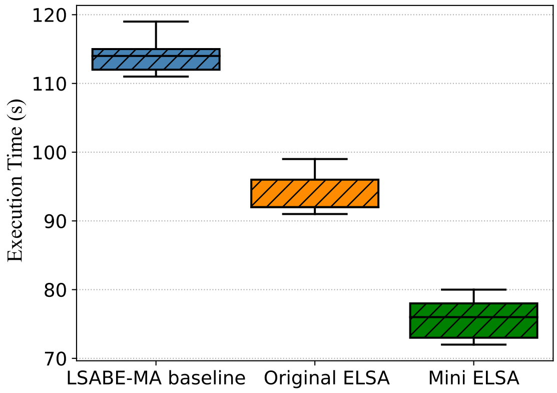

Figure 5.3: Overall Execution Time in Seconds measured over 100 repetitions for CCPP.

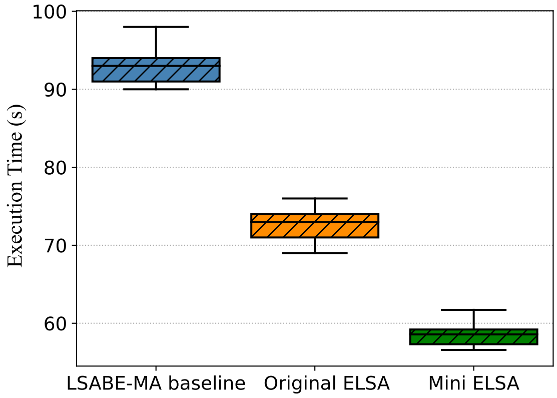

Figure 5.4: Overall Execution Time in Seconds measured over 100 repetitions for AQ.

5.6.4 Search Time

To further evaluate the search performance of the mini-ELSA approach further (as part of Goal IV), we compared the search time for mini-ELSA with the original ELSA approach and the LSABE-MA baseline presented in [44] for the CCPP and AQ dataset using the four attributes (Fig. 5.5, Fig. 5.6).Mini-ELSA performed better than the original ELSA and LSABE-MA approaches by 18.5% and 54.18% for CCPP and 21.27% and 51.59% for AQ, respectively, on average across all four cases.

5.6.5 Prediction Accuracy

To address Goal IV, regarding the predictive accuracy of the ML method, we examined the performance of the XGBoost model after the automated cleaning process using the anomaly

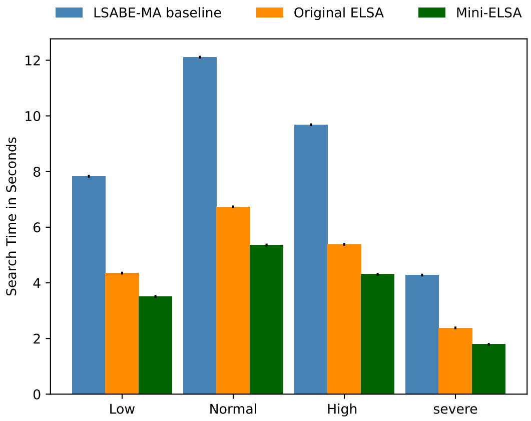

Figure 5.5: Search Time in Seconds measured over 100 for CCPP.

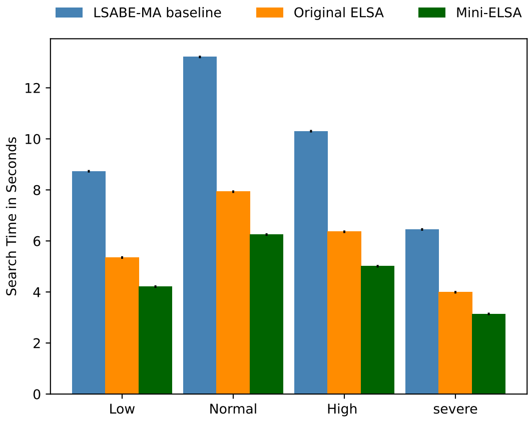

Figure 5.6: Search Time in Seconds measured over 100 for AQ.

detection and Shapley value calculation of mini-ELSA. We presented the observed vs predicted PE values in Fig. 5.7 for CCPP and Fig. 5.8 for AQ to understand the model's performance and linearity. The concentration across the diagonal demonstrates good linearity throughout the range of possible PE values supporting the suitability of the XGBoost regressor. The model has high accuracy, as there is a strong correlation between the actual observed PE values of the test subset and the predicted PE values of the model.

We also examined the performance of the isolation forest after removing the outliers to establish its ability to correctly remove datapoints correctly that deviate by more than three standard deviations from the mean of each feature assuming normal distribution. The accuracy of this algorithm reached to 95%, according to our experiment.

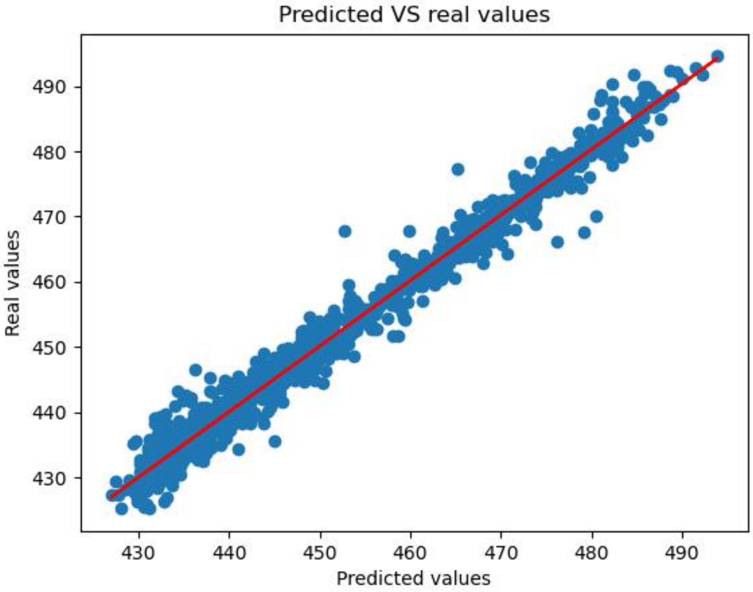

Figure 5.7: Predicted vs Real values of the XGBoost model used for the power plant decision support at the end of the mini-ELSA ML pipeline.

The residuals plot is presented in Fig. 5.9 for CCPP and Fig. 5.10 for AQ. The variance of the residual did not increase with the predicted values. Furthermore, there is no systematic curvature in the residual. Hence, there is a linear relationship, and the regression approach is suitable for our problem. As a result, we can assume that the error is normally distributed, homoscedastic, and independent. This outcome demonstrates the absence of bias in our approach, improving trust in the predicted output.

However, the residual plots demonstrate the presence of outliers, which are points higher than +10 or lower than -10 on Fig. 5.9,Fig. 5.10. Although they represent a very small minority, these outliers can influence the model's fit regarding edge cases.

To investigate further the presence of predictive outliers, we plotted a Q-Q residual plot (standardised vs theoretical quantiles) in Figure 5.11 and Figure 5.12. This graph illustrates

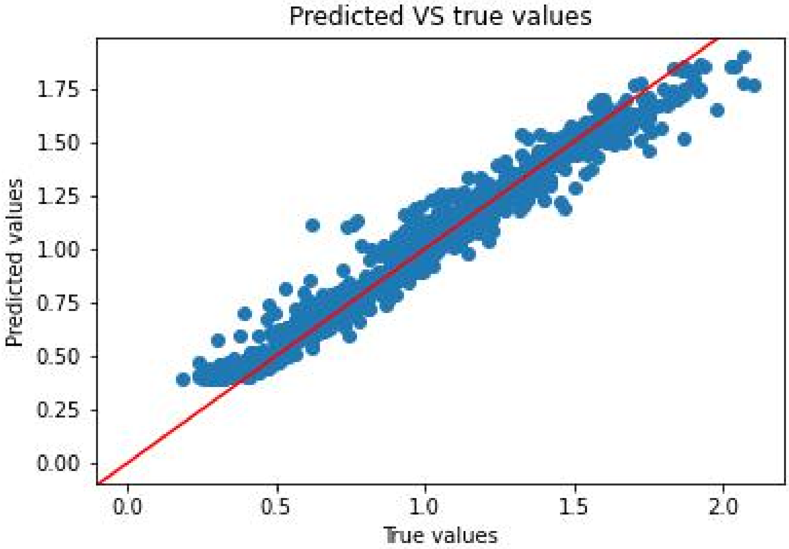

Figure 5.8: Predicted vs Real values of the XGBoost model used for the absloute humidity decision support at the end of the mini-ELSA ML pipeline.

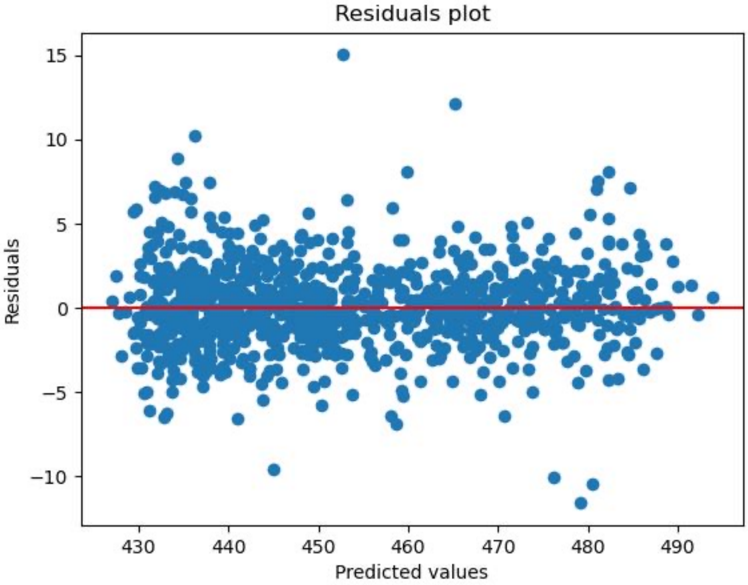

Figure 5.9: Residuals plot for XGBoost model with mini-ELSA for CCPP.

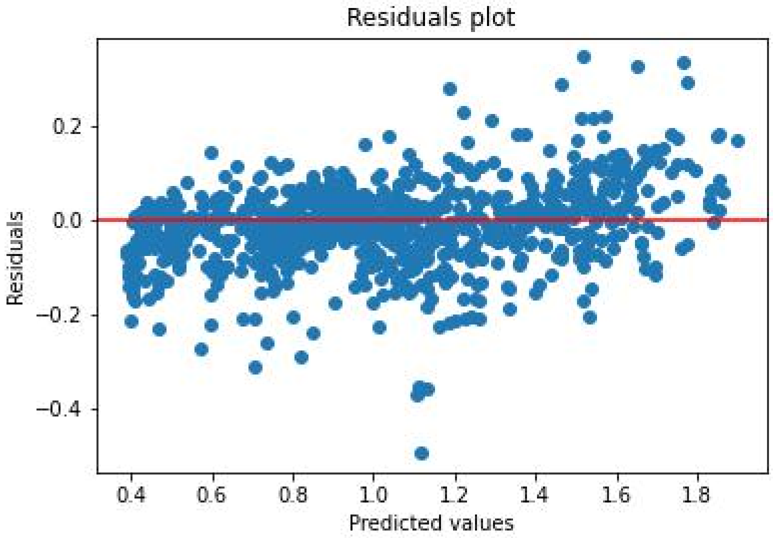

Figure 5.10: Residuals plot for XGBoost model with mini-ELSA for AQ.

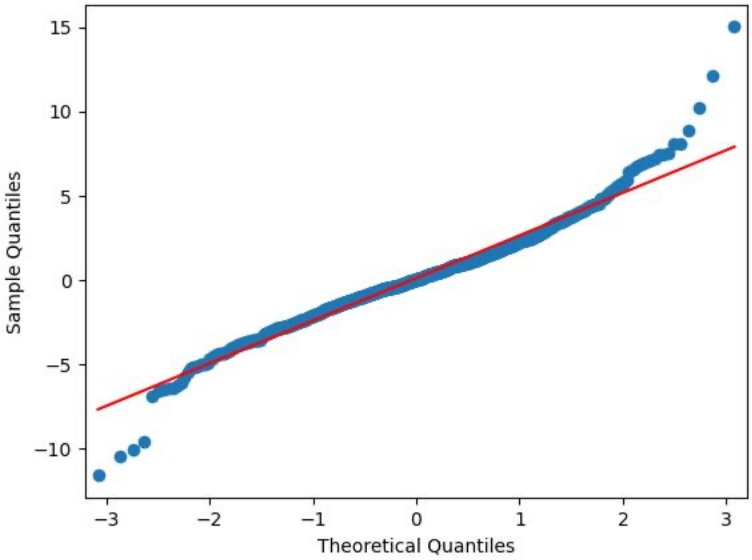

Figure 5.11: Q-Q plot of the XGBoost model with mini-ELSA for CCPP.

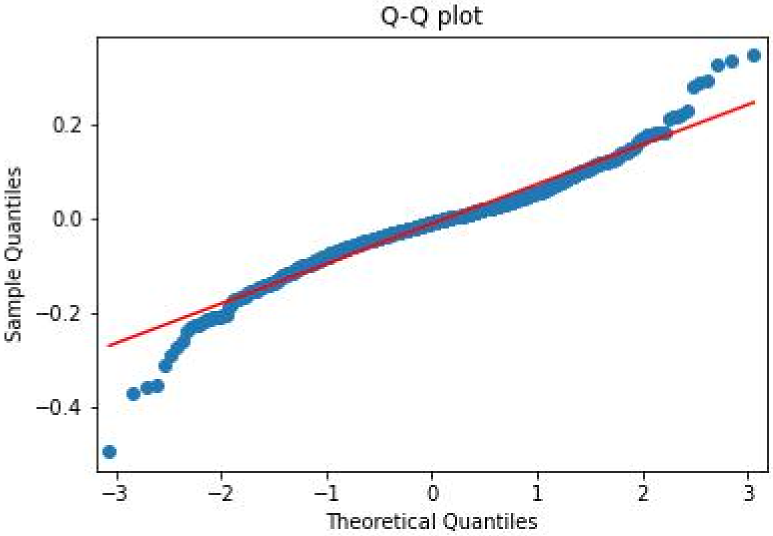

Figure 5.12: Q-Q plot of the XGBoost model with mini-ELSA for AQ.

the skew of the regression as an indication of a feature biasing the predicted performance. We observed the presence of the outliers at both ends of the Q-Q graph. However, the majority of the points reside on the diagonal, again demonstrating the error follows a normal and equal distribution, which again supports the absence of bias in the prediction.

Table 5.4: RMSE and AE.

	RMSE	AE
Baseline: Previous literature GBRT method in [138]	2.583	1.856
Previous literature Random Forest method in [142]	3.02	2.25
Original XGBoost approach	2.581	1.942
XGBoost with mini-ELSA	**2.533**	**1.851**

Table 5.5: MSE and R^2 comparison with previously reported approach for AQ dataset.

	MSE	R^2
Previous work BP [143]	3.6×10^{-7}	0.9999
mini-ELSA	0.00027	0.954

Finally, the effect of the most crucial parameters of the respective algorithms on the predicted power is presented in terms of both RMSE and AE. For the comparison, the dataset is split randomly into 90-10. We compared our XGBoost approach with the mini-ELSA implementation, with XGBoost approach without anomaly detection and data, and with Previous literature GBRT method in [138] and previous literature random forest method in [142] in

Table 5.4. The work in [142] used random forest and random tree models and an adaptive neuro-fuzzy inference system (ANFIS) for regression to predict the full load's electrical power output of the CCPP using the same dataset as in the experiments presented in Section 5.6. They applied these in two approaches: using all features and using a set of fewer features. Conducting their work using both approaches revealed the best results were achieved using random forest. For our evaluation, as we observe in the table, the automated dataset cleaning proposed in mini-ELSA improved the overall predictive performance of the most suitable regressor, as presented in [138].

Regarding the AQ dataset evaluation, we compared our work with the work represented in [143] as illustrated in Table 5.5. In their work [143], the researchers used a back-propagation (BP) learning algorithm, a radial basis function neural network (RBFNN), support vector regression (SVR), and a decision tree regressor (DTR). The best results for AQ dataset were achieved by the BP algorithm.

Furthermore, the correlation of determination (R^2) for the predicted versus real PE values of the mini-ELSA approach was 0.977 for the test CCPP dataset and 0.954 for AQ. This finding, combined with the low RMSE, supports the robustness and improved performance of mini-ELSA over and above the state-of-the-art regressor applied in this power plant dataset. Thus, mini-ELSA not only automates a manual process, but also further improves trust in the predictive performance of the power plant's decision-support system and predicted AH for the AQ system.

5.6.6 Trade-off Between Data Storage and Accuracy

There is a trade-off between data storage and accuracy. We have charactersed one point in this trade-off space. This point was the minimum data storage for which accuracy is comparable to the original algorithm. Therefore, we ran a mini-ELSA model with a different fraction of the whole data available after isolation forest and KNN, such as 10%, 20%, as presented in Figure 5.13. Moreover, we collected the difference from the best RMSE value for each case. We observed a decline in accuracy with larger removals of data; however, this might be an acceptable decline for some use cases.

5.7 Discussion

Our proposed mini-ELSA methodology can be retrained and parameterised to be suitable for any edge IIoT solution. Therefore, we need not change the framework or ML pipeline for mini-ELSA. However, to apply a new use case (new dataset), the parameters in the k-mean,

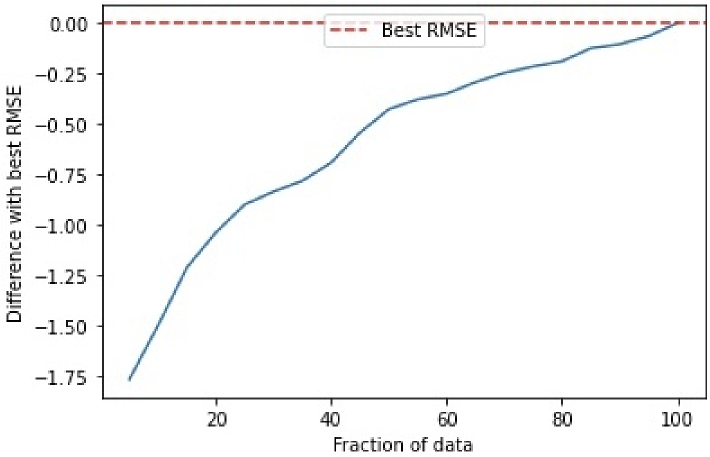

Figure 5.13: Size vs RMSE trade-off curve.

isolation foreset, KNN and XGboost in mini-ELSA need to be appropriately tuned to the best hyperparameters.

The CCPP dataset represents the sensor data from an industrial setting (it is related to the industrial use case because of our proposed mini-ELSA for industrial applications), and it is an excellent example and well-defined dataset employed in the literature [144, 145, 146].

Regarding the limitation of the isolation forest, removing more outliers increases RMSE in the final step. Furthermore, in this dataset, the mean of PE (predicted value) is $\tilde{4}00$. Therefore, a difference of 0.5 is minor considering the data; it corresponds to 0.13%. Additionally, the Isolation Forest needs to be hyper-parameterised for every new dataset.

5.8 Conclusion

We integrated an ML pipeline with the published ELSA method that supports a cloud-edge architecture for attribute-based searchable encryption with an optimised query process. We demonstrated our mini-ELSA expanded method using two well-known IIoT datasets. The results revealed that our mini-ELSA reduced the original ELSA method lookup table size at the edge by $> 21\%$, and consequently reduced the cloud storage by $> 12\%$. Furthermore, the mini-ELSA approach improved the execution time of ELSA by $> 1.39x$, providing performance benefits. Finally, the mini-ELSA pipeline reduced not only the volume of data, but also improved predictive performance by 1.11% for one of the datasets while automating the data-mining process. Additionally, the computational complexity analysis supporting these experimental findings was presented.

In future work, we aim to demonstrate the mini-ELSA methodology on a plastics factory use case to demonstrate transferability and evaluate the method's generalisation. We also plan to investigate the limitations of the proposed methodology, such as the hyperparameter tuning of the utilised models for each use case. Moreover, XGboost could also be configured to retrain incrementally for valuable new datapoints, based on their Shapley values.

Chapter 6

Conclusions & Future Work

6.1 Overview

This chapter outlines and concludes this thesis. In Section 6.2, a summary of the contributions of this thesis is presented, and in Section 6.3 the thesis statement is revisited. Section 6.4 presents several directions for future work derived from the limitations and the possible extensions to the current work. Finally, the concluding remarks are summarised in Section 6.5.

6.2 Contribution Summary

This thesis developed ELSA for IIoT. The developed system focused on adopting and improving the state-of-the-art security approaches to ensure the security and privacy of IIoT data. In addition, the developed system leverages the cloud-edge architecture to meet the IIoT requirements and reduce the execution time.

The contributions of this thesis are summarised as follows:

- **A review of searchable encryption with access control in IIoT**
  This thesis began by reviewing recent studies combining SE with AC in the IIoT context with an overview of IIoT and its application requirements. An extended literature review was conducted, with additional studies revealing the importance of IIoT and its applications. Finally, it presented previous work that integrated SE with AC and ML in the IIoT context to improve the system performance, highlighting the gap in the literature addressed in this thesis.

- **ELSA: a Keyword-based Searchable Encryption for Cloud-edge assisted Industrial Internet of Things**

This thesis proposed an edge lightweight searchable attribute- based encryption system (ELSA) to improve search time beyond the state-of-the-art solution. The ELSA approach leveraged the cloud-edge architecture and adopted and enhanced a keyword-based searchable encryption with a multi-authority (MA) access control (AC) scheme (Section 3.5). The ELSA system was implemented and evaluated against the state-of-art searchable encryption scheme, as presented in Section 3.4. The evaluation results indicated that ELSA reduced the search time by up to $14\times$ and overall performance by $1.21\times$ 3.7. The ELSA system achieved this speedup by utilising the edge server to offload and optimise computations from the cloud. This process provides better-than-linear search performance with a trade-off in utilised storage space. This contribution addresses RQ1, RQ2 and RQ4.

- **ELSA: Edge Lightweight Searchable Attribute-based encryption Multi-keyword Scalability**
  To improve the performance of ELSA further, this thesis proposed a system named extended ELSA 4.4 to illustrate the correlation between the number of keywords and ELSA performance. Briefly, the extended ELSA supports multiple keywords in trapdoor and record storage. In addition, it allows records retrieval with single-keyword queries. The experimental results in Section 4.5 revealed that the findings for scalability with a various numbers of keywords displayed a reasonable trade-off for query performance. Furthermore, the findings demonstrated that ELSA's performance remains within reasonable bounds regarding user experience and memory use even when the number of keywords increases. This contribution addresses RQ3.

- **mini-ELSA: using Machine Learning to improve space efficiency in Edge Lightweight Searchable Attribute-based encryption for Industry 4.0**
  This thesis adopted and integrated an ML pipeline into ELSA to reduce the cloud and edge storage and improve the system performance. The enhanced ELSA, named mini-ELSA, was presented in Section 5.4. mini-ELSA eliminates unnecessary data records by evaluating the data using an ML pipeline before further processing. This mechanism enables mini-ELSA to reduce the lookup table size and enhance the system's performance. We illustrated our mini-ELSA expanded method using two well-known IIoT datasets. The evaluation results revealed that our mini-ELSA reduced the original ELSA method lookup table size at the edge by $> 21\%$, and consequently reduced the cloud storage by $> 12\%$, as discussed in Section 5.6. Furthermore, the mini-ELSA approach enhanced the execution time of ELSA by $> 1.39x$ and search time by $> 50\%$, providing performance benefits. Finally, the mini-ELSA pipeline reduced the data volume and improved predictive performance by 1.11 % while automating the data-mining process. Moreover, the computational complexity analysis helped to support

these experimental findings. This contribution addresses RQ5.

6.3 Thesis Statement Revisited

In this section, the thesis statement is repeated from Section 1.2, and the remainder of this section indicates how it has been addressed. The thesis statement is restated as follows:

> Industrial Internet of Things (IIoT) systems, such as monitoring systems, collect and analyse sensitive IIoT data (e.g., sensor data) to detect unusual events and provide critical insights. These systems must meet the real-time requirements of IIoT while coping with the low computational power and storage of IIoT devices without compromising the security and privacy of the data. This thesis proposes a system to efficiently and securely collect IIoT data while allowing authorised users and monitoring systems to securely search and analyse the IIoT data based on their privileges.
> Thus, we hypothesise that by combining cloud-based Public Key Encryption with Keyword Search (PEKS) and edge-based Ciphertext-Policy Attribute Based Encryption (CP-ABE) through an edge computing architecture using statistical techniques to reduce data transfer, we can provide lower latency, lower network traffic, higher security, and privacy, without sacrificing accuracy with respect to identified state of the art encryption approaches in IIoT applications as baseline for comparison.

This thesis began with a description of the need for a system for IIoT applications to support intelligent factory needs. This work proposed ELSA to improve the performance beyond the state-of-the-art solution. Based on the evaluation, ELSA significantly reduced the search time by up to $14\times$ (i.e. provided low latency). Additionally, the evaluation results indicated that ELSA preserved higher security and privacy without sacrificing accuracy. To further improvements, we extended ELSA to support multiple keywords in trapdoor and record storage. The experiments revealed that performance was maintained even when the number of keywords increased, but the data volume increased as well. Furthermore, the ML pipeline was developed to extend ELSA as mini-ELSA to reduce data transfer. The results indicated mini-ELSA improved space efficiency. Finally, the limitations of the current research are now outlined.

6.4 Limitations and Future Directions of Research

This section discusses the limitations of the thesis and provides possible directions for future research.

The Source Code for LSABE-MA.

Due to the lack of access to the source code for the LSABE-MA framework, we created a clean room version of the scheme, which enabled us to maintain the same features but with minor changes. Therefore, we followed the LSABE-MA paper's description and comprehensively evaluated the scheme. The evaluation was conducted on a single deployment scenario. We believe this approach provided a sufficient representation of the results to enable us to hypothesise about the emerging trends.

Dataset.

We conducted several experiments on a dataset that included sensor data. Therefore, a more significant number of factors would validate the investigated approaches. Future work should consider applying the mini-ELSA methodology to a plastics factory use case to demonstrate transferability and evaluate the method's generalisation.

ML Pipeline.

We presented an ML pipeline with ELSA. We combined automated anomaly detection and well-established data-valuation methods in this new IIoT context in this pipeline. Future work needs to investigate the limitations of this proposed pipeline methodology, such as the hyperparameter tuning of the utilised models for each use case. For example, xgboost could also be configured to retrain valuable new data points incrementally based on their Shapley values. Moreover, the isolation forest must be hyperparameterised for each new dataset. For example, removing the forest in the AQ dataset case improved the R2 results to 0.98. This change adds complexity to the deployment, but it should not affect the operational performance of the pipeline. However, this issue is a limitation of every ML approach and is not particular to mini-ELSA. Furthermore, this sustains our finding of a trade-off between accuracy and storage size.

ML Methods.

We integrated ML methods to reduce data transfer and storage on the cloud. Further work needs to investigate proper techniques to reduce cloud and edge data storage, such as data summarisation or compression. In this way, the data aggregated can be further summarised by performing simple operations, such as average, maximum, minimum, and total, removing any inherent redundancies before sending it to the cloud.

IIoT Data Sensors Regulations.

As IIoT data sensor technology advances, regulatory compliance with regulations such as the General Data Protection Regulation (GDPR) of the European Union acts becomes increasingly critical [147]. Therefore, the development of data governance tools is required. These tools need to check and evaluate the legal collection, use, and sharing of IIoT data. Although the work in this thesis paves the way for continuous compliance, there remains work to be done to ensure adherence to all the clauses of the regulatory framework.

Untrusted Edge Server.

This thesis considers a fully trusted edge server and a semi-trusted cloud server. As the cloud server is usually in an outsourced cloud environment, it is more vulnerable to external and internal attackers, as seen in many incidents [148, 149, 150]. On the other hand, the edge server can be an on-premise server within the same control domain, owned and secured by an enterprise; thus, we consider it fully trusted. However, some studies consider an untrusted edge server [151]. To further strengthen the security of this thesis, we consider possible solutions to securely deploy ELSA on an untrusted edge server.

One possible approach is to extend ELSA at the edge server by encrypting the lookup table and the bloom filter. This approach can be easily implemented but has several drawbacks. First, it will increase the index table size as ELSA's ciphertext size is larger than its plaintext. The second drawback is that it will degrade the performance on the edge server as cryptographic protocols (i.e., searchable encryption) incur additional computational overhead compared to plaintext processing (searching). Such drawbacks do not meet with the research questions of this thesis, in specific **RQ3** and **RQ4** (See Section 1.3), which aim to design an efficient framework suitable for IIoT.

Another solution is to adopt hardware-based techniques, i.e., a trusted execution environment (TEE). For example, Intel's SGX [152] offers a secure execution environment for applications by dividing them into trusted and untrusted parts. The trusted part isolates the sensitive code and data in a protected memory called the enclave. The untrusted part can only communicate with the trusted part through specific interfaces, while other processes on the same CPU, including privileged ones, cannot access or tamper with the EPC.

There are several options to adopt SGX in ELSA. The first is to entirely deploy ELSA inside the enclave, including code and data (i.e., the lookup table and the bloom filter). In this option, we assume that the index and the sketch size will not exceed the limited enclave size, i.e., 90MB in SGX v1 [152]. Note that we assume the adoption of SGX v1 since it is widely adopted in commodity servers, such as edge servers. Nonetheless, if the data size exceeds the enclave size, the enclave will need to perform page swapping, an expensive operation that can add up to $10\times$ overhead [153]. Another option is to store the sketch table inside the enclave (as it requires a few kilobytes of memory) and to encrypt and store the lookup table outside the enclave. However, this approach requires fetching the lookup table inside the enclave in batches (that are not larger than the enclave size) for processing. Implementing and evaluating the performance of previous design options and exploring new ones are valuable directions for future work.

6.5 Concluding Remarks

Industrial manufacturing is moving toward a digital transformation and the use of technology to improve the efficiency and performance of its manufacturing operations. In industrial manufacturing, digitisation includes back-office, supply chain applications, factory automation, data analytics, and more, all driven by smart systems and connected devices. This digitisation can be achieved through IIoT, also known as Industry 4. The IIoT uses large quantities of data from smart sensors and machines to improve manufacturing and industrial processes through real-time analytics. Inspired by other industries and best practices, the IIoT leverages cloud platforms to deliver real-time analytics to make business decisions more quickly and accurately.

However, such data enrichment in the cloud makes the IIoT data vulnerable to internal and external attacks, as evident in many incidents. In addition, attacks on IIoT systems can place the manufacturing process at severe risk, such as tampering with production or providing inaccurate results that may affect the product's quality.

One approach to protecting IIoT data in the cloud is to adopt cryptographic protocols to process IIoT data securely while preserving privacy. Despite the benefit of cryptographic protocols, they impose a high computational overhead that can degrade the performance of an IIoT application, which affects the real-time requirements of such IIoT systems.

To overcome these limitations and fulfil the IIoT system requirements, this thesis presented an ELSA system that leverages the cloud-edge architecture to answer IIoT system queries securely while implying access controls. The ELSA approach can improve search time beyond the state-of-the-art solution (i.e. LSABE-MA). Instead of linearly searching encrypted records in the cloud, ELSA introduces a query optimiser on the trusted edge server to optimise client search queries and reduces the total number of searches for records.

The edge server adopts a clustering algorithm (i.e., k-mean) to cluster the records' by their indices. Clusters form a range of indices, which are further optimised to reduce the query size, and thus minimise network bandwidth. Therefore, ELSA substitutes a linear search in the cloud for a sublinear search, depending on the data distribution.

Compared with the previous state-of-the-art solution (i.e., LSABE-MA), the experimental results indicated that ELSA reduced the search time by up to $14\times$ and the overall performance by $1.21\times$.The ELSA system also significantly reduced execution time and communication overhead by clustering the required data without sacrificing accuracy. Furthermore, the performance of ELSA remained within acceptable user experience and memory consumption when the number of keywords increased. However, the size of the lookup table increased linearly as the number of keywords increased.

To overcome the previous drawbacks, we adopted an ML pipeline with ELSA, named mini-

ELSA. mini-ELSA adopts ML models to preprocess the data prior to insertion, eliminating noise and outliers and reducing cloud data. We demonstrated the mini-ELSA expanded method using two well-known IIoT datasets. The evaluation results revealed that mini-ELSA reduced the size of ELSA's lookup table on the edge by $> 21\%$ and the cloud storage by $> 12\%$. In addition, mini-ELSA improved the execution time of ELSA by $> 1.39x$. In addition to reducing data size, mini-ELSA improved the model's predictive performance by 1.11 $\%$ while automating the data-mining process. Finally, the experiment findings are supported by computational complexity analysis.

Bibliography

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE transactions on industrial informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[2] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," *Computers in industry*, vol. 101, pp. 1–12, 2018.

[3] M. Younan, E. H. Houssein, M. Elhoseny, and A. A. Ali, "Challenges and recommended technologies for the industrial internet of things: A comprehensive review," *Measurement*, vol. 151, p. 107198, 2020.

[4] P. Mathur, *IoT Machine Learning Applications in Telecom, Energy, and Agriculture*, 2020.

[5] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Computers in Industry*, vol. 101, no. April, pp. 1–12, 2018. [Online]. Available: https://doi.org/10.1016/j.compind.2018.04.015

[6] G. Drosatos, K. Rantos, D. Karampatzakis, T. Lagkas, and P. Sarigiannidis, "Privacy-preserving solutions in the Industrial Internet of Things," pp. 219–226, 2020.

[7] S. T. H. E. Web and F. O. R. A.-r. Devices, "IoT FOR BUSINESS Take Manufacturing ' s Shift Your Manufacturing Shift to Lightspeed to Lightspeed," 2020.

[8] W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial internet of things: Recent advances, enabling technologies and open challenges," *Computers and Electrical Engineering*, vol. 81, p. 106522, 2020. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2019.106522

[9] Accenture, "Winning with the Industrial Internet of Things," p. 12, 2018. [Online]. Available: https://www.accenture.com/t20160909T042713Z__w__/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Dualpub_11/Accenture-Industrial-Internet-of-Things-Positioning-Paper-Report-2015.pdfla=en

[10] H. P. Breivold and K. Sandstrom, "Internet of Things for Industrial Automation-Challenges and Technical Solutions," *Proceedings - 2015 IEEE International Conference on Data Science and Data Intensive Systems; 8th IEEE International Conference Cyber, Physical and Social Computing; 11th IEEE International Conference on Green Computing and Communications and 8th IEEE Inte*, pp. 532–539, 2015.

[11] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, "Wireless sensor networks for healthcare," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, 2010.

[12] T. P. Raptis, A. Passarella, and M. Conti, "Data management in industry 4.0: State of the art and open challenges," *IEEE Access*, vol. 7, pp. 97 052–97 093, 2019.

[13] M. Khan, X. Wu, X. Xu, and W. Dou, "Big data challenges and opportunities in the hype of industry 4.0," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[14] Y. Yu, R. Chen, H. Li, Y. Li, and A. Tian, "Toward data security in edge intelligent iiot," *IEEE Network*, vol. 33, no. 5, pp. 20–26, 2019.

[15] G. Drosatos, K. Rantos, D. Karampatzakis, T. Lagkas, and P. Sarigiannidis, "Privacy-preserving solutions in the industrial internet of things," in *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2020, pp. 219–226.

[16] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[17] T. Gebremichael, L. P. Ledwaba, M. H. Eldefrawy, G. P. Hancke, N. Pereira, M. Gidlund, and J. Akerberg, "Security and privacy in the industrial internet of things: Current standards and future challenges," *IEEE Access*, vol. 8, pp. 152 351–152 366, 2020.

[18] P. K. Malik, R. Sharma, R. Singh, A. Gehlot, S. C. Satapathy, W. S. Alnumay, D. Pelusi, U. Ghosh, and J. Nayak, "Industrial internet of things and its applications in industry 4.0: State of the art," *Computer Communications*, vol. 166, pp. 125–139, 2021.

[19] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial internet of things (iiot)–enabled framework for health monitoring," *Computer Networks*, vol. 101, pp. 192–202, 2016.

[20] L. Luo, Y. Zhang, B. Pearson, Z. Ling, H. Yu, and X. Fu, “On the security and data integrity of low-cost sensor networks for air quality monitoring,” *Sensors (Switzerland)*, vol. 18, no. 12, pp. 1–22, 2018.

[21] D. Preuveneers, W. Joosen, and E. Ilie-Zudor, “Data protection compliance regulations and implications for smart factories of the future,” in *2016 12th International Conference on Intelligent Environments (IE)*. IEEE, 2016, pp. 40–47.

[22] J. Khan, “The need for continuous compliance,” *Network Security*, vol. 2018, no. 6, pp. 14–15, 2018.

[23] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and Privacy Challenges in Industrial Internet of Things,” 2015. [Online]. Available: http://dx.doi.org/10.1145/2744769.2747942.

[24] M. S. Hossain and G. Muhammad, “Cloud-assisted Industrial Internet of Things (IIoT) - Enabled framework for health monitoring,” *Computer Networks*, vol. 101, pp. 192–202, 2016.

[25] P. P. Jayaraman, X. Yang, A. Yavari, D. Georgakopoulos, and X. Yi, “Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation,” *Future Generation Computer Systems*, vol. 76, pp. 540–549, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.03.001

[26] H. F. Atlam, A. Alenezi, R. K. Hussein, and G. B. Wills, “Computer Network and Information Security,” *Computer Network and Information Security*, vol. 1, pp. 26–35, 2018. [Online]. Available: http://www.mecs-press.org/

[27] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, “Challenges and opportunities in securing the industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 2985–2996, 2020.

[28] X. Yu and H. Guo, “A survey on IIoT security,” *Proceedings - 2019 IEEE VTS Asia Pacific Wireless Communications Symposium, APWCS 2019*, pp. 1–5, 2019.

[29] W. Z. Khan, M. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, “Industrial internet of things: Recent advances, enabling technologies and open challenges,” *Computers & Electrical Engineering*, vol. 81, p. 106522, 2020.

[30] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” *IEEE INTERNET OF THINGS JOURNAL*, vol. 4, no. 5, p. 1125, 2017. [Online]. Available: http://www.ieee.org/publications_standards/publications/rights/index.html

[31] T. Gebremichael, L. P. Ledwaba, M. H. Eldefrawy, G. P. Hancke, N. Pereira, M. Gidlund, and J. Akerberg, “Security and Privacy in the Industrial Internet of Things: Current Standards and Future Challenges,” *IEEE Access*, vol. 8, pp. 152 351–152 366, 2020.

[32] K. Pothong, I. Brass, M. Carr, L. Tanczer, S. Security, R. Nicolescu, B. Craggs, E. Lupu, A. Rashid, C. Maple, S. Wakenshaw, M. Taddeo, J. Lindley, S. Cannizzaro, R. Procter, and P. Coulton, “Editors of the Cybersecurity of the Internet of Things: PETRAS Stream Report 03 Privacy and Trust 05 Adoption and Acceptability,” 2019.

[33] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, “Edge Computing in IoT-Based Manufacturing,” *IEEE Communications Magazine*, vol. 56, no. 9, pp. 103–109, 2018.

[34] M. Wazid, A. K. Das, R. Hussain, G. Succi, and J. J. P. C. Rodrigues, “Accepted Manuscript Authentication in cloud-driven IoT-based big data environment: survey and outlook Authentication in cloud-driven IoT-based big data environment: survey and outlook,” *Journal of Systems Architecture*, 2018. [Online]. Available: https://doi.org/10.1016/j.sysarc.2018.12.005

[35] X. Kong, J. Chang, M. Niu, X. Huang, J. Wang, and S. I. Chang, “Research on real time feature extraction method for complex manufacturing big data,” *International Journal of Advanced Manufacturing Technology*, vol. 99, no. 5-8, pp. 1101–1108, 2018.

[36] L. Zhou, K.-H. Yeh, G. Hancke, Z. Liu, and C. Su, “Security and privacy for the industrial internet of things: An overview of approaches to safeguarding endpoints,” *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 76–87, 2018.

[37] K.-K. R. Choo, S. Gritzalis, and J. H. Park, “Cryptographic solutions for industrial internet-of-things: Research challenges and opportunities,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, 2018.

[38] K. Chamili, M. J. Nordin, W. Ismail, and A. Radman, “Searchable Encryption: A Review,” *International Journal of Security and Its Applications*, vol. 11, no. 12, pp. 79–88, 2017.

[39] Y. Zhou, N. Li, Y. Tian, D. An, and L. Wang, “Public key encryption with keyword search in cloud: A survey,” *Entropy*, vol. 22, no. 4, pp. 1–24, 2020.

[40] S. T. Hsu, C. C. Yang, and M. S. Hwang, “A study of public key encryption with keyword search,” *International Journal of Network Security*, vol. 15, no. 2, pp. 71–79, 2013.

[41] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in *Australasian conference on information security and privacy*. Springer, 2015, pp. 59–76.

[42] S. Shekhar and H. Xiong, "Geo-Role-Based Access Control," *Encyclopedia of GIS*, pp. 368–368, 2008.

[43] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.

[44] K. Zhang, J. Long, X. Wang, H.-N. Dai, K. Liang, and M. Imran, "Lightweight Searchable Encryption Protocol for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 3203, no. c, pp. 1–1, 2020.

[45] M. Hermann, T. . Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios: A Literature Review," Tech. Rep. [Online]. Available: www.snom.mb.tu-dortmund.de

[46] P. R. Newswire, "Global Industrial IoT Market: Research report 2015-2019," *Lon-Reportbuyer*, 2015.

[47] Freddie Roberts, "9 examples of manufacturers making IIoT work for them," 2016.

[48] C. Liu, F. Chen, J. Zhu, Z. Zhang, C. Zhang, and C. Zhao, *Industrial IoT Technologies and Applications*, 2017, vol. 202. [Online]. Available: http://link.springer.com/10.1007/978-3-319-60753-5

[49] J. Wan, J. Li, M. Imran, and D. Li, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3652–3660, 2019.

[50] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

[51] P. Jayalaxmi, R. Saha, G. Kumar, N. Kumar, and T.-h. Kim, "A Taxonomy of Security Issues in Industrial Internet-of-Things: Scoping Review for Existing Solutions, Future Implications, and Research Challenges," *IEEE Access*, vol. 9, pp. 1–1, 2021.

[52] M. S. Virat, S. M. Bindu, B. Aishwarya, B. N. Dhanush, and M. R. Kounte, "Security and Privacy Challenges in Internet of Things," *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018*, pp. 454–460, 2018.

[53] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges," *IEEE Communications Surveys & Tutorials*, no. c, pp. 1–1, 2020.

[54] I. Ungurean and N. C. Gaitan, "A software architecture for the industrial internet of things—a conceptual model," *Sensors (Switzerland)*, vol. 20, no. 19, pp. 1–19, 2020.

[55] B. Chen, L. Wu, N. Kumar, K.-K. R. Choo, and D. He, "Lightweight Searchable Public-key Encryption with Forward Privacy over IIoT Outsourced Data," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. c, pp. 1–1, 2019.

[56] S. Yu, C. Wang, K. Ren, and W. Lou, *Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5462174/

[57] R. Charanya, M. A. . I. C. on, and u. 2016, "Survey on access control issues in cloud computing," *ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7603014/

[58] M. U. Aftab, Z. Qin, N. W. Hundera, O. Ariyo, Zakria, N. T. Son, and T. Van Dinh, "Permission-based separation of duty in dynamic role-based access control model," *Symmetry*, vol. 11, no. 5, 2019.

[59] S. Bhatt, L. A. Tawalbeh, P. Chhetri, and P. Bhatt, "Authorizations in cloud-based internet of things: Current trends and use cases," *2019 4th International Conference on Fog and Mobile Edge Computing, FMEC 2019*, vol. 1, pp. 241–246, 2019.

[60] K. Rajesh Rao, I. G. Ray, W. Asif, A. Nayak, and M. Rajarajan, "R-PEKS: RBAC Enabled PEKS for Secure Access of Cloud Data," *IEEE Access*, vol. 7, pp. 133 274–133 289, 2019.

[61] P. J. Sun, "Privacy Protection and Data Security in Cloud Computing: A Survey, Challenges, and Solutions," *IEEE Access*, vol. 7, pp. 147 420–147 452, 2019.

[62] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward Efficient Multi-Keyword Fuzzy Search over Encrypted Outsourced Data with Accuracy Improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[63] M. Rasori, "fABElous : An Attribute-Based Scheme for," 2019.

[64] D. Sathya and P. G. Kumar, "Secured remote health monitoring system," *Healthcare Technology Letters*, vol. 4, no. 6, pp. 1–5, 2017.

[65] Y. Miao, X. Liu, K. K. R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, "Privacy-Preserving Attribute-Based Keyword Search in Shared Multi-owner Setting," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–15, 2019.

[66] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable Attribute-based Keyword Search over Outsourced Encrypted Data," Tech. Rep.

[67] H. Yin, J. Zhang, Y. Xiong, L. Ou, F. Li, S. Liao, and K. Li, "CP-ABSE: A Ciphertext-Policy Attribute-Based Searchable Encryption Scheme," *IEEE Access*, vol. 7, pp. 5682–5694, 2019.

[68] Q. Li, Y. Yue, and Z. Wang, "Deep Robust Cramer Shoup Delay Optimized Fully Homomorphic For IIOT secured transmission in cloud computing," *Computer Communications*, vol. 161, pp. 10–18, 2020. [Online]. Available: https://doi.org/10.1016/j.comcom.2020.06.017

[69] L. Guo, Z. Li, W. C. Yau, and S. Y. Tan, "A Decryptable Attribute-Based Keyword Search Scheme on eHealth Cloud in Internet of Things Platforms," *IEEE Access*, vol. 8, pp. 26 107–26 118, 2020.

[70] Y. W. Hwang, I. Y. Lee, and K. Yim, "A Study on Access Control Scheme Based on ABE Using Searchable Encryption in Cloud Environment," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 47, pp. 215–221, 2020.

[71] D. Ziegler, A. Marsalek, B. Prünster, and J. Sabongui, "Efficient Access-Control in the IIoT through Attribute-Based Encryption with Outsourced Decryption," *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications: SECRYPT*, 2020.

[72] N. Löken, "Searchable encryption with access control," *ACM International Conference Proceeding Series*, vol. Part F1305, no. Sfb 901, 2017.

[73] P. Chaudhari and M. L. Das, "Privacy Preserving Searchable Encryption with Fine-grained Access Control," *IEEE Transactions on Cloud Computing*, vol. 7161, no. c, pp. 1–1, 2019.

[74] S. Qi, Y. Lu, W. Wei, and X. Chen, "Efficient Data Access Control with Fine-Grained Data Protection in Cloud-Assisted IIoT," *IEEE Internet of Things Journal*, vol. 4662, no. c, pp. 1–1, 2020.

[75] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2018.

[76] Y. Miao, X. Liu, R. H. Deng, H. Wu, H. Li, J. Li, and D. Wu, "Hybrid keyword-field search with efficient key management for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3206–3217, 2019.

[77] M. Javaid, A. Haleem, R. P. Singh, S. Rab, and R. Suman, "Upgrading the manufacturing sector via applications of industrial internet of things (iiot)," *Sensors International*, vol. 2, p. 100129, 2021.

[78] P. Moens, V. Bracke, C. Soete, S. Vanden Hautte, D. Nieves Avendano, T. Ooijevaar, S. Devos, B. Volckaert, and S. Van Hoecke, "Scalable fleet monitoring and visualization for smart machine maintenance and industrial iot applications," *Sensors*, vol. 20, no. 15, p. 4308, 2020.

[79] A. Vaclavova, P. Strelec, T. Horak, M. Kebisek, P. Tanuska, and L. Huraj, "Proposal for an iiot device solution according to industry 4.0 concept," *Sensors*, vol. 22, no. 1, p. 325, 2022.

[80] B. Singh, "Industrial internet of things: Applications and challenges," *Futuristic Research Trends and Applications of Internet of Things*, pp. 229–248, 2022.

[81] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, "Security issues in iiot: A comprehensive survey of attacks on iiot and its countermeasures," in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN).* IEEE, 2018, pp. 124–130.

[82] B. Jiang, J. Li, G. Yue, and H. Song, "Differential privacy for industrial internet of things: Opportunities, applications, and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 430–10 451, 2021.

[83] P. Zhang, Y. Wu, and H. Zhu, "Open ecosystem for future industrial internet of things (iiot): architecture and application," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 1, pp. 1–11, 2020.

[84] S. Pal and Z. Jadidi, "Protocol-based and hybrid access control for the iot: Approaches and research opportunities," *Sensors*, vol. 21, no. 20, p. 6832, 2021.

[85] N. Andola, R. Gahlot, V. K. Yadav, S. Venkatesan, and S. Verma, "Searchable encryption on the cloud: a survey," *The Journal of Supercomputing*, vol. 78, no. 7, pp. 9952–9984, 2022.

[86] J. Bader and A. L. Michala, "Searchable encryption with access control in industrial internet of things (iiot)," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[87] H. Yin, Y. Li, F. Li, H. Deng, W. Zhang, and K. Li, "An efficient and access policy-hiding keyword search and data sharing scheme in cloud-assisted iot," *Journal of Systems Architecture*, p. 102533, 2022.

[88] D. He, M. Ma, S. Zeadally, N. Kumar, and K. Liang, "Certificateless public key authenticated encryption with keyword search for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3618–3627, 2017.

[89] S. VENKATESAN, K. R. RAJESHWARI, and M. RAMAKRISHNAN, "A lightweight searchable encryption and delegation mechanism with forward privacy for improving the security of industrial internet of things-cloud systems," 2022.

[90] P. Ambika, "Machine learning and deep learning algorithms on the industrial internet of things (iiot)," *Advances in computers*, vol. 117, no. 1, pp. 321–338, 2020.

[91] S. Mukkamala, A. H. Sung, A. Abraham, and V. Ramos, "Intrusion detection systems using adaptive regression spines," in *Enterprise information systems VI.* Springer, 2006, pp. 211–218.

[92] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "Iot security techniques based on machine learning: How do iot devices use ai to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[93] A. O. Prokofiev, Y. S. Smirnova, and V. A. Surov, "A method to detect internet of things botnets," in *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).* IEEE, 2018, pp. 105–108.

[94] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[95] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on iot security: Challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet of Things*, vol. 11, p. 100227, 2020.

[96] B. Yang, X. Cao, X. Li, Q. Zhang, and L. Qian, "Mobile-edge-computing-based hierarchical machine learning tasks distribution for iiot," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2169–2180, 2019.

[97] F. Raheem and N. Iqbal, "and machine learning for the industrial internet of things (iiot)," *Industrial Internet of Things: Technologies and Research Directions*, p. 1, 2022.

[98] P. Bellavista, R. Della Penna, L. Foschini, and D. Scotece, "Machine learning for predictive diagnostics at the edge: An iiot practical example," in *ICC 2020-2020 IEEE International Conference on Communications (ICC).* IEEE, 2020, pp. 1–7.

[99] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Noise removal in the presence of significant anomalies for industrial iot sensor data in manufacturing," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7084–7096, 2020.

[100] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2017.11.022

[101] J. Blömer and N. Löken, "Cloud architectures for searchable encryption," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018, pp. 1–10.

[102] G. S. Poh, J.-J. Chin, W.-C. Yau, K.-K. R. Choo, and M. S. Mohamad, "Searchable symmetric encryption: designs and challenges," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–37, 2017.

[103] D. W. Chadwick, W. Fan, G. Costantino, R. De Lemos, F. Di Cerbo, I. Herwono, M. Manea, P. Mori, A. Sajjad, and X.-S. Wang, "A cloud-edge based data security architecture for sharing and analysing cyber threat information," *Future Generation Computer Systems*, vol. 102, pp. 710–722, 2020.

[104] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[105] L. Zhou, M. H. Samavatian, A. Bacha, S. Majumdar, and R. Teodorescu, "Adaptive parallel execution of deep neural networks on heterogeneous edge devices," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 195–208.

[106] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE symposium on security and privacy. S&P 2000.* IEEE, 2000, pp. 44–55.

[107] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *International conference on the theory and applications of cryptographic techniques.* Springer, 2004, pp. 506–522.

[108] R. Chen, Y. Mu, G. Yang, F. Guo, X. Huang, X. Wang, and Y. Wang, "Server-aided public key encryption with keyword search," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2833–2842, 2016.

[109] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, “Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data,” *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 312–325, 2015.

[110] Y. Miao, R. H. Deng, X. Liu, K.-K. R. Choo, H. Wu, and H. Li, “Multi-authority attribute-based keyword search over encrypted cloud data,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1667–1680, 2019.

[111] Q. Huang, G. Yan, and Y. Yang, “Privacy-preserving traceable attribute-based keyword search in multi-authority medical cloud,” *IEEE Transactions on Cloud Computing*, 2021.

[112] M. A. Jarwar, S. A. Khowaja, K. Dev, M. Adhikari, and S. Hakak, “NEAT: A resilient deep representational learning for fault detection using acoustic signals in IIoT environment,” *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[113] L. D. Xu and L. Duan, “Big data for cyber physical systems in industry 4.0: a survey,” *Enterprise Information Systems*, vol. 13, no. 2, pp. 148–169, 2019.

[114] B. H. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[115] G. Cormode, “Sketch techniques for approximate query processing,” *Foundations and Trends in Databases. NOW publishers*, 2011.

[116] L. Liu and M. T. Özsu, *Encyclopedia of database systems.* Springer New York, NY, USA:, 2009, vol. 6.

[117] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications.* SIAM, 2020.

[118] L. Rokach, “A survey of clustering algorithms,” in *Data mining and knowledge discovery handbook.* Springer, 2009, pp. 269–298.

[119] M. Buckland and F. Gey, “The relationship between recall and precision,” *Journal of the American society for information science*, vol. 45, no. 1, pp. 12–19, 1994.

[120] J. Aljabri, A. L. Michala, and J. Singer, “ELSA: a keyword-based searchable encryption for cloud-edge assisted industrial internet of things,” in *22nd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2022),Taormina (Messina), Italy, 16-19 May 2022.* IEEE/ACM, 2022.

[121] A. L. Michala, I. Vourganas, and A. Coraddu, “Vibration edge computing in maritime iot,” *ACM Transactions on Internet of Things*, vol. 3, no. 1, pp. 1–18, 2021.

[122] A. L. Michala, I. Lazakis, G. Theotokatos, and T. Varelas, “Wireless condition monitoring for ship applications,” *Smart Ship Technology*, pp. 59–66, 2016.

[123] I. Lazakis, K. Dikis, A. L. Michala, and G. Theotokatos, “Advanced ship systems condition monitoring for enhanced inspection, maintenance and decision making in ship operations,” *Transportation Research Procedia*, vol. 14, pp. 1679–1688, 2016.

[124] R. Filepp, C. Adam, M. Hernandez, M. Vukovic, N. Anerousis, and G. Q. Zhang, “Continuous compliance: Experiences, challenges, and opportunities,” in *2018 IEEE World Congress on Services (SERVICES)*. IEEE, 2018, pp. 31–32.

[125] J. Aljabri, A. L. Michala, and J. Singer, “Elsa: Edge lightweight searchable attribute-based encryption multi-keyword scalability,” 2022.

[126] C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, S. U. Khan *et al.*, “Big data reduction methods: a survey,” *Data Science and Engineering*, vol. 1, no. 4, pp. 265–284, 2016.

[127] Y. Cui, S. Kara, and K. C. Chan, “Manufacturing big data ecosystem: A systematic literature review,” *Robotics and computer-integrated Manufacturing*, vol. 62, p. 101861, 2020.

[128] M. Stonebraker, I. F. Ilyas *et al.*, “Data integration: The current status and the way forward.” *IEEE Data Eng. Bull.*, vol. 41, no. 2, pp. 3–9, 2018.

[129] J. P. Verma, S. Tanwar, S. Garg, I. Gandhi, and N. H. Bachani, “Evaluation of pattern based customized approach for stock market trend prediction with big data and machine learning techniques,” *International Journal of Business Analytics (IJBAN)*, vol. 6, no. 3, pp. 1–15, 2019.

[130] T. Vyas, S. Desai, and A. Ruparelia, “Fog data processing and analytics for health care-based iot applications,” in *Fog Data Analytics for IoT Applications*. Springer, 2020, pp. 445–469.

[131] D. Cemernek, S. Cemernek, H. Gursch, A. Pandeshwar, T. Leitner, M. Berger, G. Klösch, and R. Kern, “Machine learning in continuous casting of steel: A state-of-the-art survey,” *Journal of Intelligent Manufacturing*, pp. 1–19, 2021.

[132] B. Mons *et al.*, “Invest 5% of research funds in ensuring data are reusable,” *Nature*, vol. 578, no. 7796, pp. 491–491, 2020.

[133] N. Staroverova, M. Shustrova, and M. Fatykhov, "Machine learning in the development of regulators: the specifics of data preparation and the choice of the ml model algorithm," in *IOP Conference Series: Materials Science and Engineering*, vol. 1155, no. 1. IOP Publishing, 2021, p. 012050.

[134] A. Dogan and D. Birant, "Machine learning and data mining in manufacturing," *Expert Systems with Applications*, vol. 166, p. 114060, 2021.

[135] C. Wang, X. Tan, S. Tor, and C. Lim, "Machine learning in additive manufacturing: State-of-the-art and perspectives," *Additive Manufacturing*, vol. 36, p. 101538, 2020.

[136] L. Bandić, M. Hasičić, and J. Kevrić, "Prediction of power output for combined cycle power plant using random decision tree algorithms and anfis," in *International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies*. Springer, 2019, pp. 406–416.

[137] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gürel, B. Li, C. Zhang, C. J. Spanos, and D. Song, "Efficient task-specific data valuation for nearest neighbor algorithms," *CoRR*, vol. abs/1908.08619, 2019. [Online]. Available: http://arxiv.org/abs/1908.08619

[138] R. Siddiqui, H. Anwar, F. Ullah, R. Ullah, M. A. Rehman, N. Jan, and F. Zaman, "Power prediction of combined cycle power plant (ccpp) using machine learning algorithm-based paradigm," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[139] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 126–140, 2014.

[140] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.

[141] D. R. Roberts, V. Bahn, S. Ciuti, M. S. Boyce, J. Elith, G. Guillera-Arroita, S. Hauenstein, J. J. Lahoz-Monfort, B. Schröder, W. Thuiller *et al.*, "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," *Ecography*, vol. 40, no. 8, pp. 913–929, 2017.

[142] S. Avdaković, A. Mujčić, A. Mujezinović, T. Uzunović, and I. Volić, *Advanced Technologies, Systems, and Applications IV-Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT 2019)*. Springer, 2019, vol. 83.

[143] Y. Kirsal Ever, K. Dimililer, and B. Sekeroglu, “Comparison of machine learning techniques for prediction problems,” in *Workshops of the international conference on advanced information networking and applications*. Springer, 2019, pp. 713–723.

[144] B. Akdemir, “Prediction of hourly generated electric power using artificial neural network for combined cycle power plant,” *Int. J. Electr. Energy*, vol. 4, no. 2, pp. 91–95, 2016.

[145] T. Ghosh, K. Martinsen, and P. K. Dan, “Data-driven beetle antennae search algorithm for electrical power modeling of a combined cycle power plant,” in *World Congress on Global Optimization*. Springer, 2019, pp. 906–915.

[146] M. G. R. Shuvo, N. Sultana, L. Motin, and M. R. Islam, “Prediction of hourly total energy in combined cycle power plant using machine learning techniques,” in *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*. IEEE, 2021, pp. 170–175.

[147] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, “A systematic survey of industrial internet of things security: Requirements and fog computing opportunities,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

[148] C. Quintin, “HealthCare.gov Sends Personal Data to Dozens of Tracking Websites,” https://www.eff.org/deeplinks/2015/01/healthcare.gov-sends-personal-data[online], 2015.

[149] M. Netten, “Capital One Experiences Third Largest Financial Hack from AWS Insider,” https://securityboulevard.com/2019/08/capital-one-experiences-third-largest-financial-hack-from-aws-insider/, 2019.

[150] I. is Beautiful, “World’s Biggest Data Breaches & Hacks,” https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/[online], 2020.

[151] H. Park, S. Zhai, L. Lu, and F. X. Lin, “StreamBox-TZ: Secure Stream Analytics at the Edge with TrustZone,” in *USENIX ATC*, 2019.

[152] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, “Innovative instructions and software model for isolated execution.” *Hasp@ isca*, vol. 10, no. 1, 2013.

[153] M. Taassori, A. Shafiee, and R. Balasubramonian, “VAULT: Reducing Paging Overheads in SGX with Efficient Integrity Verification Structures,” in *ACM ASPLOS*, 2018.