Universitatea
Transilvania
din Brașov

HABILITATION THESIS

Title: Perspectives Concerning Significant Conceptual and Practical Computer Science and Interdisciplinary Aspects

Domain: Computer Science

Author: Assoc. Prof. Dr. Răzvan Bocu

University: Transilvania University of Brașov, România

BRAŞOV, 2024

CONTENTS

A. Rezumat

Activitatea științifică desfășurată după obținerea titlului de doctor în domeniul Informatică are legătură cu variate domenii de cercetare științifică. Astfel, câteva dintre cele mai semnificative domenii de cercetare științifică avute în vedere sunt: ingineria sistemelor software, soluții avansate pentru administrarea sigură a rețelelor de calculatoare, aplicații software pentru rețele de calculatoare, proiecte de cercetare interdisciplinară în domeniul bioinformaticii, învățare automată și inteligență artificială, tehnici avansate privind securitatea și confidențialitatea datelor personale, specificarea și implementarea eficientă a unor modele de criptare și semnătură digitală, inclusiv modelarea *post-quantum* a acestora, aplicații și arhitecturi instalate în infrastructuri și rețele de date de generația următoare, soluții integrate privind confidențialitatea datelor personale bazate pe tehnici de criptare homomorfică (*homomorphic encryption*), soluții software avansate pentru vehicule autonome, soluții algoritmice și software pentru procesarea și analizarea bazelor de date mari folosind tehnici de învățare automată. De asemenea, contribuții științifice relevante au legătură cu proiectarea, implementarea și optimizarea structurilor de rețea virtualizate, precum și cu studiul rețelelor complexe, inclusiv cele biologice.

Este important să menționăm că lista de referință a lucrărilor științifice publicate este disponibilă la următoarea adresă web: https://www.razvanbocu.bocu.ro/?page_id=18 .

Întrucât activitatea de cercetare științifică este reflectată într-un număr mare de articole științifice publicate, în continuare, vor fi enumerate doar zece dintre cele mai semnificative articole publicate în jurnale Web of Science/Clarivate, precum și în volumele conferințelor indexate în CORE. O prezentare detaliată a activității de cercetare științifică este realizată în corpul principal al acestei teze de abilitare.

- R. Bocu, M. Iavich. **Enhanced detection of low-rate DDoS attack patterns using machine learning models**. *Journal of Network and Computer Applications*, volume 227, 103903. https://doi.org/10.1016/j.jnca.2024.103903, 2024.
- R. Bocu. **Dynamic Monitoring of Time-Dependent Evolution of Biomolecules Using Quantum Dots-Based Biosensors Assemblies**. *Biosensors* **2024**, *14*, 380. https://doi.org/10.3390/bios14080380, 2024.
- R. Bocu, A. Baicoianu, A. Kerestely. **An Extended Survey Concerning the Significance of Artificial Intelligence and Machine Learning Techniques for Bug Triage and Management**. IEEE Access, vol. 11, pp. 123924-123937, doi: 10.1109/ACCESS.2023.3329732, 2023.
- C.L. Aldea, R. Bocu, A. Vasilescu. **Relevant Cybersecurity Aspects of IoT Microservices Architectures Deployed over Next-Generation Mobile Networks**. Sensors 2023, 23(1), 189; https://doi.org/10.3390/s23010189, 2022.
- R. Bocu, D. Bocu, M. Iavich. **Objects Detection Using Sensors Data Fusion in Autonomous Driving Scenarios**. Electronics 2021, 10, 2903. https://doi.org/10.3390/electronics10232903, 2021.
- R. Bocu, C. Costache. **A Homomorphic Encryption-Based System for Securely Managing Personal Health Metrics Data**. IBM Journal of Research and Development ISSN 0018-8646, Volume 62, Issue 1, pp. 1:1-1:10, 2018.
- R. Bocu, M. Iavich, A. Gagnidze. **Real Time Self-developing Cybersecurity Function for 5G**. Proceedings of the Conference „Advanced Information Networking and Applications", 2022.
- R. Bocu, M. Iavich, S. Tabirca. **A Real-Time Intrusion Detection System for Software Defined 5G Networks**. Proceedings of the Conference „Advanced Information Networking and Applications", 2021.
- R. Bocu, A. Kerestely, A. Baicoianu. **A Research Study on Running Machine Learning Algorithms on Big Data with Spark**. Proceedings of The 14th International Conference on Knowledge Science, Engineering and Management (KSEM), 2021.

- C. Costache, O. Machidon, A. Mladin, F. Sandu, R. Bocu. **Software-Defined Networking of Linux Containers**. IEEE Computer Society RoEduNet Conference, 2014.

În continuare, este relevant să menționăm că numeroase activități complementare pot fi menționate, care îmbogățesc și adaugă valențe importante relativ la activitatea de cercetare științifică principală. Astfel, fac parte din comitetele de evaluatori ale multor jurnale prestigioase indexate de Clarivate/Web of Science. Câteva jurnale selectate din respectiva listă sunt: "Journal of Network and Computer Applications", "IEEE Transactions on Dependable and Secure Computing", "IEEE Access", "International Journal of Computers Communications & Control".

Am depus propuneri de proiecte de cercetare științifică pe care le-am câștigat și am obținut respectivele finanțări de la acele organizații și instituții semnificative. Astfel, de exemplu, am obținut finanțarea care a susținut activitatea de cercetare științifică desfășurată la doctorat, în Irlanda. Instituția care a acordat finanțarea a fost Guvernul Irlandei. Mai mult, am obținut, din partea NATO, în urma unei competiții foarte riguroase, finanțarea pentru un proiect de cercetare avansată. Scopul fundamental al acestui proiect este reprezentat de crearea unui cadru complex privind asigurarea confidențialității datelor, inclusiv în contextul apariției calculatoarelor cuantice. Datele de identificare ale acestui proiect sunt: „**NATO SPS G7394 - Post-quantum Digital Signature using Verkle Trees**".

Am fost membru în echipele multor proiecte de cercetare științifică, care au fost finanțate de Uniunea Europeană, Guvernul României, Guvernul Georgiei, printre alte organisme. Este important să menționăm că în perioada Octombrie 2007-Octombrie 2010, am fost profesor invitat la *National University of Ireland, Cork*, în Departamentul de Informatică. Acolo, am susținut activități didactice legate de cursuri, laboratoare și pachete de lucru ale unor discipline relevante din domeniul Informatică. Această universitate se află printre cele mai bune 200 de universități din lume.

Am fondat grupul de cercetare științifică "High Performance and Cloud Computing", care funcționează sub auspiciile Universității

Transilvania din Brașov, România. Astfel, o prezentare, în sinteză, a acestui grup de cercetare, este disponibilă pe site-ul Facultății de Matematică și Informatică a Universității Transilvania din Brașov: Grupul de cercetare HPCC . Este important să menționăm că acest grup de cercetare a întreprins o intensă activitate de cercetare științifică în domeniul de interes definit, reflectată prin numeroase articole științifice publicate în jurnale indexate Web of Science/Clarivate cu un proces riguros de evaluare, precum și în volumele unor conferințe științifice prestigioase indexate în CORE (https://portal.core.edu.au/conf-ranks/).

Am fost membru în mai multe comisii pentru evaluarea tezelor de doctorat. Astfel, cea mai recentă asemenea participare a fost la Kaunas Institute of Technology, în Lituania, unde o teză de doctorat a fost susținută în cadrul Departamentului de Informatică. De asemenea, am propus și am supervizat un număr special al jurnalului Clarivate "Symmetry" (https://www.mdpi.com/journal/symmetry).

Am cultivat, în permanență, colaborarea cu actorii industriali relevanți. Astfel, începând cu anul 2010, am colaborat cu General Magic Brașov, Siemens Corporate Technology, In-Tech Engineering Services și Siemens Industry Software. Sub auspiciile acestor colaborări, am coordonat proiecte semnificative de cercetare științifică, ale căror rezultate au sprijinit activitatea unor actori industriali importanți și care au fost publicate în jurnale prestigioase. Astfel, articolele relevante sunt enumerate, iar câteva dintre ele sunt prezentate în detaliu în **secțiunea B-i** a tezei de abilitare. De asemenea, în colaborare cu aceste companii partenere, am organizat școli de vară, precum și alte activități extracurriculare care au îmbunătățit abilitățile profesionale ale studenților mei de la Informatică.

Aspectele relevante ale carierei profesionale vor fi cultivate și dezvoltate continuu. Astfel, strategia didactică validată empiric va fi optimizată în strânsă legătură cu realitățile activităților zilnice. Este important să se rețină că principiile fundamentale ale acestei strategii didactice inovatoare au fost descrise în primul capitol al **secțiunii B-ii**, care face parte din corpul principal al tezei de abilitare.

Întreaga paletă a activităților științifice complementare va fi abordată, iar o anumită importanță va primi supervizarea grupului de

cercetare științifică “High Performance and Cloud Computing”. Astfel, anumite proiecte de cercetare științifică, pe care acest grup le administrează, sunt legate de arhitecturi avansate pentru gemeni digitali (*digital twins*) bazate pe folosirea microserviciilor, modele algoritmice de învățare automată și arhitecturi software adecvate, care abordează realități complexe, din punct de vedere informațional, ale lumii reale. De asemenea, este important să menționăm proiectele de cercetare interdisciplinare, din aria de interes a bioinformaticii, bioingineriei și a specialităților biomedicale.

Activitatea de cercetare științifică va fi continuată relativ la direcțiile de interes care au fost descrise, atât în corpul principal al tezei de abilitare, cât și în acest rezumat. Cu toate acestea, considerând proiectele de cercetare științifică care au fost câștigate, precum și activitatea desfășurată în cadrul grupului de cercetare „High Performance and Cloud Computing”, se poate spune că următoarele teme de cercetare vor fi, cu precădere, avute în vedere pe termen mediu: învățare automată și inteligență artificială, tehnici avansate pentru confidențialitatea și securitatea datelor, modele eficiente de criptate și semnătură digitală, inclusiv rezistente la atacurile calculatoarelor cuantice (*post-quantum*). De asemenea, proiectele de cercetare planificate vor include evaluarea aplicațiilor și arhitecturilor software instalate peste infrastructuri și rețele de date de generația următoare.

B. Scientific and professional achievements and the evolution and development plans for career development

B-i. Scientific and professional achievements

Introduction

Computer Science, as a member of the formal sciences branch that studies computation, information, and automation, is inextricably

influenced by the complex nature of the real-world that it has to model in the realm of all relevant scientific research and experimental processes. Therefore, it is immediate to note that interdisciplinary scientific research approaches are required in the scope of Computer Science. Although a dedicated computer scientist, the author of this thesis essentially believes that the ability to design and manage interdisciplinary scientific research projects is fundamental for any efficient computer scientist, who is firmly interested in the proper mapping of the research efforts' results to the requirements of the specific real-world use case scenarios. Therefore, this principle has been considered since the beginning of the author's scientific career.

Thus, the doctoral research, which was conducted at the National University of Ireland, Cork, relates to a topic from the scope of Bioinformatics, which concerns the design, implementation, testing, and optimization of efficient algorithms that are used to process large networks of human proteins [1]. This had the goal to discover the causal links between mutations that occur at the level of the amino acids, which structurally and functionally define proteins, and the inception and development of problematic health conditions, such as cancer. Considering the relevance of the PhD scientific research project for the development of the scientific career, which this thesis describes, the following paragraphs of this introductory section include related relevant remarks.

The post-genomic age brought interactome networks at the front line of the biological research. This development is a direct consequence of its discovered influence over all the essential physiological processes that occur in humans and the vast majority of biological entities. Furthermore, as a natural consequence, the study of some complex contemporary diseases can greatly benefit from researches on the existent proteomic data. The central hypothesis, which governs all the developments presented in the PhD thesis, is based on the assumption that carcinogenesis is significantly and decisively influenced by mutated proteins.

The main contributions of the PhD thesis are separated in two categories. The first one refers to the speedup of the protein data analysis, at the algorithmic and programming level. Therefore, it

concerns the protein data analysis method designed during the course of the research activity. The second category comprises the results that were obtained following the application of the proposed protein data analysis method on the aggregated human protein data set.

Among the multiple theoretical and practical contributions, which this PhD thesis describes, the concrete suggestions regarding the re-engineering of certain cancer drugs are worth to be mentioned. Thus, cancer chemotherapy has gradually improved with the development of novel antitumor drugs and has profound, positive results, when applied to many hematologic malignancies, and some solid tumors, especially germ cell and some childhood malignancies. While treatment of certain malignancies with chemotherapy has been successful and encouraging, the effectiveness has often been limited by drug resistance of tumors and by side effects on normal tissues and cells. In fact, many tumors are intrinsically resistant to many of the more potent cytotoxic agents used in cancer therapy. Other tumors, initially sensitive, recur and are resistant not only to the initial therapeutic agents, but also to other drugs not used in the treatment. Because of the serious problem that is induced by the clinical drug resistance, much effort has been put into broadening the perspective on the mechanisms of drug resistance in cancer cells [2].

The remarks in the previous paragraph suggest that the focus is mainly directed towards the creation of cancer drugs that act by inhibiting the inflow of nutrients to the tumor, through some chemical compounds that are released into the circulatory system. Unfortunately, this approach has its drawbacks. First, the cancer drugs that are produced according to this procedure generate numerous side effects for the patient's organism, due to their strong chemical nature. Second, the tumor itself develops self-defense mechanisms against the drugs that are administered, which greatly reduce the efficiency of the remedy over a short period of time.

In light of the discoveries produced by this PhD research, it can be stated that rather than concentrating on chemically blocking the nutrients supply to the tumor, an efficient cancer drug should focus on preventing the mutations that may occur to the essential regulatory proteins, such as BLCAP in the case of bladder cancer. If the disease is

already progressing, the drug should be capable to revert the affected protein's gene information back to its original state, thus allowing the apoptosis (programmed cell death) to resume normal function.

This PhD research has shown that human proteins, which are related to cancer development, exhibit a network topology and internal properties that are different from those of proteins that do not get mutated in cancer. The observation is primarily based on the study of the aggregated protein-protein interaction network that has been introduced and described. It is based on existent biological data that is publicly available in various protein data sets. Thus, it has been proved that proteins, which suffer mutations in a carcinogenetic process, belong to the most important protein communities in the interactome (overall network of proteins). Furthermore, the proteins that get mutated are located centrally in these globally central protein communities. Therefore, the normal function of the biological pathways and cellular apoptosis processes cannot continue to occur normally and, thus, the uncontrolled cancerous cell division has proper conditions to initiate and proliferate.

The integrated protein data analysis method, which was proposed, allows for an in-depth and informative processing of the available protein data to be performed, using available computational resources. Actually, it is not necessary to use complicated and expensive computer architectures, as the steps that have to be followed involve operations that can be handled even by current desktop computers. This possibility, which has been out of reach relative to regular research groups until recently, is a consequence of the efficient betweenness and community detection algorithms that were designed, implemented, and integrated into the protein data analysis method.

The effectiveness of the analysis method is practically assessed in relation to BLCAP, which is a newly discovered bladder cancer-related protein. The results prove the method is not only suitable for the analysis of the involved protein itself, but it is also useful for the discovery of potential candidate proteins that have an influence on the evolution of the same type of cancer. Naturally, the method can be

used in order to assess carcinogenesis, at proteomic level, in the case of any human organ or system.

The contributions of this PhD research process to the study of protein networks can be grouped in two categories: methodological advances and practical results. The practical developments provide significant insights into the global network properties of cancer proteins, and can be used to guide experiments towards certain sub-communities of the interactome, which are likely to regulate cellular processes that are related to cancer.

It is immediate to note that the proposed contribution is essentially interdisciplinary. It also proves that the algorithmic, programmatic, and mathematical apparatuses, which are fundamental in the scope of Computer Science, may be considered to properly model and solve complex real-world problems. Consequently, the possibility to pursue interdisciplinary research strategies has been continuously considered, relative to the specific of each particular research project, after the PhD degree in Computer Science was awarded.

The research activity that ensued after the award of the PhD degree considered several research topics. Thus, some of the most relevant subjects relate to software engineering, advanced management and security solutions for computer networks, software applications that pertain to computer networks, interdisciplinary research projects in the scope of bioinformatics, machine learning and artificial intelligence, advanced data privacy and security techniques, efficient encryption and digital signature models, including post-quantum approaches, applications and architectures deployed over next generation infrastructures and data networks, personal data privacy solutions, including homomorphic encryption systems. Additionally, certain relevant research contributions relate to autonomous driving solutions, big data processing and large-scale data analytics using machine learning models, the design, implementation and optimization of virtualized networked infrastructures. Furthermore, significant contributions, which were reported through reviewed scientific papers, relate to the study of complex networked structures, which include complex biological networks.

The reference list of papers may be consulted on the following web page: https://www.razvanbocu.bocu.ro/?page_id=18 .

Following, the habilitation thesis is structured according to the following sections. The most relevant research results are presented through the respective peer reviewed papers, and other publicly available resources. The presentation follows the enumerated research topics. Moreover, the personal development plans, both regarding the didactic and scientific research perspectives, are presented. This section also discusses about the author's ability to supervise research teams, and also presents the author's proven ability to write successful scientific research proposals. The last section concludes this habilitation thesis.

Chapter 1. Software engineering

The scientific contributions that pertain to software engineering comprise books, book chapters, reviewed conference papers, and reviewed journal papers.

The software engineering books, which were published after the PhD degree was awarded, concern comprehensive analyses of the software engineering scope [3], while relevant practical solutions are proposed relative to various real-world use cases. Furthermore, the book [4] expands on the applied discussion concerning the proactive detection of possible flaws during the various modelling stages of the software development processes. Additionally, the book effectively and consistently expands on existing solutions, and consequently introduces new conceptual elements, and also experimentally validated practical software systems development solutions.

Following, we shall expand on the most relevant aspects of the problematic presented by chapter [5], which relates to a discussion about the role of the WEB technologies relative to the communication's streamlining and diversification (CSD) between the actors of a learning system. Thus, the main idea that is valued in this chapter is that a software system of the type CSD is ideal for the elegant trafficking of

some information resources and services, without being engaged in the methodic monitoring of a knowledgebase development and consultation.

The knowledgebase exists in the teacher's mind. The digitization tools of this knowledgebase constitute a distinct and complex problem. The student excels through his willingness towards communication. This native willingness must be efficiently speculated through making use of proper technologies. Thus, a CSD system unites the two potentials, and a system with a dynamic that is usually difficult to estimate can be born. We certainly speak about the usefulness of such an approach in the academic and industrial environments. In the daily people's life, one can already notice some mutations, which are primarily behavioral, as a direct consequence of the availability of communication technologies that were unthinkable at the beginning of the information era. The social networks ultimately are, genuine communication tools, in which, although this fact is not readily visible, the preoccupation for streamlining and diversification is constant. This assertion could be better understood if we explored the structural foundations of the "social network" systems. It is relevant to note that more details are provided in **Figure 1**.

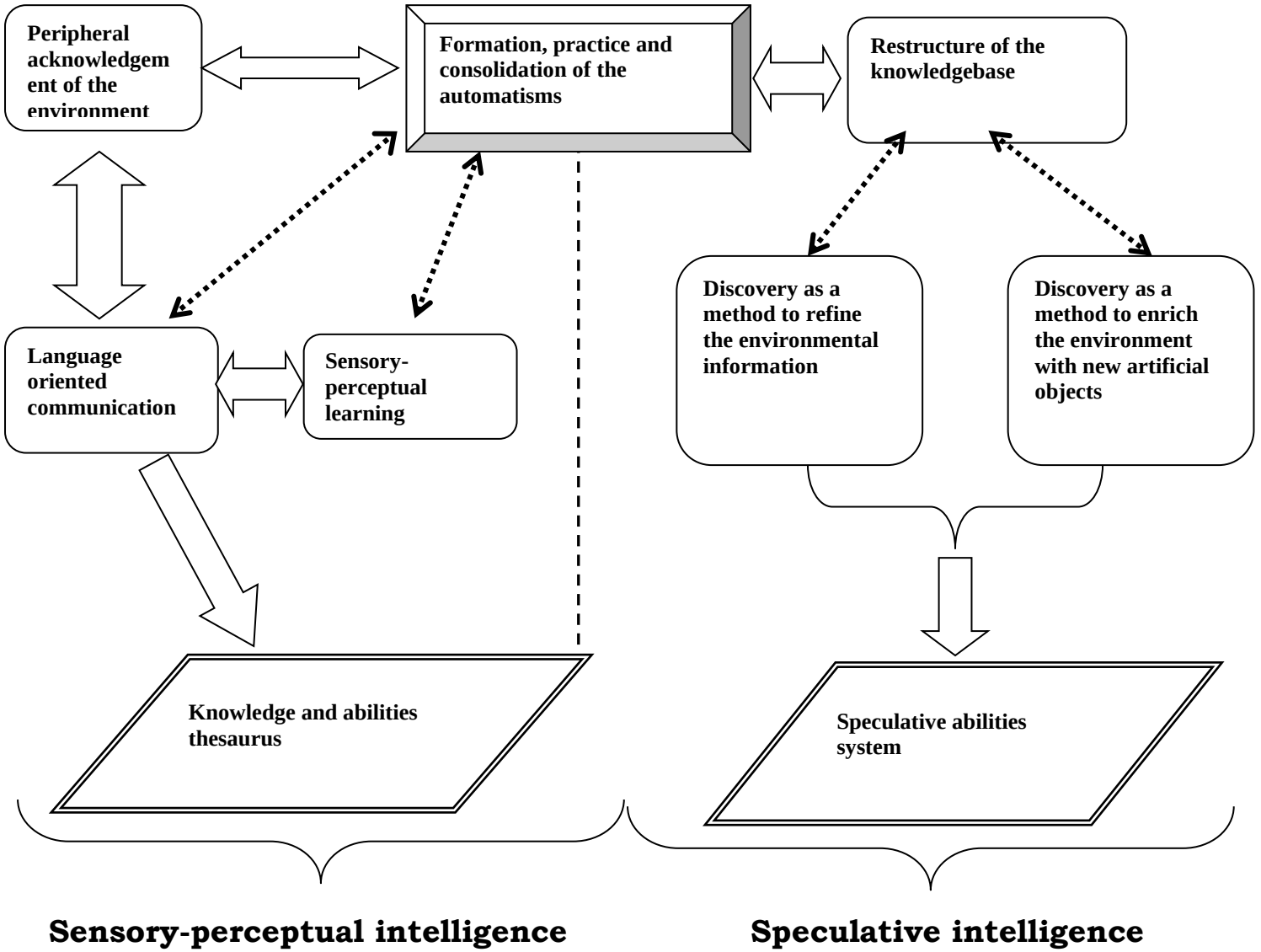

Figure 1. The approximate map of the behavioral invariants, based on which the intelligent systems-related explanation can be commenced.

It is stated that the developers of software systems are used to the unpleasant situation, which may be encountered when insufficient attention is paid to the problems that may arise when ensuring an optimal streamlining of the IT-based communication exists. The situation that we refer to is that in which an IT system that generously offers functional capabilities falls short as a consequence of its cumbersome, ambiguous and insipid interface. Therefore, the first recommendation that we may suggest in a fully knowledgeable manner to the developers of the IT systems is the elaboration of some user-

system interfaces, whose structure respects a few minimal requirements:

- The semantic and syntactic stability of the interfaces. The compliance to this requirement guarantees a natural streamlining potential of the communication user-system.
- The removal of the interfaces' ambiguities up to the limit that is acceptable at the education level of the user;
- The elimination of the quantitative redundancies in the demarche to realize the user-system interfaces;
- The rational usage of the qualitative redundancies in the demarche that involves the realization the user-system interfaces;
- The minimization of the syntactic streams that are associated to the interfaces' usage;
- The maximization of the semantic streams that are associated to the interfaces' usage.

Furthermore, in order to support the user-system communication's streamlining, there are necessary technologies that operate in the background of the IT systems, with the goal to facilitate a visible contribution regarding the user's comfort. Among these possible technologies, we enumerate a few that seem to be urgent and validated in practice:

- The realization of a framework that relieves the user of the possibility/obligation to interact with documents that are represented according to different standards (.doc, .pdf, .rtf, .wp, etc.), which favors focusing the attention on the document's content, in other words there are created favorable conditions for the communication's streamlining. When we say document, we refer to any product in an electronic format, which represents according to a certain standard a certain type of semantics. In other words, under the syntagm document we can designate texts, presentations, images, sounds, spreadsheets, models diagrams, projects, etc.
- It is also relevant to mention the elaboration of some highly abstracted APIs for the communication capabilities of the type

"voice over IP" could also contribute to the communication's streamlining;

- We add to the list of the technologies that favor CSD the elaboration of some framework solutions for solving some practical problems, which may appear during the activity that concerns the documents, and hyper documents manipulation.

Finally, but obviously not the least important, it is significant to mention the elaboration of some technologies that timidly, but increasingly wider, open the gates for the semantic analysis of the text-oriented messages content.

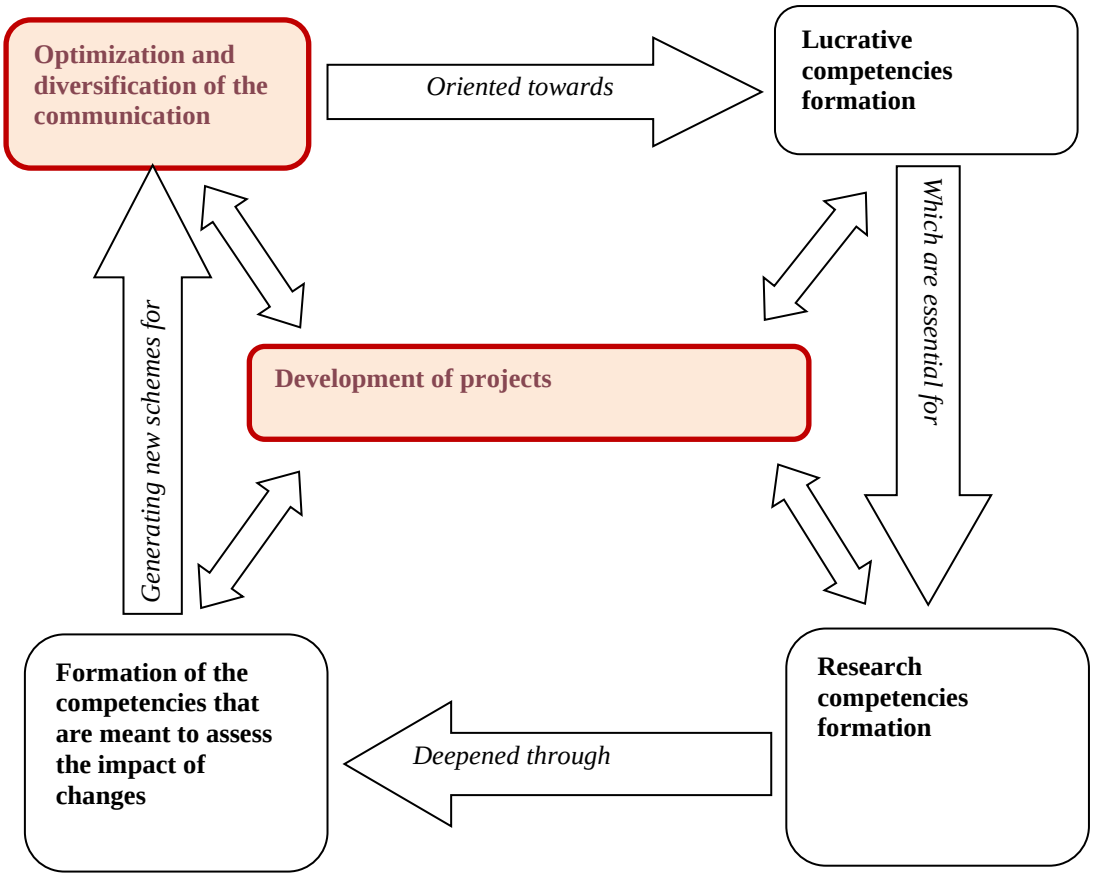

Figure 2. The architecture of a projects-oriented learning system.

Furthermore, the paper discusses about projects-oriented learning systems, whose logical architecture is presented in **Figure 2**. Thus, the key concepts relative to a projects-oriented learning system are project, communication, lucrative competencies, research

competencies, and competencies that are related to the ability to assess the impact of changes, which are generated by the implementation of the projects. Moreover, Figure 2 suggests that the proposed development model relates to the paradigm of the spiral. Thus, it is founded on an organic iterative model, which determines the optimal operation of the whole system.

Moreover, the paper describes the details of the CSD software architecture model, and a visual representation is contained in **Figure 3**.

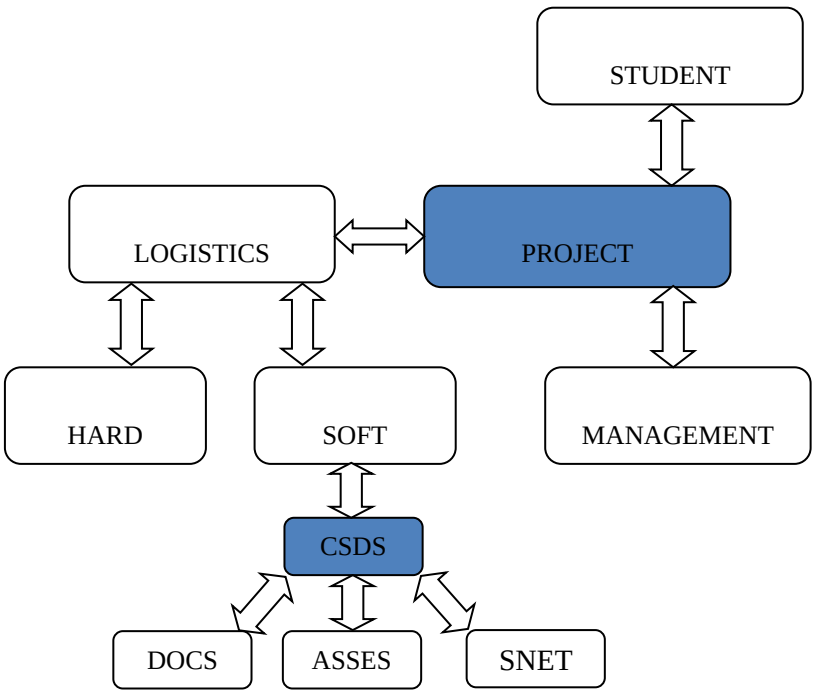

Figure 3. The architecture of a learning system that is developed according to the CSD model.

Thus, Figure 3 considers the following acronyms: communication streamlining and diversification subsystem (CSDS), documentation system (DOCS), assessment system (ASSES), internal social network (SNET), all of them being fingerprinted by CSD.

It may be asserted that this chapter conducted a structured analysis relative to a theoretical model that concerns the communication's streamlining and diversification in the case of the computer-aided learning systems. We drafted the theoretical limits of this initiative and we indicated the available technological resources, eventually together with their limitations.

Additionally, we specified, at an architectural level, a learning system that is conceived according to the CSD model, with the goal to

indicate the place and the real-world relevance of the social networks in the effort to realize an e-learning system. The association of this approach with the idea of projects-oriented learning intended to be an additional argument for the real-world relevance of the CSD-oriented learning systems.

It is immediate to note that the materialization and the continued development of the ideas, which have been presented, require consistent analysis, design and implementation efforts, on which computer specialists, the education experts, and other categories of interest holders have to reach an agreement.

Furthermore, it is significant to note that relevant software engineering contributions are reported in papers [6], [7], [8], [9], [10], [11], and [12], which have been published following the award of the PhD degree. Thus, paper [6] approaches fundamental aspects concerning the design of efficient software architectures, which are precisely tailored towards the specific of the modeled real-world use case, while article [7] discusses on the role of the conceptual invariants regarding the prevention of the software artefacts' obsolescence. Furthermore, article [8] thoroughly analyzes the possibility to design efficient software interfaces, which are proper for implementation of software artefacts that are mapped on the respective real-world use case scenario. Additionally, paper [9] further expands on this problematic through a conceptual and applied discussion that relates to the modeling of interface-oriented software systems. It is also relevant to note that the contribution, which is reported in paper [10], approaches significant aspects regarding the design and implementation of strongly project-oriented learning software systems. It is also relevant to note that the contribution, which is presented in article [11], thoroughly evaluates the utilization of abstractization, as a fundamental strategy for the design of robust, reliable and valid software systems architectures. Moreover, paper [12] reports a real-world use case, which concerns the conceptual foundations of code rationalization through a case study in the functional programming language Haskell. The following sections elaborate on this particular article.

The Haskell programming language is a stylish representative of the functional paradigm. The literature has presented over time its many advantages. Although the language did not receive the attention of the industry, the academic environment and the enthusiasm of the fans kept the Haskell project in the attention of a large number of curious, enterprising and contributors to the huge potential of the functional paradigm. This paper aims to present Haskell templates for solving problems of general interest. In fact, it is not the problems that are the target of the approach, but the algorithmic implications that the newcomer to Haskell has to face. The Haskell language does not claim to rethink problem-solving algorithms, but rather to choose the optimal template for implementing these algorithms in Haskell. More precisely, as is natural, there are differences between the implementation templates of an algorithm in a procedural language, such as Java, C++, or C#, and Haskell. The presentation of such templates can help to shorten the time needed to familiarize newcomers with the style of coding that is specific to Haskell. What are the reasons why the Haskell language is worth the effort to adapt with certain recipes for the implementation, at first sight unintuitive, of some algorithms encountered in many other languages? Are they the amazing compactness of the code, its memory usage efficiency, its code readability, or remarkable theoretical foundations? These aspects are considered in this paper.

The world of programming languages is constantly expanding. Perhaps, this is also why programming in any language, at present bears, in many respects, the imprint of the theoretical and practical experience gained by each programmer. The methodologies of many software companies, aware of the disadvantages of such an approach, come with improvements, aiming to comply with predominantly formal standards, which concern code writing. These standards are of real interest in terms of project management. Concerning the software system solution engineering, other approaches are needed that could contribute to the optimization of the result of the programming activity, in general. Nevertheless, the programming activity is only a stage of the complex process, which realizes a software system. The quality of the programming activity is even strongly influenced by other

activities, such as analysis, design, testing, which represent only the essential processes from a technical point of view. This paper discusses on the idea of code rationalization, and it takes into account the essential factors that can influence the outcome of the rationalization effort. The syntagm code rationalization encompasses aspects that are discussed in the related scientific literature, and pertain to requirements that are specific to a certain phase during the development of the software system, such as: compliance with the fundamental principles of the software systems engineering, elaboration and systematic utilization of design patterns**,** elaboration and systematic utilization of implementational patterns, and the utilization of the optimal programming paradigm in order to solve a certain type of problem (object orientation, aspect orientation, component orientation, services orientation, etc.).

It may already have been intuited that the idea of code rationalization, considering relatively isolated perspectives, is already a consistent presence in the world of programmers. This is also confirmed and augmented by the latest versions of established programming environments, which already incorporate support for the use of architectural templates, design templates, partial automation of testing or implementation. Nevertheless, the theoretical reasons, although usually well-founded, are not always a priority at the level of the so-called good practices. The systematic following of the exhortations of these theoretical reasons could become attractive from a practical point of view if we elaborate software development scenarios in which the rationalization of the code written in a certain language is assimilated to a process of searching for the state of symmetry of the code. This is the state in which the requirements assumed by developers in the process of realizing a software system are harmonized. The random walk in the programming activity favors the appearance of some contradictions between the assumed requirements. Furthermore, significant loopholes are also possible in meeting these requirements. Therefore, in order to systematically monitor these contradictions and to eliminate leaks, this paper proposes a framework that underlies the code rationalization.

Thus, let us consider a software artefact, which may be the object of several requirements. Let us note with ***C*** such a requirement. The identification of a requirement becomes useful from a methodological perspective if it is associated with a metric, so that the requirement is quantifiable**,** and it has at least one associated constraint, let us note it with ***C****.

The specification of a metric for the requirement ***C*** involves, without excessive formalism, the identification of two fundamental components of the metric: the list or the domain of valid values that are associated to requirement ***C***, and the method to calculate the current values, which are associated to requirement ***C***.

Concerning the constraint ***C****, it is specified through a type, and the limit value of the requirement, let us refer to it as LV. We consider that there are two fundamental types of constraints: constraints of the type **lower threshold (LT)**, constraints of the type **upper threshold (UT)**.

Thus, a constraint of the type **UT** is satisfied if the current value that is associated to the requirement is less or equal to **LV**. Furthermore, a constraint of the type LT is satisfied if the current value that is associated to the requirement is greater or equal to LV. Obviously, the limit value LV is established by the developer in such a way that the software artefact meets the expectations of its potential user.

We consider that such an approach regarding the concept of requirement mandates the consideration of the generic code rationalization processes. Additionally, an automated framework that may support the actual code rationalization processes can be specified. The systematic monitoring of the progresses concerning the code rationalization turns from a conscience-related fact into a task that is considered during the planning of a software project.

The assimilation of a program written in a certain language with a system implies that the choice for the optimal circumstantial version (OCV) of the program is an approach whose success can be made more efficient if the programmer capitalizes on the contribution of code rationalization in specifying it. The following paragraphs discuss about

the usefulness of the concept of symmetry relative to the programming activity.

In order to ensure a more precise presentation, let us state that if $C_1^*, C_2^*, \ldots, C_n^*$ are the constraints that are considered during the implementation of a programme, then we can state that the programme is in an OCV state if the decisions that determine the code writing ensure the fulfillment of all the constraints at an appropriate circumstantial level. Conversely, a programme that does not fulfill at least a constraint is specified in a non-optimal state.

Beyond the many recommendations that the literature makes to developers in general and, consequently, to programmers, in this paper we focus on the following three issues, considered in many papers as a kind of resistance structure of software systems: the search for the optimal abstraction scheme of the problem's data (OASD), the search for the optimal algorithmic scheme (OAS), which values the potential of OASD, and the search for the optimal implementation scheme (OIS).

As there are not many programs that have nothing to do with the world of data, it goes without saying that the realization of a program invariably involves the effort of finding the optimal scheme for the abstraction of the data. It is relevant to note that the first type of abstraction of a program's data is the conceptual approach by which data from external format (associated with certain circumstances) are progressively transformed into an internal format.

The second type of abstraction of a program's data is the conceptual approach by which data from internal format are returned to external format, which is associated with certain circumstances. The challenge in this type of abstraction is to find the minimum core of redundancies with which to obtain a maximum impact on the external user of the data. Practice clearly shows that this type of abstraction must, at the same time, control the impact that the excessive specification of the external data format can have on the associated algorithmic schemes.

There are many requirements we can have when critically evaluating algorithmic schemes to capitalize on the potential of the types of data we work with. Here is a list of essential requirements: the execution time, the internal memory requirements, which is tightly

connected to the processed data's structure, the index for favoring some optimal subsequent data processing operations, and the coding simplicity.

If we consider only these four requirements, then we understand that choosing the optimal algorithmic scheme for processing a type of data (in certain circumstances) is an activity in which the effort of abstraction (in another plane of course) is intense, often necessarily assorted with exemplary bursts of creativity, which determine basic pieces of real problem-solving templates of a certain type. If requirements, such as those exemplified above, are associated with constraints, then it is clear that the need to meet a constraint may lead us to rethink the data on which the algorithmic scheme operates. The naturally iterative searches continue until the constraint is satisfied.

The translation of OASD and OAS into a certain programming language is partially dependent on the level of the programmer's expertise relative to the language. The choice of the optimal variant for implementing OASD and OAS decisions is a search process in which the acquired expertise must be related to the state in which the code is from OCV point of view.

The illustration of the content and usefulness of a code streamlining approach in Haskell considers the implementation of the optimal Haskell code to determine how often words appear in a list of words. This code could be of interest in a project dedicated to the analysis of texts in order to identify some patterns in the use of words. At the same time, we specify that, for the function that generates the list of frequencies with which the words appear in the randomly generated text, there is a requirement that its execution time respect the constraint of not exceeding 2 seconds at a test volume of 2,000,000 words. Thus, given a text, let us make the following assumptions: the text we intend to analyze is stored in a file, the code that solves all tokenization issues is assumed to exist (using this code, we can generate a list that contains all the words in the text). Furthermore, we will assume that sorting the list of words, if necessary, is solved. This is a well-founded assumption, since in the <Prelude> module, which is delivered and accessible with the installation of the Haskell platform, there are several predefined solutions, including the <sort> function.

In general, but also in this case, algorithmic considerations can contribute to a positive approach to coding activity. The specifications required in accordance with the requirements of the case study are: the list of words is sorted; the occurrences of a certain word are consequently grouped. Additionally, considering this basic logical structure, the first iteration of the algorithm may consist of the following steps: the first element of the list is fetched; considering this first element, the number of occurrences in the list is determined; the occurrences of the first element are deleted from the initial list. The iterative process continues until the target list becomes empty.

Coding by a beginner in Haskell would involve the use of three functions, through the composition of which, the problem of determining the frequency with which each word appears in the given list can be solved. As you can see below, these three functions could be: genFIt_1 (to determine the list of frequencies); contapIt_1 (to count the number of occurrences of a word in the list, making the assumption that the occurrences are grouped); delap (to delete the occurrences of a word in the list, which is the word on the first position in the list).

```
-- Iteration 1
data Token=Token String|Err
     deriving (Read,Show,Eq,Ord)
genFIt_1::[Token]->[(Token,Int)]
genFIt_1 []=[]
genFIt_1 lt=((head lt),(contapIt_1 (head lt)(tail lt))):
            (genFIt_1 (delap (head lt) lt))
-- The function determines the number of consecutive
-- occurrences of a certain Token
contapIt_1::Token->[Token]->Int
contapIt_1 _ []=1
contapIt_1 tok (tc:rl) |(tok==tc)=1+(contapIt_1 tok rl)
                       |otherwise=1
-- The function deletes the consecutive occurrences
-- of a certain Token
delap::Token->[Token]->[Token]
delap _ []=[]
delap tok (tc:rl) |(tok==tc)=(delap tok rl)
                  |otherwise=(tc:rl)
```

The above code illustrates part of the virtues of Haskell language regarding its code rationalization capabilities. Thus, the type Token, which is declared with the reserved word <data>, in addition to representing the programmer's choice as to the name of the type

constructor, the value constructor and the way a word is represented in internal format, requires the compiler to generate instances of the classes of types Read, Show, Eq, Ord, which would allow their use in the context of the type Token.

Furthermore, the function genFIt_1, according to the definition of its type, has as input a list of Token data and as output a list for which each element is a tuple, the first element of the tuple being a Token data and the second being the frequency with which the Token appears in the list. This way of defining the type of the genFIt_1 function will put its mark on the successive schemes for specifying the behavior of the genFIt_1 function.

It is also relevant to note that the problem with this way of implementing the genFIt_1 function is that it is based on the contapIt_1 function, which for each word, traverses twice the group to which it belongs, once when counting the occurrences of a word, and once again when deleting the occurrences of the word. This becomes problematic in the case of test lists with many elements in terms of the necessary execution time.

Table 1. The data that is generated by the function genFIt_1.

Tested function	**Number of elements in the test list**	**Execution time in seconds**	**LV**	**Type of constraint**
genFIt_1	2,000,000 words	5.78 s	2 s	UT

As it can be observed in **Table 1**, considering a test with 2,000,000 elements, the execution of the genFIt_1 function takes about 11.72 seconds. It is a reality that a demanding programmer should not be accustomed to. Many queries can be categorized as inadmissible in terms of response times if they were based on a function such as genFIt_1. Given the value of LV, the contribution of the genFIt_1 function to the symmetry of the code of an application that incorporates it, even if it is one-dimensional, is unacceptable.

```
-- Iteration 2
genFIt_2::[Token]->[(Token,Int)]
genFIt_2 []=[]
genFIt_2 lt=((head lt),ftok):(genFIt_2 rl)
            where (ftok,rl)=contapIt_2 (head lt) lt
contapIt_2::Token->[Token]->(Int,[Token])
contapIt_2 _ []=(0,[])
contapIt_2 tok (tc:rl)=if (tok==tc) then ((1+fi),rli)
                                    else (0,(tc:rl))
                       where (fi,rli)=contapIt_2 tok rl
```

Starting from the observation that the source of excessive processor time consumption is the contapIt_1 function, the natural solution seemed to us to be an amendment to the signature of the contapIt_1 function, obviously having implications for its implementation. The change caused the execution time of the genFIt_2 function to decrease to approximately 0.80 s, according to **Table 2**, which means that the constraint for which LV was set to 2 s is properly satisfied.

Table 2. The data that is generated by the function genFIt_2.

Tested function	**Number of elements in the test list**	**Execution time in seconds**	**LV**	**Type of constraint**
genFIt_2	2,000,000 words	0.80 s	2 s	UT

```
-- Iteration 3
genFIO::[Token]->IO [(Token,Int)]
genFIO lnr=do
{
  if (lnr==[]) then return []
               else do
     {
       (nrap,rl)<-return (contapIt_2 (head lnr) lnr);
       lsi<-genFIO rl;
       return (((head lnr),nrap):lsi)
     }
}
```

The genFIO function is indispensable when the methodical observance of the monadic style of code writing is desired. It is interesting to see that the logic of using the monadic style brings advantages both in terms of the necessary ingredients and, somewhat

surprisingly, in terms of execution time, according to the data in **Table 3**.

Table 3. The data that is generated by the function genFIO.

Tested function	**Number of elements in the test list**	**Execution time in seconds**	**LV**	**Type of constraint**
genFIO	2,000,000 words	0.86 s	2 s	UT

The approach presented is intended to be a concrete proof of the fact that the programming activity has two fundamental dimensions: the circumstantial dimension, and the speculative dimension.

Considering the circumstantial dimension, we described a case study in order to draw attention to the close connection of the products of the human mind with the well-being of the environment to which man himself refers. The metaphorical well-being we are talking about is the result of a search process, which can be even tree-like, at the end of which we can live the satisfaction of finding a shorter, less invasive, less polluting way of relating to the world we belong to. The related search process led us to the conclusion that a possible optimal solution to generate the list of frequencies with which words appear in a text is, according to taste or needs, the functions genFIt_2 or genFIO. The difference between them is measured in tenths of a second in the case of a list of two million items. This level of performance should be satisfactory at the moment. The presented code occasionally uses lists of integers instead of lists of words, as test data.

Furthermore, relative to the speculative dimensions, it can be stated that a genuine programmer knows that the odds for success during a research process are higher if the speculative module of thinking is properly used. The immersion into the concrete without being endowed with an adequate speculative armor, which effectively combines thought templates with bursts of creativity, does not guarantee the success of a research approach. Even if humanity is going through a period in which the respect for speculative thinking is declining, the opinion of the authors of this paper is that without the contribution of speculative thinking humanity, as a species, cannot evolve.

The consideration of the conceptual framework that is presented, and the way it is used in the case study implies that the next possible step is concerned with the creation of a framework, which would be useful in order to monitor the software artefacts' state considering the OCV perspective.

It is relevant to note that the analyzed contributions individualize themselves from the existing similar solutions both regarding the theoretical and conceptual perspectives, and also relative to the practical solutions that are provided. Therefore, due to the inherent space constraints, this assertion is thoroughly and comprehensively justified in the referenced literature.

Chapter 2. Advanced management and security solutions for computer networks

The contributions that relate to this category are described in numerous peer reviewed papers. Thus, some of the selected contributions are reported in papers [13], [14], [15], [16], [17], [18], [19], and [20].

The approach that is described in article [14] relates to a real-time monitoring and management solution for hardware and software resources in heterogeneous computer networks through an integrated system architecture, while paper [15] proposes a real-time intrusion detection and prevention system for 5G and beyond software-defined networks. Both presented approaches relate to innovative algorithmic, implementation, and deployment solution, which mediate a proper management and efficient operation of high throughput next generation data networks.

The portfolio of published papers report solutions that pertain to the device tracking threats, which occur in next generation data networks [16]. Moreover, paper [17] comprehensively presents a machine learning-based real time integrated system, which filters and labels potential intrusion attempts in software defined 5G data networks. To the best of our knowledge, this is one of the very few similar solutions, which has been fully validated using data provided by a mobile telecommunication operator. Additionally, it is relevant to

note that a semantically related paper, which has received a large number of citations, is linked to reference number [18]. Also, an equally influential approach is reported in article [19], which thoroughly presents a virtualized networking infrastructure that considers software defined Linux containers. Moreover, paper [20] reports an experimentally validated solution regarding the preservation of the IP addresses' reputation, if they are used to relay email data traffic. It is important to note that, to the best of our knowledge, this is one of the few solutions, which proposed a fully functional IP addresses reputation preservation system that was fully validated considering a comprehensive experimental approach.

It is relevant to mention the comprehensive solution, which was reported in paper [13]. This pertains to the problematic of enhanced detection of low-rate DDoS attack patterns using machine learning models. Considering the uniqueness of the proposed solution, and the extensive experimental validation process that was applied, this solution is described in detail.

Low-Rate Distributed Denial of Service (LRDDoS) attacks represent a problematic aspect of network security research, considering that they are determined by periodic slightly variable data pulses, which degrade the overall network responsiveness and stability. Considering the limited amount of data, which are processed by a particular network client, it is difficult to train a proper detection model. Consequently, Federated Learning (FL) represents a suitable approach, which mediates the collaboration between several network clients. The integrated approach that was reported in the paper published by the reputable "Journal of Network and Computer Applications" presents an enhanced LRDDoS detection model, which is based on an asynchronous federated learning approach. The proposed model has been implemented, and the resulting integrated system has been thoroughly evaluated considering the real-time corporate data traffic of a major Romanian Information Technology company, which administers multiple branches. The results of the experimental process demonstrate that the described model performs better than similar existing approaches, considering the assessed accuracy and other relevant performance metrics, which are presented. Consequently, this

significantly decreases the load that is placed on the data network, which effectively reduces the probability for the LRDDoS attack patterns to occur.

The increasing number of Internet of Things (IoT) devices determines a substantial growth on the number of illegitimate data traffic patterns, which include network intrusion attempts and distributed denial-of-service traffic. Relative to IoT devices, the surveyed experimental data suggest that Low-rate distributed denial-of-service (LRDDoS) represents the most common type of illegitimate data traffic pattern. Several types of LRDDoS attacks have been identified lately. Thus, a multi-target LRDDoS attack pattern [21] uses the empty
time slots, which exist between the respective data pulses. This determines a more efficient and destructive behaviour of the LRDDoS attack pattern.

Furthermore, Hivenets [22] are proved to allow for a particular IoT device into a relatively intelligent robotic entity, which is able to make decisions with minimal external input. Additionally, there are two approaches, which are designated as LSTM-CGAN [23] and TTS-GAN [24], which can be employed as data generation tools for the proper simulation of LRDDoS attack scenarios. Thus, LRDDoS is an example of a relatively sophisticated and large-scale attack pattern, which is featured by a consistent time-domain model in a network of IoT devices. Contrary to standard distributed denial-of-service (DDoS) attacks, which are based on large amounts of transferred data, LRDDoS attack patterns are featured by similar amounts of data traffic speed and volume, relative to the usual behaviour of legitimate clients. Consequently, LRDDoS patterns are difficult to detect.

Considering the thorough literature review, which this paper considers, it seems that Blacknurse [25] represents the sole large-scale LRDDoS attack, which consistently manifested in real-world settings. Consequently, there are insufficient relevant data, which can be used to properly train the respective machine learning models. Moreover, during an LRDDoS attack, both the source and the destination of illegitimate traffic are usually affected. Therefore, additional data packets are lost, which further degrades the quality of the implied

training datasets. Thus, a possible solution may be regarded by the aggregation of data, which originate from several data centers. Nevertheless, in practice, such data are protected by strict data privacy agreements. Consequently, alternate solutions are necessary to solve this problem. Thus, federated learning (FL) [26] represents a proper algorithmic solution. This is a distributed machine learning model, which considers the collaborative input of involved clients to train the model under the supervision of a central server. The process maintains a decentralized storage model of the training data. Additionally, it is necessary to address the problem of missing data packets, relative to the training dataset.

Thus, Bi-Directional Long Short-Term Memory (Bi-LSTM) is regarded as a solution to address this issue, as it can determine the missing data values, and it is able to learn from previous data values. Consequently, it is possible to ameliorate the impact of missing data packets. The resulting algorithmic model is named ”Federated Learning Detection of Low-Rate Distributed Denial of Service” (FLD-LRDDoS) attacks. This is also the designation of the resulting integrated software system.

The proposed model brings the following contributions.

- The enhancement of the data usage efficiency is ensured by the design and implementation of a data preprocessing module.
- The main data processing module is based on the consideration of Bi-LSTM, which has the role to circumvent the effect of noisy data.
- Additionally, a main data processing node selection algorithm is designed, tested and implemented into the integrated FLD-LRDDoS software system. This ensures the highest possible accuracy for the classification process. Moreover, the data are stored using a decentralized model, and the computational time complexity is minimized.

The paper comprehensively addresses the very pertinent conceptual and practical problematic, which is approached in a fundamentally innovative manner, through the following structural

sections. Thus, a literature review of the most relevant similar contributions is conducted, which is followed by a comprehensive study of possible LRDDoS attack models. Furthermore, the architectural and implementation details of the proposed FLD-LRDDoS solution are provided. The integrated system was evaluated using the real-time corporate data traffic of a major Romanian Information Technology company, which administers multiple branches. The effectiveness of the proposed solution is assessed using several metrics, such as accuracy of classification process, precision, and recall. Additionally, the numerical evaluation is conducted using accuracy, cross-entropy loss, and average detection time. The last section concludes the paper.

LRDDoS attack patterns have been formally studied since 2003. They demonstrate that it is possible to maliciously use the retransmission timeout mechanism of TCP. Thus, the generation of sudden low-rate data flows may determine a rapid decrease of the overall TCP data throughput. The specification of proper LRDDoS attacks mitigation models relies on the existence of sufficiently populated datasets. Nevertheless, these are not readily available, which determines a particular aspect of the problematic that is approached in the surveyed literature.

Certain researchers try to enhance the utilization of public datasets using improved machine learning models and representation learning algorithms. Thus, the authors of paper [27] comparatively evaluated and practically assessed classic and deep learning models relative to public datasets. They asserted that deep learning models provide better performance compared to classic approaches, regarding the false alarm rate, scalability, and detection accuracy. Additionally, paper [28] evaluated an identification model, which considers the utilization of wavelet transforms and combined neural networks. These are considered in order to discern general data network traffic from LRDDoS patterns.

It is interesting to note that article [29] presented a comprehensive review concerning relevant security aspects, privacy models, and emerging defense technologies relative to unmanned aerial vehicles. This survey highlights the importance of properly designed and deployed data filtering policies to ensure the reliable

operation of the enrolled unmanned aerial vehicles. Moreover, paper [30] discussed on significant aspects regarding task offloading models in mobile edge computing environments. Although this approach may contribute to a relatively efficient utilization of the implied data links, it does not include the necessary algorithmic routines to properly detect and filter LRDDoS patterns.

Article [31] introduced a preclassification using locality-sensitive features extraction. This considers the features that are similar and groups them with a sufficiently high probability. Moreover, the paper proposes the usage of Convolutional Neural Networks (CNN), which are useful in order to process high-dimensional data and determine relevant features in preclassified smaller buckets. Let us note that high-dimensional data is featured by a significantly greater number of features relative to the number of observations. Considering that the network data, which interests an LRDDoS detection process, is defined by numerous technical features, and usually fewer observations, then it is immediately to note that the relevant LRDDoS model processes high-dimensional data. Nevertheless, the considered datasets are not sufficiently large, which essentially affects the performance of the proposed detection model. Therefore, certain researchers consider the generation of synthetic network and traffic data, in order to properly calibrate the training process. Thus, the authors of article [32] considered the NS3 tool [31] to generate a synthetic networked structure, which created the required datasets. Furthermore, article [33] described an emulated LDDoS approach that considers lightweight virtualization models. This may construct an emulated LDDoS scenario, which features 400 data routing hosts (nodes) hosted by the physical machine (server). Furthermore, article [34] presented SynGAN, a framework that generated adversarial network attacks through the consideration of Generative Adversarial Networks (GAN) that process real-world attack patterns.

It is important to note that article [35] described an efficient Automated Machine Learning (AutoML) approach, which selects optimal supervised classifiers that are considered to develop an enhanced ensemble learning strategy that includes a soft voting method to detect network intrusion attempts. This is intended to

enhance the accuracy of the detection process, and reduce to a minimum the rate of the false alarms. The evaluation that was conducted suggested that although the approach is proper for the detection of general intrusion patterns, it performs inefficiently in the case of low-rate data transfer patterns, which include the LRDDoS attacks. Additionally, the contribution that was reported in paper [36] addresses relevant security problems that were identified during the operation of enterprise cloud computing applications. The proposed solutions mostly pertain to general intrusion detection scenarios, which are less applicable to LRDDoS patterns that are difficult to detect.

Paper [23] described an approach for the generation of LDDoS data samples, which are based on Long Short-Term Memory and Conditional Generative Adversarial Networks (LSTM-CGAN). Thus, the described model identifies relevant time-domain features of LDDoS traffic patterns using LSTM structures. Moreover, the required attack patterns are generated using a Conditional Generative Adversarial Networks (CGAN) model. Additionally, paper [24] further extended the perspective over CGAN to a transformer-based model, which may be able to generate longer LDDoS attack patterns. These contributions propose relevant approaches for the generation of large synthetic datasets.

Nevertheless, relative to the reference or ground truth data, the appropriateness or validity of the generated attack patterns using CGAN networks should be assessed using supplementary processes. Consequently, considering that the data collected in a particular networked environment or datacenter is not sufficient, federated learning (FL) models may be used in order to aggregate, process and properly analyze the data that are collected from multiple networked environments or datacenters. Thus, paper [37] described a real-world synchronous FL model, which models data distributions that are not Independent and Identically Distributed (non-IID).

Moreover, paper [38] described a synchronous FL-related model concerning intrusion detection in IoT networks. Considering the reported experimental results, the private data are processed at a superior level of accuracy. It is interesting to note that paper [39] reported a comprehensive literature review concerning the detection

and proper processing of intrusion attempts in customized IoT devices networks. This comprehensive survey, which includes 167 references, interestingly suggests that the very significant real-world problematic of LRDDoS attacks is not properly approached.

Also, authors of paper [40] proposed an intrusion detection model, which considers FL and convolutional neural networks (CNN) to address the problem of accurate training using the limited labelled data relative to a particular generation mechanism. It is relevant to note that article [41] described a federated deep learning scheme, which is useful to detect threat patterns through the consideration of CNN structures and gated recurrent units (GRU). The authors reported experimental data considering a real-world dataset, which may demonstrate the proposed model's capability to detect several types of illegitimate data traffic patterns. Nevertheless, the reported values of the accuracy are not satisfactory, and the synchronous FL algorithmic models are computationally expensive. In essence, the paper demonstrates the suitability of LSTM networks for the efficient detection of long time-dependent LDDoS attack patterns.

It is relevant to note that paper [42] approached the DoS and DDoS attacks detection using a fuzzy intrusion detection system. The fuzzy algorithmic model allows for the identification of both DoS and DDoS data traffic patterns to occur, but the implied rates of detection accuracy, and also the efficiency of the related computational processes are not sufficient.

The authors of article [43] described a hybrid CNN and GRU approach, which is considered to determine relevant temporal and spatial features of LDDoS attack patterns. Moreover, article [44] reported an autoencoder-based anomaly detection model, which is used to determine time-based features (TAE) relative to several time windows. This approach is regarded as an enhancement of DDoS attack patterns detection. Also, paper [45] described a variational LSTM (VLSTM) machine learning model, which is considered to detect reconstruction anomalies concerning the representation of features. Thus, a neural network that specifies an encoder-decoder was specified, which is useful in order to learn the low-dimensional representation of features relative to high-dimensional raw data.

The described experimental process used public data, which proved that the reported VLSTM approach may address high dimensional issues, while essentially improving the accuracy, and the rate of false positives. The paper also discusses about the application of LSTM models to FL-based approaches. Although interesting, this contribution seems to be less flexible, computational efficient and accurate than the integrated software system that is proposed in this paper. Furthermore, article [46] described an intrusion detection system, which uses an algorithmic model that is based on federated learning aided long short-term memory (FL-LSTM). This may possibly address the issue of personal data privacy violation during the process of threat patterns detection. The conducted literature survey identified certain articles that report asynchronous FL methods.

Thus, article [47] described an approach that may be useful to detect computationally powerful network nodes considering an asynchronous model. Additionally, it is relevant to note that papers [48] and [49] propose interesting solutions regarding the detection of DDoS traffic in software defined networks.

The thorough literature review that was conducted demonstrates that the existing approaches are suitable, at most, to detect DDoS and LRDDoS attack patterns in rather low-throughput network infrastructures. The integrated software system, which the analyzed paper proposes, addresses all the existing algorithmic and functional shortcomings. It is also interesting to note that the thorough review of the scientific literature that has been published over the past three years reveals that the reported FLD-LRDDoS integrated system proposes unique algorithmic, architectural, and implementation solutions. Therefore, the evaluated FLD-LRDDoS system demonstrates that attack patterns can be efficiently and timely detected, even in high-throughput network infrastructures.

Considering an architectural perspective, the integrated solution relates to the following three software modules. Thus, the first module implements the data preprocessing stage, which conducts the relevant features extraction, relative to the processed dataset, and it also performs the analysis of the network traffic. It is relevant to note that this module efficiently selects the features that specifically pertain to

the LRDDoS traffic patterns. Moreover, the second software module implements the mediation functional features. These conduct a precise and efficient classification of the analyzed data traffic patterns, which effectively detect the LRDDoS data packets. Thus, an LRDDoS detection model is implemented and locally trained. The relevant algorithmic model uses a Bi-LSTM network, which is designed to learn the chronologically determined features.

The selection of the relevant features that are included in the detection process is conducted by an attention-based [50] neural network layer. The third software module represents the global aggregation operation, which is conducted using a federated learning model. This module also performs the important task of selecting the main data processing node. The selection is performed asynchronously through the continuous update of assigned importance scores. The following sub-sections present the relevant details regarding the three software modules. The following sections describe the functional logic of the three fundamental data processing stages.

The principal workload during the data preprocessing stage is represented by the creation of parameters dataset, which models a bidirectional data stream. This is provided as input to the LRDDoS data traffic patterns detection algorithm. Considering that the Bi-LSTM network accepts a constant number of input neurons, then it is immediate to note that the length of the input features is also constant. Consequently, the approach considers a classic networking model, a sliding windows mechanism, which is based on constant chronological steps. This approach effectively specifies and implements the generation of chronologically ordered and constant data samples, and it also determines an enrichment of the training dataset. The logical structure of the sliding windows mechanism is presented in **Figure 4**.

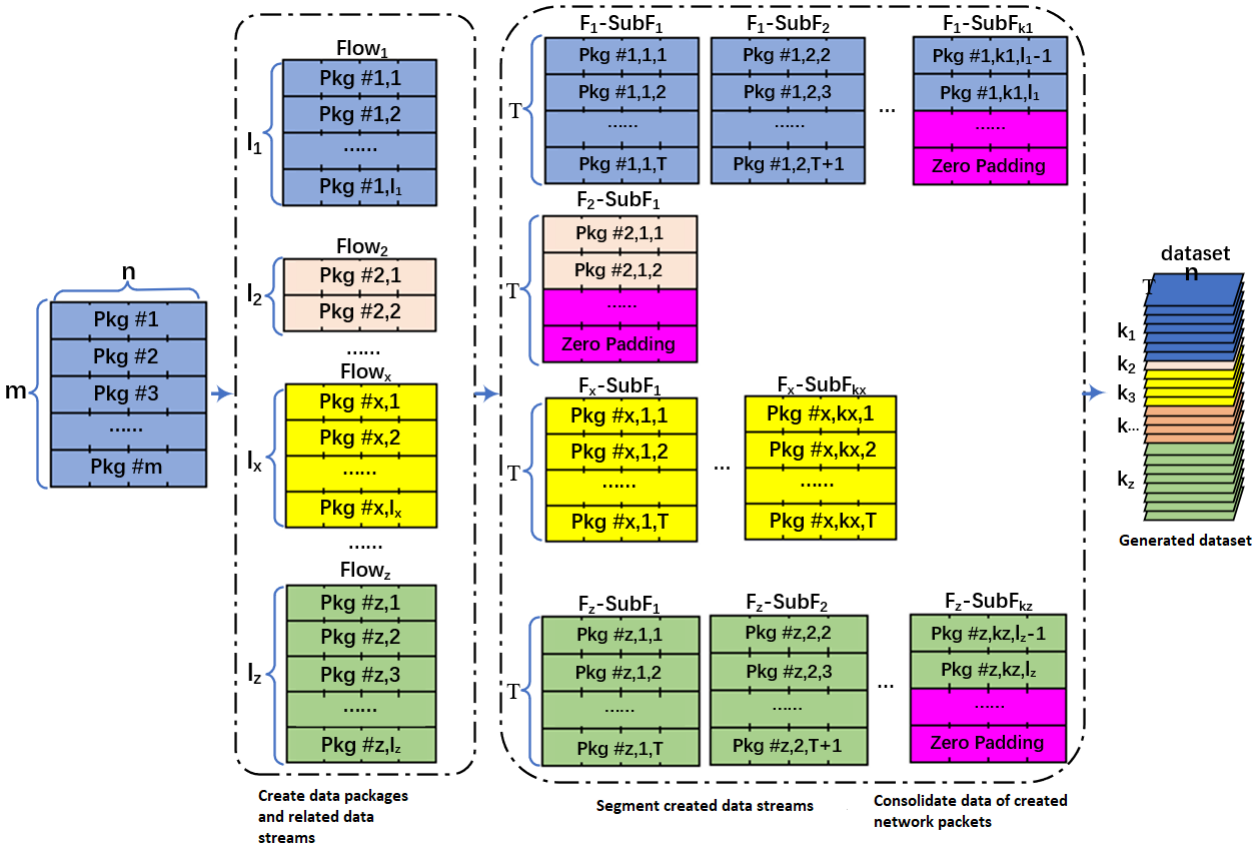

Figure 4. Logical structure of the sliding windows mechanism.

Thus, Figure 4 presents the logical chronological steps. First, the raw data are used to create basic data packages, and the related data streams. Furthermore, the generated data streams are properly segmented. Moreover, the data goes through a consolidation process, which improves the computational efficiency of the respective data processing routines. The process outputs the required training dataset. The following sections detail the presentation of the data preprocessing stage.

Initially, the relevant features are extracted and evaluated. We have experimentally determined that the LRDDoS attack patterns are defined by detectable data traffic burst models. This implies that the legitimate network data transfer presents a consistent and chronologically stable pattern, while LRDDoS patterns are featured by the mentioned data traffic bursts. Thus, let us divide the network data traffic considering 30 seconds intervals. Consequently, the IP data

packets features may be statistically aggregated (DPSF) considering the following formula:

$$DPSF = \Sigma NP_{\Delta t}.$$

Here, *NP* represents the number of processed data packets. Consequently, if multiple IP data traffic sources send LRDDoS attack patterns to a particular IP address, then the number of IP data packets increases relative to a chronological *Δt* pattern. Similarly, if an IP data source generates attack packets to many recipient ports of a destination host considering a chronological *Δt* pattern, then the number of affected port numbers also increases significantly. It is relevant to note that relative to the aggregated score *DPSF*, certain technical connection features are identified, which are presented in **Table 4**. Thus, *CT* represents the analyzed communication time length, which is an important parameter that is used in order to detect the LRDDoS data traffic patterns. Additionally, *LFDP* describes the type of the detected LRDDoS traffic pattern.

Table 4. Relevant technical connection features.

Feature number	Feature name	Description
1	DPSF	The statistical aggregation of the IP data packets features at 30 seconds intervals
2	CT	The analyzed communication time length
3	LFDP	The values range regarding the Length of First Data Packet

Furthermore, the intercepted data streams are sorted considering a chronological model. Thus, each time slot is determined by the preceding and succeeding ones. Therefore, the dataset that is used in order to train the LRDDoS attack patterns detection process essential relies on the chronological succession of data packages. This may be modeled through a bidimensional array, which contains related LRDDoS training data relative to each particular time slot, which contains the most relevant features.

Thus, the matrix that is described in the below expression suggests that each chronologically determined data stream is featured by the most relevant features. Additionally, *cts* represents the current time slot. It is immediately evident that the designed model allows for further features to be considered, if necessary. It is also relevant to note that the described matrix is able to accommodate part or all of the processed data streams. Thus, if all data streams and available features are modeled, then the array's dimension will be $s \times f$. Here, s represents the total number of data streams, while f determines the total amount of available features.

$$\begin{pmatrix} DPSF_0 & CT_0 & LFDP_0 & ... \\ DPSF_1 & CT_1 & LFDP_1 & ... \\ ..... & ..... & ..... & ..... \\ DPSF_{cts} & CT_{cts} & LFDP_{cts} & ... \end{pmatrix}$$

Furthermore, the mechanism of sliding windows is applied, considering a constant sliding windows step size *SWS*. This approach is considered in order to partition the chronologically determined LRDDoS data streams, which are part of the overall processed data traffic. If the number of data messages in the overall stream is lower than *SWS*, then the structure is filled with values of zero. This approach enhances the LSTM-related learning process, without inducing a noticeable computational overhead. These conventional data can be assimilated to the idea of noise, which may be processed, if required, using specific algorithms. The efficiency of the computational process is further enhanced through the aggregation of the determined features into the feature sequence that is described in the following mathematical expression.

$$FS = \Sigma_{k=1}^{n} k_i \times f \times SWS.$$

Here, f is the number of available features, while *SWS* represents the sliding windows step size. Furthermore, k_i is represented by the following mathematical expression.

$$k_i = \begin{cases} l_i - SWS + 1, l_i \geq SWS \\ 1, l_i < SWS \end{cases}$$

It is also relevant to note that in this mathematical expression, l_i represents a numerical value, which is essential in order to efficiently detect LRDDoS attack patterns relative to the respective labelled features. Consequently, the number of the available data streams *s* can be modeled according to the following mathematical expression.

$$s = \Sigma_{i=1}^{f} l_i.$$

The complete processing of the described conversion operations implies that the label of the raw (original) data stream can be affixed to the related block of features. This effectively generates multiple smaller arrays (matrices), which are used in order to properly train the model that automatically detects the LRDDoS traffic patterns. Additionally, Figure 4 suggests that the packages-related features are transformed into the required chronological features. Therefore, the proposed algorithmic model both learns the features of the currently processed data package, and also the features of the preceding (*SWS-1*), and succeeding (*SWS+1*) data packages. This effectively implements a bidirectional learning process, which significantly enhances the proposed model's detection accuracy, and also its computational efficiency.

Furthermore, the data processing pipeline continues with the second stage, which relates to the actual detection of the LRDDoS data packets. Essentially, the acquired LRDDoS attack data packets are accompanied by missing packets in the respective data streams. This negatively impacts the accuracy of the prediction process. Therefore, let us recall that the proposed model considers a bidirectional LSTM network, which is capable to accurately predict the values of the missing data packets. This is determined by the proposed algorithm's ability to consider both the informational states of preceding and succeeding data packets. Additionally, the proposed approach is

designed to reorder the selected features based on their relative functional importance through the utilization of a parameters' weights system. The efficiency of the proposed model is further enhanced by the consideration of an attention mechanism [51], which creates an essentially hybrid bidirectional LSTM (Bi-LSTM) network. This improves the learning process of the chronological features relative to the preceding and succeeding packets, which are part of the bidirectional data streams that are processed. This also induces the amelioration of essential performance metrics' values, such as recall [52] and precision [53].

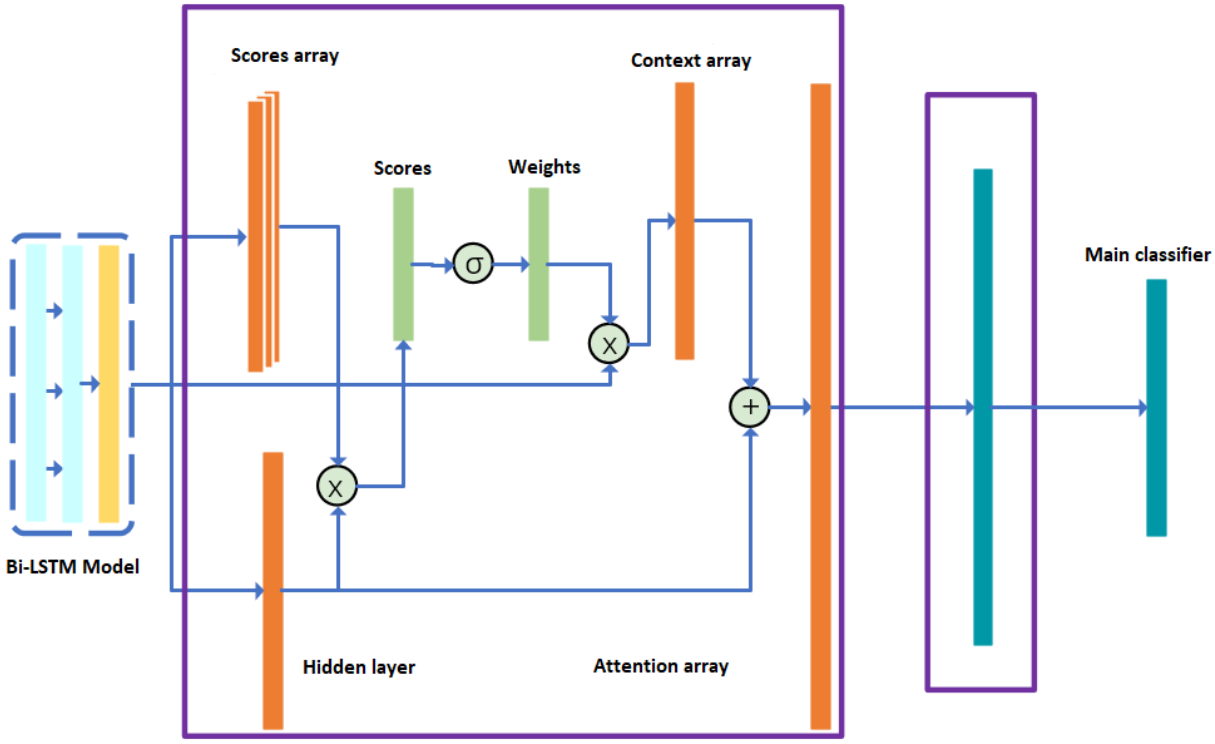

Figure 5. The logical data flow through the bidirectional LSTM layer.

The logical informational structure of the proposed Bi-LSTM layer is presented in **Figure 5**. Thus, the sequenced LRDDoS data packets are provided as input to the Bi-LSTM network. These data are used to discern valid data from noise relative to the forward and backward hidden layers. The attention mechanism is used to aggregate relevant features using a weights redistribution mechanism. Consequently, two fully connected network layers are considered to perform the actual classification of the processed data packets. The issue of overfitting [54]

is addressed using the L2-normalization function [55]. The logical architecture of the neural network layers is presented in **Table 5**. Thus, the reported detection model considers 229,888 parameters. It is relevant to note that the layers that pertain to the attention mechanism are not displayed, as they don't contain any parameters.

Table 5. Logical architecture of the neural network layers.

Neural network layer information	**Output configuration**	**Number of parameters**
Attention vector (Dense)	(none, 128)	65,536
dense-1 (Dense)	(none, 1)	512
dense (Dense)	(none, 128)	dense (Dense) (none, 128) 32,768
Bidirectional LSTM	(none, 25, 128)	65,536
attention-score-vector	(none, 25, 128)	65,536

The bidirectional LSTM network is structured considering two LSTM layers, which may be regarded as placed one over the other. Each LSTM layer is made of 128 neurons. Thus, one of the layers is concerned with the learning process through the forward hidden states, while the other layer relates to the backward hidden states. The two layers produce separate outputs, which are aggregated and sent over to the succeeding layer. This effectively ensures that the data processing has complete chronological context information, which optimizes the accuracy of the predictions. The following expressions describe the operations that the cells of the bidirectional LSTM network support.

$$fw\_output_t = \tanh(weight_i input_t + weight_2 fw_{t-1})$$

$$bw\_output_t = \tanh(weight_3 input_t + weight_5 bw_{t-1})$$

$$output\_gate_t = \tanh(weight_4 fw_t + weight_6 bw_t)$$

Here, $input_t$ represents the input that is received at time *t*, while *w* determines the weights that are assigned to the cells of the LSTM network. Additionally, fw_output_t and bw_output_t represent the forward and backward output, respectively, and the output gate $output\_gate_t$ aggregates output data related to the bidirectional data flows. Consequently, the bidirectional LSTM network is able to successfully record a significant amount of chronological state information. This further improves the model's ability to recover lost data packets.

In essence, the array of selected features is provided as input to the bidirectional LSTM network. Furthermore, the L2-normalization function is applied, together with the activation function [56] *tanh*, which represents a hyperbolic tangent. This approach effectively normalizes the input data and helps prevent the phenomenon of overfitting. Nevertheless, it was experimentally observed that not all selected features are relevant for the detection of the LRDDoS attack patterns. Therefore, the integrated model that this paper proposes includes an attention mechanism, which is described in the next subsection.

In general, an attention mechanism is modeled through a mathematical attention function. Such a function represents an association (mapping) of a query and a set of key-value pairs to a certain output. Consequently, the reported model uses this mechanism in order to recalibrate the considered weights, with the goal to select the most significant features during the LRDDoS data traffic detection process. The attention-related algorithmic routines are considered in order to determine the features that are particularly important for the detection process. Thus, the attention algorithm computes, for each selected feature, a value that is related to an array of features-related scores. Furthermore, the calculated scores are provided as inputs to a softmax function [57], and the scores are consequently transformed into proper weights. The basic mechanism is presented through the following mathematical expressions.

$$r_t = \tanh(weight_d fw\_output_t + offset_d)$$

$$nw_t = \frac{\exp(r_t s_d)}{\Sigma_t \exp(r_t s_d)}$$

Here, r_t represents the information that is stored in the neural network's hidden layer, s_d refers to the similarity of the features arrays, and nw_t computes the normalized weight values. The experimental evaluation that was conducted proves that the inclusion of this attention mechanism ensures that the LRDDoS traffic patterns are detected accurately and in a computationally efficient manner.

The neural network also includes two fully connected layers, which implement two functional features. First, one of the layers helps to reduce the size of the features, and the other one supports the traffic analysis module, which effectively determines whether the intercepted data packets are LRDDoS patterns, or represent legitimate data transfer sessions. These additional neural network layers consider the utilization of softmax and sigmoid [58] activation functions. The phenomenon of overfitting is efficiently prevented through the inclusion of a dropout mechanism [59], which effectively transforms certain values into zeros. The considered probability (dropout rate), which was experimentally calibrated, is 0.47. Additionally, the binary cross-entropy loss function [60] is considered to compute the data loss, which is relative to the validation dataset and, also, to the training dataset.

It is relevant to note that the computed weights are further adjusted through the usage of a Stochastic Gradient Descent (SGD) [61] optimizer, and also by considering the mechanism of backpropagation [62]. This essentially propagates the overall data loss back into the neural network, which offers the possibility to measure each node's responsibility during this process. Consequently, the values of the weights are updated, so that the nodes featured by greater error rates are assigned lower weight values. This effectively ameliorates the phenomenon of data loss. These assertions are also confirmed by the outcomes of the experimental assessment process.

The third stage of the data processing flow relates to the global data aggregation operation. Thus, it has been empirically proven that federated learning is able to support the implementation of reliable data privacy mechanisms. Nevertheless, regular federated learning solutions are negatively influenced by improper selection of features, parameters, or even by incorrect computation of implied values. This is particularly true considering the data aggregation operations. Therefore, the proposed integrated system is able to reduce the side effects of improper data aggregation operations, and it also enhances the overall computational stability of the proposed solution. Thus, the described algorithmic model and related implemented integrated system proposes two essential improvements. The data processing considers an efficient and precise selection of the main data processing node, which is tasked with the actual data aggregation. Furthermore, the federated learning model is considered to conduct the actual correction of the assigned weights' values. This determines an efficient and precise data aggregation operation.

Contextually, the first aspect to be approached is represented by the selection of the main computing node. Thus, federated learning is considered in various real-world use case scenarios, which require a certain kind of data aggregation between geographically or otherwise dispersed systems, such as cloud computing datacenters. It is usually difficult to precisely select the main data processing node. Therefore, the proposed model selects the main data processing node considering the size of the IP addresses pool, and also the age of the available training data, which are stored on that particular node. More precisely, the node with the most recent training data is favored during the selection process.

The main node selection algorithm considers, as input, the triplet (*size_of_IP_pool*, *current_data_size*, *updated_data_size*). Here, *size_of_IP_pool* represents the dimension of the available IP addresses pool, the *current_data_size* is the number of currently available training data items, while *updated_data_size* designates the up to date number of training data items. The main node selection process begins with each of the data processing nodes updating its local parameter

updated_data_size, and electing itself as main node. Furthermore, the vote is sent to all the other available nodes through a broadcast message, which consequently updates the value of *current_data_size*, relative to the sender node. The required selection decision is effected according to the following formula.

$$node\_election\_ratio = 1 - \frac{UDS_{remote}}{UDS_{remote} + UDS_{local}}$$

Here, UDS_{remote} represents the updated data size of the remote data processing node, while UDS_{local} is the *updated_data_size* of the local data processing node. Thus, each node compares the global value of *node_selection_threshold* with the value of *node_election_ratio*.

The value of *node_selection_threshold* was calibrated through successive experimental attempts, in order to optimize the accuracy of the LRDDoS traffic detection process. If the *node_election_ratio* is greater or equal than *node_selection_threshold*, then the local node selects itself. Otherwise, the remote data processing node is selected. This algorithm is able to select the main data processing node after only a few iterations. This substantially enhances the computational and time efficiency of the process, relative to other relatively similar contributions. The logical structure of the algorithm is presented in the following pseudocode.

Thus, the completion of the algorithm also allows the selected main data node to know the total number of available data processing nodes, through the addition of the values that are stored in the parameter *current_data_size*. Moreover, if the *current_data_size* of the selected main data processing node is sufficient, then the node may be able to separate its local dataset into testing and training subsets. The inclusion of this feature into the algorithmic model allows the selected node to both process the data, and it also contributes to the update of the parameters' values. This enhances the computational efficiency of the implemented integrated LRDDoS detection system, considering the actual generation of the data traffic detection model.

Input: $(IP\_pool, current\_data\_size, updated\_data\_size)$
Output: The selected main node
Global Variable: $node\_selection\_threshold$

procedure ReceiveVote()
Compute **node_election_ratio** using formula 11.
if($node\_election\_ratio > node\_selection\_threshold$) **then**
    Store $(IP\_pool, current\_data\_size, updated\_data\_size)$
else
if($IP\_pool < remote\_IP$) **then**
    Store $(IP\_pool, current\_data\_size, updated\_data\_size)$
endif
Count $number\_of\_stored\_votes$
if $number\_of\_stored\_votes > 0.5$ **then**
    Select as main data processing node.
endif

procedure SendVote()
if vote data not existing **then**
    Select itself as main node.
else
    Send the stored votes data.
endif

Moreover, another important algorithmic routine implements the weights correction mechanism. The federated learning process occurs differently across the available computing nodes, considering the variable configuration of the local training datasets, and also the heterogeneous computational configuration. Consequently, the weights of the global parameters, particularly relative to the main data processing node, are not properly updated. Therefore, this unequal update of the parameters, which is specific to certain types of federated learning models, should be properly addressed.

The analyzed paper proposes a simple, but effective solution for this problem. Let us recall that the number of data items is precisely computed and recorded during the main node selection process. Thus, the normalization of the parameters' weights is described by the following formulae.

$$NW_i = \frac{Dataset_i}{Entire\_Dataset}$$

$$Entire\_Dataset = \sum_{i=0}^{N} Dataset_i$$

Thus, during the main node selection process, a scaled down test dataset is available for the main node, and the normalized weights are specified. More precisely, as soon as the main node gets the updated parameters' values, they are stored in the local space of the main node, and the accuracy of the LRDDoS detection process is updated relative to the specific weight values. This approach introduces the explicit specification of the individual computing nodes' contribution to the aggregated result, and it consequently addresses the issue of inconsistency induced by the considered federated learning model. Moreover, the obtained weights are further processed using a softmax model [57], and the normalized weights values NW_i are computed.

It is relevant to note that the integrated algorithmic model considers an additional optimization layer that is applied to the global parameters, after weights are normalized. This further enhancement is implemented considering a customized version of the approach that is reported in the scientific literature as DeFTA [63]. The main data processing node propagates the updated values to all the available nodes, and the new values are considered during the next iteration. This additional data processing layer completely solves the empirically observed data imbalances, and also the relatively lower computational speed during the training process. Moreover, the results of the experimental evaluation process demonstrate a smooth numerical variation of the cross-entropy loss. This effectively prevents the phenomenon of overfitting to occur, following the inclusion of the weights correction mechanism into the integrated federated learning model. This effectively distinguishes the proposed approach from certain similar solutions, which have been reviewed during the preliminary phase of the research process, and then properly included into the literature review discussion.

Additionally, it is important to note that the proposed federated learning model uses a cache memory layer. This stores previously detected LRDDoS attack patterns, and the related parametric data. We have determined, during the experimental assessment process, that this strategy significantly improves the computational and time efficiency of the LRDDoS attacks detection process. This generates an average classification time of 0.79 seconds relative to the incoming data packets for all detected LRDDoS data traffic types. This effectively ensures that the system processes both legitimate and LRDDoS data traffic in a real-time manner or, at worst for some data packets, near real-time manner. Nevertheless, the results of the experimental process suggest that most of the packets are fully processed in less than a second. To the best of our knowledge, the integrated FLD-LRDDoS system is one of the very few approaches that includes the necessary data processing optimizations, which ensure its ability to consistently process the received data packets in less than a second. Moreover, there are even less approaches that are reported in the surveyed scientific literature, which use a cache memory mechanism to speed up the overall computational behaviour of the system. The experimental assessment section includes a separate analysis of certain performance aspects, which include the effect of the cache memory on the data packets processing speed.

The experimental assessment considers 300 million data samples, which have been intercepted in the data network core of a major Romanian Information Technology company. Thus, 150 million data samples relate to LRDDoS traffic patterns, while the rest of 150 million data samples represent legitimate data traffic. It is relevant to note that this splitting model also contributes to the prevention of data imbalance. Additionally, 15% of the original data samples are transformed into random values, which simulates the presence of noise.

The experimental evaluation also considered the federated learning routines. For this purpose, ten computing nodes were created, which administered local datasets that stored the following number of data items, respectively: 30 million, 40 million, 60 million, 80 million, 90

million, 110 million, 150 million, 210 million, 260 million, and 280 million. Considering the main node selection algorithm, which this paper proposes, the data processing node that managed 280 million data items was chosen. The respective local dataset was partitioned into 170 million data items and 110 million data items, which represented the training dataset and testing dataset, respectively. The data processing nodes were setup using the Vagrant virtualization platform [64], which offered the possibility to efficiently create the necessary computing and storage resources. Each node accessed 64 logical cores of an AMD EPYC 7702 processor, 256 GB of random access memory (RAM), and the necessary storage resources.

Additionally, certain details regarding the calibration of the machine learning model's hyperparameters are relevant. Thus, the proper evaluation and validation of the bidirectional LSTM network model is conducted. Thus, this paper compares the loss and accuracy values between the following models: LSTM, Bi-LSTM, Bi-LSTM with dropout, and the customized version of Bi-LSTM, which this paper proposes. The evaluation is conducted over 500 epochs. The proposed model produces the most precise results, with an accuracy score of, at most, 98.79%, while the optimal cross-entropy loss [65] is less than 0.076. The results that are obtained after epoch 500 are considered as the reference level of performance, which is consequently used to validate the test dataset.

It is relevant to note that during the initial design process of the federated learning model, we have assessed both cross entropy and divergence-based metrics. It was determined that cross entropy better evaluates the difference between predicted probabilities, and also true probabilities in a more efficient computational manner. Overall, cross entropy determined a superior computational behaviour of the classification process.

The values of accuracy and cross-entropy loss have also been computed relative to the reference models, LSTM, Bi-LSTM, and Bi-LSTM with dropout. Thus, the following values are obtained for accuracy and cross-entropy loss, respectively: (96.29%, 0.156) (LSTM), (96.55%, 0.142) (Bi-LSTM), and (96.82%, 0.128) (Bi-LSTM with dropout). It is immediate to note that the customized proposed model performs

better than all the algorithmic approaches, which were reviewed during the initial stages of the reported research process.

The basic configuration of the model is followed by the tuning of the hyperparameters, which is accomplished through the successive evaluation of the obtained values. Thus, the greedy optimization method grid search [66] is considered to obtain the optimal performance. The calibration process established that a dropout rate of 0.315 determines the optimal values for the accuracy and loss. It is relevant to note that the optimization of the federated learning model considers three fundamental hyperparameters. These are ”Dropout rate”, ”Neuron Units of bidirectional LSTM”, and ”Dominant layers”. The related reference values are 0.29, 128, [Bidirectional LSTM; Attention; Dropout], respectively.

Consequently, the utilization of the optimized dropout rate is followed by the application of grid search, which calibrates the number of included neurons that is determined to be 128. Thus, this optimized model is able to properly learn the representation features, and it generates an accuracy of 99.08%, which is superior to most of the reviewed relatively similar approaches. Let us recall that the proposed model considers a Stochastic Gradient Descent optimizer, which avoids the implementation of a learning rate optimization algorithm. This saves valuable computational resources, and it also contributes to the simplicity of the proposed model.

Let us recall that the experimental assessment considers 300 million data samples, which have been intercepted in the data network core of a major Romanian Information Technology company. The effectiveness of the proposed approach is formally assessed using two metrics, the accuracy of the LRDDoS patterns detection, and the cross-entropy loss [65]. In essence, the cross-entropy loss measures how much the predicted value diverges from the correctly labelled value. The experimental process considers several subsets of the overall dataset, which store the following number of data items: 30 million, 50 million, 80 million, 110 million, 140 million, 180 million, 220 million, 250 million, 270 million, and 300 million. Furthermore, the analysis evaluates the advantages, which the consideration of the federated learning model brings. Thus, the ten datasets are considered both

considering a localized node-based approach, and also a distributed federated learning model, which is described in this paper.

The evaluation is conducted over 500 epochs. The proposed model produces the most precise results, with an accuracy score of, at most, 98.79%, while the optimal cross-entropy loss is 0.076. The results that are obtained after epoch 500 are considered as the reference level of performance, which the experimental evaluation process considers. The data processing nodes, which are setup using the Vagrant virtualization platform, feature identical configuration parameters, more precisely, 64 logical cores of an AMD EPYC 7702 processor, and 256 GB of random access memory. This ensures that the assessment of the local node-based data processing generates identical results, regardless of the node that is used. It also determines a predictable improvement of the data processing results, as long as new data nodes are added to the federated learning infrastructure.

Table 6. Results of the local node-based data processing experiment.

Dataset size	**Value of accuracy**	**Value of cross-entropy loss**
30 million	81.79%	0.274
50 million	82.93%	0.269
80 million	84.09%	0.261
110 million	85.99%	0.248
140 million	87.96%	0.236
180 million	88.79%	0.223
220 million	89.89%	0.204
250 million	90.49%	0.178
270 million	91.39%	0.166
300 million	92.29%	0.153

Thus, **Table 6** presents the results of the local node-based data processing experiment. The evolution of the presented metrics values demonstrates that it is important to process an as comprehensive as possible subset relative to the entire available dataset, in order to obtain accurate results. Nevertheless, the selection of a particular data processing node is improper considering that the data processing time is significant, and the inability to use the federated learning computation model also affects the overall accuracy of the data

processing results. Thus, the maximum value of the accuracy is 92.29%, which was obtained in the case when all the 300 million data items are processed. Furthermore, the value of cross-entropy loss belongs to the discrete domain {0.153, ..., 0.274}.

Table 7. Results of the local node-based data processing experiment.

Dataset size	Value of accuracy	Value of cross-entropy loss
30 million	89.74%	0.212
50 million	90.92%	0.201
80 million	91.79%	0.189
110 million	92.96%	0.178
140 million	93.88%	0.161
180 million	94.81%	0.148
220 million	95.87%	0.129
250 million	96.75%	0.105
270 million	97.88%	0.093
300 million	98.79%	0.076

Furthermore, the results that were determined by the utilization of the fully featured integrated data processing system, which is based on a customized federated learning model, are presented in **Table 7**. Thus, it can be observed that the federated learning-based approach ensures an acceptable level of the accuracy, even in the case when only a tenth of the entire dataset is processed, and the maximum value of the accuracy is 98.79%, with a value of cross-entropy loss of 0.076. Furthermore, the value of cross-entropy loss belongs to the discrete domain {0.076, ..., 0.212}. This effectively ensures that the LRDDoS data traffic patterns are accurately detected relative to nearly all of the intercepted data items.

Considering the empirical evaluation of numerous relatively similar approaches, which has been conducted during the initial stages of the research process that this paper reports, it can be asserted that this integrated LRDDoS data traffic detection system behaves better than most of the existing similar approaches. Additionally, to the best of our knowledge, this is the only reviewed experimental process, which uses a very large experimental dataset that stores real-world

data. This represents another merit of the contribution that this paper reports.

In conclusion to the presentation of this paper, let us recall that the proposed model considers a federated learning system, which aggregates the computational resources of multiple data processing nodes. The described integrated system considers three software modules. Thus, the first module implements the data preprocessing stage, which conducts the relevant features extraction, relative to the processed dataset, and it also performs the analysis of the network traffic. It is relevant to note that this module efficiently selects the features that specifically pertain to the LRDDoS traffic patterns, and it also tunes the federated learning model's hyperparameters. Moreover, the second software module implements the mediation functional features. These conduct a precise and efficient classification of the analyzed data traffic patterns, which effectively detect the LRDDoS data packets. Thus, an LRDDoS detection model is implemented and locally trained. The relevant algorithmic model uses a Bi-LSTM network, which is designed to learn the chronologically determined features. The selection of the relevant features that are included in the detection process is conducted by an attention-based neural network layer. The third software module represents the global aggregation operation, which is conducted using a federated learning model. This module also performs the important task of selecting the main data processing node. The selection is performed asynchronously through the continuous update of assigned importance scores.

The experimental assessment comprises several perspectives. Thus, it describes the considered dataset and the types of LRDDoS attack patterns that it contains. Furthermore, it discusses about the calibration of hyperparameters, and also about the federated learning data processing modules. The proposed integrated system's performance is comparatively assessed relative to significant LRDDoS detection approaches.

Additionally, the proposed integrated model's performance is numerically assessed considering the accuracy of the LRDDoS patterns detection, and the cross-entropy loss. Moreover, the experimental process included an extended performance assessment phase, which is

also presented. This considers LRDDoS attack patterns that belong to the three fundamental categories of attacks: Shrew and CICADAS from the category of QoS attacks, Slowloris and SlowDrop from the class of Slow DoS attacks, and LoRDAS as a significant exponent of the Service queue attacks. The extended performance evaluation also considers an additional performance metric, the average detection time.

The experimental results demonstrate that the proposed integrated system is able to efficiently process very large amounts of data, and the values of the considered performance metrics prove that the LRDDoS attack patterns are accurately detected. It is also relevant to note that, considering the conducted literature review, this is one of the very few approaches, which is capable to process very large amounts of data, with an accuracy rate of almost 99%. This essentially individualizes the proposed integrated LRDDoS detection model.

Chapter 3. Software applications related to computer networks

The contributions that relate to this category are described in numerous peer reviewed papers. Thus, some of the selected contributions are reported in papers [13], [14], [15], [16], [17], [18], [19], and [20]. The scientific case study for this chapter relates to the presentation of the contribution that was reported in paper [15], which relates to a real-time intrusion detection and prevention system for 5G and beyond software-defined networks.

The philosophy of the IoT world is becoming important for a projected, always-connected world. The 5G networks will significantly improve the value of 4G networks in the day-to-day world, making them fundamental to the next-generation IoT device networks. This article presents the current advances in the improvement of the standards, which simulate 5G networks. This article evaluates the experience that the authors gained when implementing the 5G network services of a major telecommunications operator, illustrates the experience gained in context by analyzing relevant peer-to-peer work and used technologies, and outlines the relevant research areas and challenges that are likely to affect the design and implementation of

large 5G data networks. This paper presents a machine learning-based real-time intrusion detection system, and also the corresponding intrusion prevention system. The approach considers a convolutional neural network (CNN) to train the model. The system was evaluated in the context of the 5G data network. The smart intrusion detection system (IDS) takes the creation of software-defined networks into account. It uses models based on artificial intelligence. The system is capable to reveal not previously detected intrusions using software components based on machine learning, using the specified convolutional neural network. The intrusion prevention system (IPS) blocks the malicious traffic. This system was evaluated, and the results confirmed that it provides higher efficiencies compared to less overhead-like approaches, allowing for real-time deployment in 5G networks. The presented system can be used for symmetric and asymmetric communication scenarios.

Fifth-generation cellular networks forced the implementation of 5G and beyond networks, which offer capacity expansion strategies to handle great connectivity issues and can offer very high throughput and low-latency. Thus, 5G and beyond technology uses IoT, AI/ML, and blockchain, and its goal is to establish secure and reliable UAV networks. Therefore, the relevant research efforts must be conducted to ensure security of 5G and beyond networks [67]. It can be very relevant to the integration of protected mechanisms, which use machine learning (ML) and artificial intelligence (AI) techniques. Scientists apply ML algorithms to the development of IDS systems in order to identify and classify malicious traffic [68].

The intrusion detection system, which is presented in this paper, addresses the stringent design, implementation, and deployment aspects of high-bandwidth 5G network cores. Thus, traditionally, the data traffic is filtered mostly using semiautomatic approaches. These usually generate low levels of data patterns detection accuracy, and they are not able to adapt and detect unknown data patterns. It is important to mention that the integrated intrusion detection system, which is presented in this paper, is one of the very few relevant systems that are proven to detect known and unknown threat patterns in a large 5G network data core, with high accuracy and without

interfering with the low-latency levels of the implied data network, as they are perceived by the end users.

It is relevant to note that 5G networks create broad-bandwidth channels. However, the increased efficiency that these new networks create is largely due to the number of intelligent devices supported and the related applications. Broad-bandwidth data links imply that intelligent application deployments require data links capable of a minimum of 25 Mbps, and are designed to sustain meaningful augmented reality (AR) and virtual reality (VR) use case scenarios [69]. Large scale and structurally flexible networks are defined by the network function virtualization (NFV) mechanism to create the required networked structures. The 5G IoT low-latency data networks are designed to sustain intelligent applications that need to transmit and receive data in real-time, and use communication channels with delays of no more than five milliseconds [70]. Safety and fault tolerance are determined by the presence of significantly fewer base stations in the 5G network, so handoffs must be done while maintaining optimal coverage of the network. Data privacy and protection relate to the applications that work with sensitive data, such as patient personal information, and mechanisms to prevent any unauthorized access attempts are required. Battery life is central to 5G data networks. Therefore, energy efficiency must be taken into account. The connectivity of 5G data networks must provide simultaneous, stable access to a huge number of devices, which means making the right design and implementation decisions. Additionally, mobility logically supplements the need to create the right environment for the development of many devices that require reliable mobile data links. It must be noted that although smart devices that need to work on 5G networks must handle huge amounts of data, they do not, in most cases, have sufficient hardware resources to process the given data. Therefore, in most cases, data processing is transferred to systems in the cloud that extract useful information from unprocessed data by considering data analysis techniques [71][72].

The contribution that is presented in this paper relates to the following perspectives. It presents the current advances in the improvement of standards that simulate 5G networks through

virtualized infrastructures. This determines a significant improvement in the telecommunications operators' administrative and infrastructure maintenance costs. This paper presents the experience that the authors gained when deploying the described system on the core infrastructure of a major telecommunications operator, and it outlines the relevant research areas and challenges that are likely to affect the efficient design and implementation of large 5G data networks, which use secure and economically efficient virtualized infrastructures. Additionally, the performance evaluation of the system, which considered a comprehensive sample of real networking data, demonstrates that the system is capable of detecting unknown and existing malicious data traffic patterns in a timely manner with a high level of accuracy. It is relevant to note that, to the best of our knowledge, this is one of the very few machine learning-based intrusion detection systems that is compatible with the proper and timely detection of malicious data traffic patterns in large broadband 5G data networks.

One of the fundamental goals of this contribution is to determine the best architectural model for designing 5G data transmission networks. However, any architectural design must take into account two points of view. The data perspective deals with real-time data analysis that uses software-based frontend data paths, while the management perspective deals with the suitable administration of the network components, and the associated services that they define. It must be mentioned that the structure of a 5G data transmission network must take into account considerable technical requirements, such as scalability and the ability to virtualize network functions, when implementing network resources and providing necessary capabilities to virtualize network functions. Therefore, comprehensive functional requirements must be accessible to support the required effective network management processes. This should include the effective setting of guidelines under which mobile devices will behave optimally, defining a policy to control access to network resources, and the ability to virtualize given physical network resources.

The virtualized wireless network function (VWNF), is the main function in the design and implementation of 5G data networks. It is

effectively used to design the 5G network's core (5GC). The process is able to logically define a self-sufficient 5G data network using network function virtualization (NFV). The basic technology of 5G is visualized in **Figure 6**.

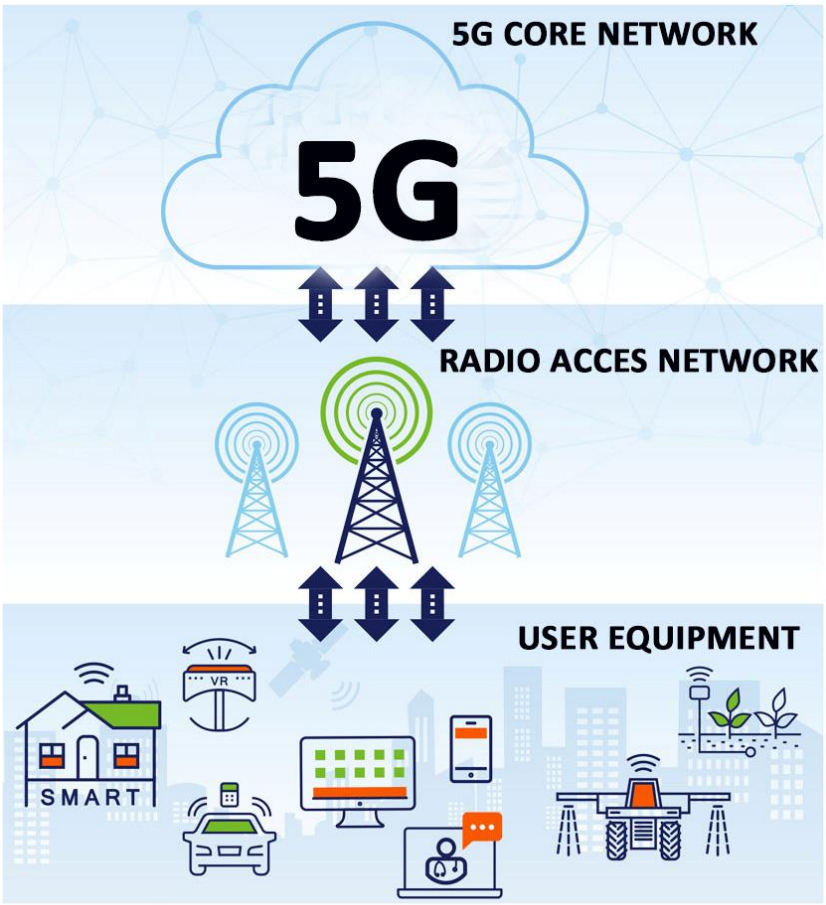

Figure 6. Functional logic of reference technologies.

It is worth noting that the mentioned process is important from both a theoretical and a research point of view. In addition, it allows the deployment of dedicated 5G networks in certain infrastructures, such as telecommunications or cloud providers, which provide various network services. We effectively worked with this system to realize particular network services in the 5G data network of the target telecommunications service provider. Thus, during implementation, we noticed that the virtualized network environment has the necessary logical plasticity and scalability, which allowed us to effectively develop an intrusion detection system in a real-time manner. The logical structure of the related network virtualization model is visualized in

Figure 7. In fact, we have determined that the considered virtualization engine provides the ability to properly handle data streams passing through the 5G network to detect potential or known threat patterns.

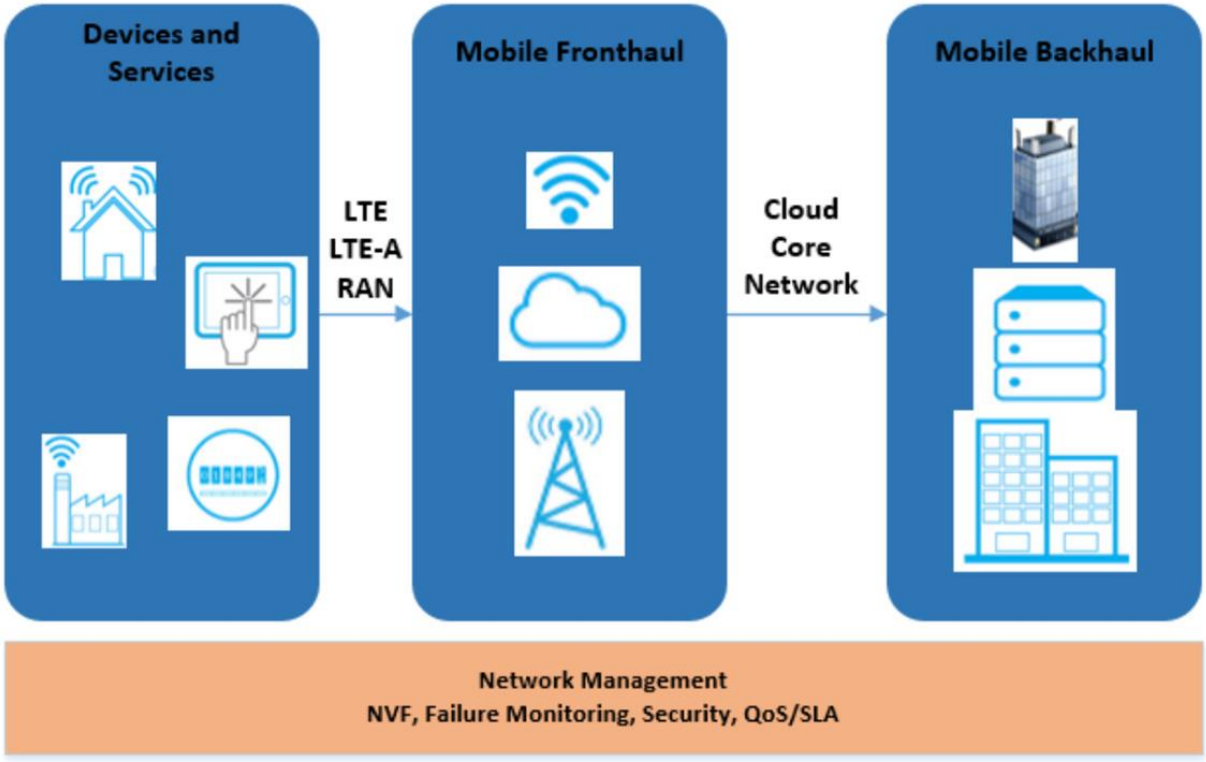

Figure 7. Logical partition of a virtualized network.

The experimental evaluation process that was conducted demonstrated that this system is appropriate for the correct formation of the needed dedicated virtual data network, which by itself confirms the conclusions outlined in paper [73].

Relative to Figure 7, let us particularly note the mobile fronthaul and the mobile backhaul components. In its simplest form, the backhaul connects the mobile network to the wired network by backhauling traffic from geographically dispersed cell sites to mobile switching telephone offices (MSTO). These links, which interconnect macro cell sites (sites housing the large mobile towers) to MSTO compounds through the utilization of 1 Gbps+ physical data tranmission interfaces. The proposed logical architecture of the fronthaul and backhaul components is described in **Figure 8**.

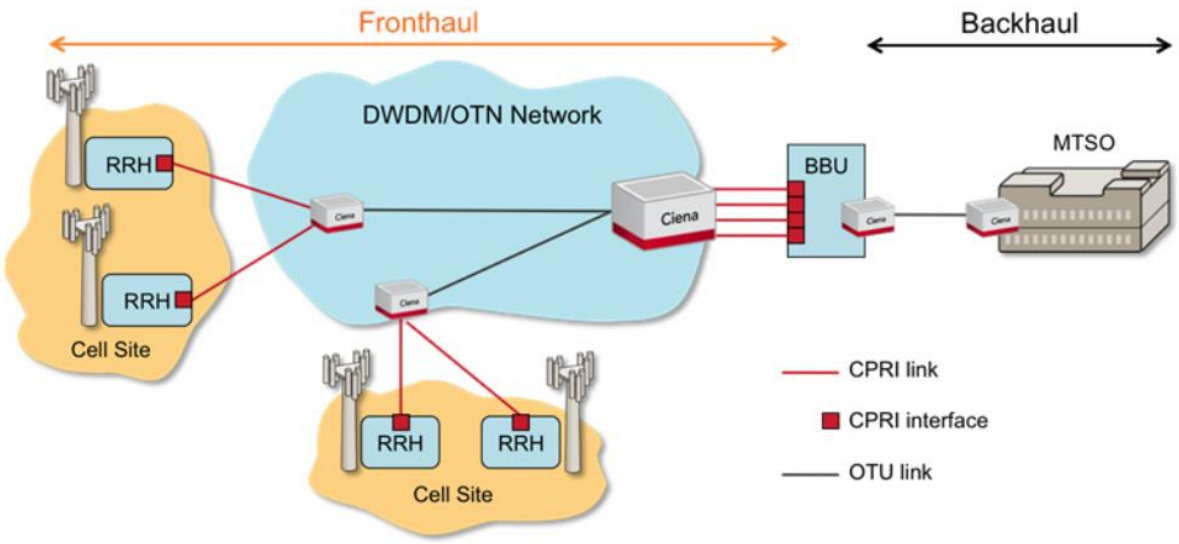

Figure 8. Logical architecture of the fronthaul and backhaul components.

We have also determined that the logical specification of the 5G networks optimizes the allocation and usage of the radio resources, as we were able to define logical sub-networks that are separately analyzing the 5G data traffic through individual instances of the real-time intrusion detection system. Thus, our findings extend and refine the work that is reported in article [74]. The experimental work that we conducted confirms that properly defined and sized virtual 5G networks are able to support even applications that process large amounts of real-time data, like the intrusion detection system.

The architecture of the proposed intelligent intrusion detection system is illustrated in **Figure 9**. The architecture of the system consists of three layers: the data traffic forwarding layer, the data management and control layer, and the machine learning-based data analysis layer. The data forwarding layer is responsible for the data traffic monitoring and capturing. It can collect and send the suspect data streams to the control layer, and it also blocks the malicious data traffic according to the instructions of the controller. The data management and control layer identifies the suspicious data patterns, and detects anomalies using the analyzed intercepted data. It also takes proper protection

measures according to the decisions made by the data analysis layer, and it consequently instructs the data forwarding layer.

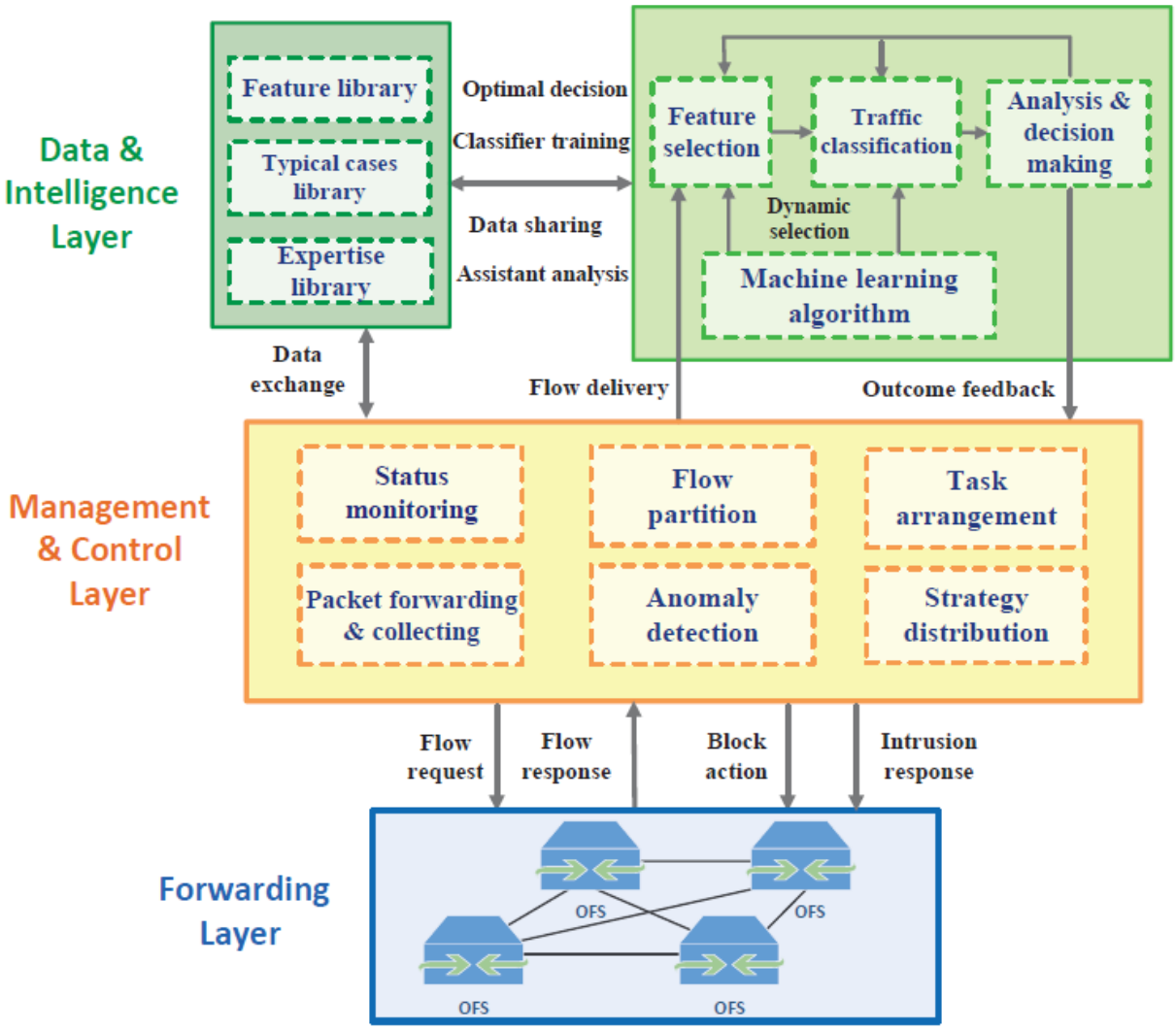

Figure 9. The architecture of the intrusion detection system.

The data forwarding layer provides the data management and control layer with real-time network status information through the real-time collection of suspect data patterns. Furthermore, intrusions are immediately blocked by dropping the malicious packets under the supervision of the other system layers.

The packet collection and data flow partitioning layer provides a more global view of the entire 5G network. The status monitoring module supervises the data network status and continuously analyzes the data packets that it receives in order to analyze them. The data management and control layer processes and parses the received data

traffic. Furthermore, it creates relevant clusters of data packets and generates a data fingerprint, which keeps track of the following logical network parameters: the source IP address, the destination IP address, the source port, the destination port, the session duration, and the considered network protocol.

The data fingerprints are used in order to define and label different data flow records, which represent specific network connections and activities. The packet collection and inspection are performed continuously. The data collection and inspection time interval is optimized in order to avoid possible undesirable delays concerning the real-time data analysis process.

The anomaly detection considers some basic flow statistics, which are used to roughly recognize abnormal behaviors and potential anomalies. The particular intrusion detection system's module applies an entropy-based analysis, which is based on the Shannon's theory in order to detect the distribution variations of the analyzed data packet samples. The entropy of a random variable *x* is computed considering the following formula.

$$H(x) = -\sum_{i=1}^{n} p(x_i) log(p(x_i))$$

Here, $p(x_i)$ designates the probability for *x* to take the value x_i considering all the already detected values. The equation considers four fundamental parameters: the source IP address, the source port, the destination IP address, and the destination port. The values of these parameters are gathered by the real-time traffic analysis component of the system. Thus, considering a particular moment in time, the continuously updated value that the entropy function *H(x)* provides helps to detect possible malicious data traffic patterns. Considering that *E* stands for the mean entropy, and *S* represents the corresponding standard deviation, a possible suspect pattern involves that the value of *H(X)* is outside the interval [*(E-S),(E+S)*]. Consequently, the suspect data packets are sent over to the proactive data analysis layer for supplementary analysis.

The feature selection component is designed in order to construct and update the features set, which is specific to the detected malicious data patterns. This component is capable to process large amounts of data in a real-time fashion, while removing the data features that are irrelevant to the machine learning core of the system's proactive data analysis layer. Consequently, the data is partitioned into relevant categories, so that malicious data traffic patterns are clearly separated from the benign traffic patterns.

The feature selection component is designed in order to construct and update the features set, which is specific to the detected malicious data patterns. This component is capable to process large amounts of data in a real-time fashion, while removing the data features that are irrelevant to the machine learning core of the system's proactive data analysis layer. Consequently, the data is partitioned into relevant categories, so that malicious data traffic patterns are clearly separated from the benign traffic patterns.

The data that are presented in **Table 8** considers the performance metrics, which determine five of the table's columns. Furthermore, the performance metrics are calculated considering several fractions of the input data set, which are mentioned in the first column of the table. Let us recall that the data set that is considered for the performance assessment contains 32,000,000 network connections that were analyzed by the intrusion detection system. Furthermore, each connection entity consists of 39 features that are analyzed by the machine learning core of the intrusion detection system. The values of the performance assessment metrics prove that the system scales well with the size of the analyzed data set.

Table 8. Numerical values of the performance assessment metrics.

Dataset size	P	R	T	A	FP
20%	97.21%	96.87%	94.24%	94.13%	0.86%
40%	97.02%	96.53%	94.05%	93.92%	1.03%
60%	96.13%	95.89%	93.68%	93.39%	0.93%
80%	96.04%	95.81%	93.48%	93.18%	0.91%
100%	95.47%	95.12%	93.04%	92.07%	1.06%

Furthermore, the system is able to accurately determine the malicious traffic patterns, while reducing to the minimum the incidence of the false positives. The practical behaviour of the system is especially important in the case of commercial 5G data networks, which transport and process a large number of data transfer sessions that have to be analyzed in a proactive manner.

The integrated system was installed on the infrastructure of a significant telecommunications services provider. The performance analysis considers the data that was effectively gathered during the real-time intrusion detection process on the provider's 5G data network. The dataset that was considered for the performance assessment contains 32,000,000 analyzed network connections. Each individual connection entity consists of 39 features that are separated into three categories. Thus, the system considers network connections-based features, content-based features, and data traffic-based features. Furthermore, each data traffic entity is marked either as a normal traffic entity, or as a suspicious traffic entity. The latter ones are grouped into four distinct categories: remote to local, probe, user to root, and denial of service.

The performance assessment considers the following metrics: precision (P), reliability (R), tradeoff (T), accuracy (A), and the false positives rate (FP). The precision is defined as the percentage of valid malicious data traffic predictions relative to the total number of predictions that the intrusion detection system makes. The reliability is calculated as the total number of accurately determined intrusion attempts relative to the total number of intrusions. Furthermore, the tradeoff represents a hybrid performance metric between the precision and the reliability, which has the role to provide a better accuracy of the data classification through the following formula.

$$T = 2/((1/P) + (1/R))$$

The accuracy is a ratio that is determined by the sum of the number of legitimate packets and malicious packets properly detected

at the numerator, while the denominator is the sum of the accurately detected legitimate and malicious packets plus the incorrectly detected legitimate and malicious packets. Moreover, the false positives rate is determined by the number of legitimate packets that are incorrectly classified over the sum between properly classified legitimate packets and incorrectly classified legitimate packets. The values of the performance metrics, which were obtained, are displayed in Table 8.

The data that are presented in Table 8 considers the performance metrics, which determine five of the table's columns. Furthermore, the performance metrics are calculated considering several fractions of the input data set, which are mentioned in the first column of the table. Let us recall that the data set that is considered for the performance assessment contains 32,000,000 network connections that were analyzed by the intrusion detection system. Furthermore, each connection entity consists of 39 features that are analyzed by the machine learning core of the intrusion detection system. The values of the performance assessment metrics prove that the system scales well with the size of the analyzed data set. Furthermore, the system is able to accurately determine the malicious traffic patterns, while reducing to the minimum the incidence of the false positives. The practical behaviour of the system is especially important in the case of commercial 5G data networks, which transport and process a large number of data transfer sessions that have to be analyzed in a proactive manner.

The 5G data networks already support relevant real-world applications, and they have the potential to become the backbone of the future always connected human society. Consequently, there are rather difficult design, implementation and deployment problems, which concern all aspects of the 5G networks. Among them, the timely detection of any illegitimate access attempt is essential, especially in the context of a commercial data network. Therefore, this paper presents the state-of-the-art concerning the research that has been made on this very important topic.

Furthermore, a real-time intrusion detection system, which is based on the utilization of machine learning techniques, is described. The performance of the system has been tested using real-world data,

which has been obtained through the real-time monitoring of the 5G data traffic on the network of a significant telecommunications services provider. This assessment demonstrates that it is possible to design a software system that blocks most of the illegitimate traffic, which occurs on a high-traffic 5G data network, in a real-time fashion. Moreover, the various existing contributions, which are relevant to the approached topic, are presented in a constructive analytical manner, while the problems that have to be addressed are analyzed, and possible solutions are suggested for their resolution. The thorough analysis, which was conducted, demonstrated that the proposed integrated solution is one of the very few approaches that is suitable to proactively scan very high traffic data infrastructures, through the design and implementation of efficient real-time algorithmic solutions.

Chapter 4. Machine learning and artificial intelligence

The author's scientific research activity in this scope has been consistent since the PhD degree was awarded. Thus, some of the selected relevant papers are represented by the bibliographic references [13], [75], [76], [77], [78], [79], [80], and [81]. Thus, paper [13] relates to the enhanced integrated detection of low-rate distributed denial of service attacks detection, which has already been presented. Moreover, paper [76] reports an extended review concerning the relevance of deep learning and privacy techniques for data-driven soft sensors, while papers [77] and [78] propose an integrated automatic driving approach for autonomous vehicles that includes an accurate and computationally efficient 3D objects detection. Furthermore, it is interesting to note that paper [80] approaches the problems of proactive attacks detection in 5G data networks using machine learning models. Considering the scientific survey that was conducted, this is one of the very few similar approaches that is proper for the real-time monitoring of data packets that are transferred through high-throughput data networks. Additionally, the contribution that was reported in paper [81] further extends this problematic through the presentation of a novel method for the identification of hardware security problems.

The following paragraphs of this chapter pertain to an interesting and relevant real-world use case scenario that is analyzed in paper [79], which concerns a research study on running machine learning algorithms on big data using the Spark unified engine for large-scale data analytics. The experimental data relates to the automatic detection of faults that manifest during the manufacturing of bearings.

The design and implementation of proactive fault diagnosis systems concerning the bearings during their manufacturing process requires the selection of robust representation learning techniques, which belong to the broader scope of the machine learning techniques. Particular systems, such as those that are based on machine learning libraries like Scikit-learn, favor the actual processing of the data, while essentially disregarding relevant computational parameters, such as the speed of the data processing, or the consideration of scalability as an important design and implementation feature. This paper describes an integrated machine learning-based data analytics system, which processes the large amounts of data that are generated by the bearings manufacturing processes using a multinode cluster infrastructure. The data analytics system uses an optimally configured and deployed Spark environment. The proposed data analytics system is thoroughly assessed using a large dataset that stores real manufacturing data, which is generated by the respective bearings manufacturing processes. The performance assessment demonstrates that the described approach ensures the timely and scalable processing of the data. This achievement is relevant, as it exceeds the processing capabilities of significant existing data analytics systems.

The scientific developments that have been achieved during the past few years imply that the valuable knowledge that can be extracted from the analysis of large-scale data sets has produced the development of the so-called "Big Data" field. Thus, the clusters and computational grids [82] allow for the analysis of big data to occur in a cost-effective way [83].

The analyzed article presents an integrated data analytics system, which considers an assessment model that aims to optimize the big data analytics processes, with a focus on the automotive industry and the necessary car parts manufacturing processes. The contribution's

relevance is enhanced by the fact that real-world data is used in order to evaluate the validity of the described model. The experimental evaluation that is reported in this paper considers a dataset concerning the bearings' fault detection. Furthermore, vibrations and acoustic signals were measured on an electric engine mounted on bearings, with four different bearing conditions: healthy, inner and outer race fault, and ball defect. The bearing condition marks the class of each entry, enabling the utilization of supervised learning. Concerning the classification process, we can talk about a multi-class classification problem, which considers four classes.

The dataset was generated using a fault machine simulator from SpectraQuest, and it was subsequently used by the authors of the work that was reported in article [84] in order to compute results on vibrations and acoustic signals under medium rotational speeds. It is worth pointing out that only parts of the dataset were used in their study, and that the choice of their algorithms is inclined towards the ones that require engineering expertise. This contrasts with the work that is reported in most of the review similar scientific contributions, where the authors report typical, general purpose classification algorithms.

The bearing fault detection dataset originally came split into 336 MATLAB files, with a total size of 19.69 GB. After the decompression has been applied, the size of the dataset increased to 28.3 GB. Considering the purposes of efficient storage and easier data processing, the 336 MATLAB files were converted into the same number of Apache Parquet files, with a total size of 9.75 GB. While there is no general agreement concerning the threshold for a dataset to be categorized as big, let us consider the suggestions in [85]. Thus, this dataset falls into the medium category, as its size belongs to the range 10 GB-1 TB, and it can be stored on a machine's storage drive, rather than in its memory. This assertion considers the specifications of a typical end user computer.

The dataset contains 14 float64 and int32 columns (features), and about 262 million rows (entries / samples). The features are as follows: BL_[X, Y, Z] (Left bearing - axis X, Y, and Z), BR_[X, Y, Z] (Right bearing - axis X, Y, and Z), MR_[X, Y, Z] (Motor - axis X, Y, and Z), [BL, BR]_AE (Left

bearing and Right bearing Acoustic Emission), [BL, BR]_Mic (Left bearing and Right bearing Microphone), defect_type. The last column represents the class and can take four different values: Healthy, Ball, Inner, Outer. These are stored as integer numbers, which take the values 0, 1, 2, 3, respectively. The speed feature is also interesting, as it depicts the motors' rotation per minute (rpm) parameter, and it has its values clustered in the vicinity of 300, 420, 540, ..., 2580, 2700, with an increment of 120 rpm.

It has already been stated that the data was converted to the Parquet [86] format, which is a columnar format supported by a number of data processing frameworks, including Pandas [87] and Spark. This conversion allowed for the input data to be easily read considering both the Scikit-learn and Spark tests. The Parquet format is also an efficient file format, which provides good compression levels and fast access to the data. The supplementary increase in the read speed of the parquet files is achieved by using Apache Arrow [88] as the processing engine, considering both the evaluated frameworks. Thus, Apache Arrow speeds up the communication between the multiple frameworks, as a parquet file is read with Pandas, and then transformed into a Spark data frame without worrying about the conversion. Additionally, it takes advantage of a columnar buffer in order to reduce input-output (IO) operations. Furthermore, it accelerates the performance of the data analytics processes [89].

The consideration of Scikit-learn in order to run the tests on large datasets can be an issue for two reasons. First, if the data are loaded with Pandas, as its usually the case, the data won't fit into memory. Second, if the testing process is conducted on the whole dataset, the computation of the results takes a long time. Thus, the reaction and the improvement can take a significant amount of time. Consequently, in an attempt to overcome these issues, four smaller datasets were generated. First, 10^2 lines were randomly sampled from each of the 336 files, each of them originally containing 780800 lines, thus resulting in a dataset of uncompressed data with the size of 3.71 MB, which represents approximately 0.01% of the entire dataset. For further reference, the name of this dataset will be *data100*. Moreover, 10^3,

10^4, and 10^5 lines were extracted in the same manner, which produced three datasets with the sizes of 37.17 MB (approximately 0.12%), 371.7 MB (approximately 1.28%), and 3.62 GB (approximately 12.8%). These datasets will be referred as *data1k*, *data10k*, and *data100k*. These subsets of the initial dataset enable the quick assessment of some initials setups, but they also enable the progressive comparison of the results of the two frameworks on larger datasets. This intends to determine the threshold, assuming there was one, where Spark becomes faster than Scikit-learn. The assumption is that for small datasets Scikit-learn would always run faster.

The proper preparation of the datasets allows for the tests, measurements and comparison of the classification jobs to be conducted on both platforms, Scikit-learn and Spark.

Scikit-learn is a framework that provides a high-level application programming interface (API) in order to easily specify and run standard machine learning models, such as Logistic Regression, SVM, and so on. It is usually used for running classification, regression and clustering algorithms on datasets that fit into the system's memory. The Scikit-learn models require as input a two-dimensional data structure, which consists of numeric values. The Pandas data frames (DataFrame) are widely used in order to satisfy this requirement, while the consideration of NumPy arrays under the hood also determines a really good computational performance.

The Scikit-learn tests consider the following runtime environment: Python 3.8.2, Pandas 1.0.3, SciPy 1.4.1, and scikit-learn 0.22.1. Considering the algorithms that are evaluated by the authors of [1] on the same data set, K-Nearest Neighbors was randomly chosen in order to be tested on the scenarios required for this paper. However, after running some initial tests, the immediate concern was to see if K-Nearest Neighbor was implemented in Spark. The answer was immediate after looking up the list of possible algorithms, and came to confirm that Spark covers most of the common classification algorithms, but not the selected one. Therefore, the decision was to choose the Artificial Neural Networks, which are also known as Multilayer Perceptrons, with the assumption that Spark would validly equate Scikit-learn, and exhibit no further limitations.

Before outlining the results of the tests from the current study, there are some aspects to consider regarding how the Scikit-learn version of the Multilayer Perceptron (MLPClassifier) was used. The considered network is composed of a 14-neuron input layer, three hidden layers with 50, 100 and 50 neurons respectively, and a four-neuron output layer. Scikit-learn required only the hidden layer setup, as the input and output were inferred from the training dataset. The maximum number of iterations was empirically fine-tuned at 500. Additionally, while the rest of the hyperparameters were left at their default values, it is worth mentioning a few of them. Thus, the activation function was ReLu, the solver was Adam, the L2 regularization term was *1e-4*, the learning rate was constant throughout the training and had a value of *1e-3*. Finally, if the score did not improve for ten consecutive iterations by at least 1e-4, the training process was stopped. Some of these hyperparameters were showcased just for easier reference, while some other ones will be important later in this discussion for the comparative analysis with the Spark implementation.

The training process of the model involves that 80% of the dataset was used, while the rest of the 20% is left for testing. Additionally, shuffling and stratifying were considered during the split operations. The assessment of the model's performance involves the usage of certain accuracy metrics, as the dataset is balanced.

It is relevant to note that the Scikit-learn algorithms and the results they generate in comparison with the Spark results, suggest that they may take advantage of the parallel computing power of multicore machines. Some of the Scikit-learn algorithms are naturally linear. Therefore, they cannot be calculated in parallel, but in the case of the MLPClassifier, one can talk about parallelization, as this benefits from the optimized BLAS implementation, which ensures the occurrence of multithreaded calls for different linear algebra routines, such as matrix multiplications. The proper implementation of this parallelization, as it is observed during the training process, implies that there are times when almost all cores of the central processing unit (CPU) are used.

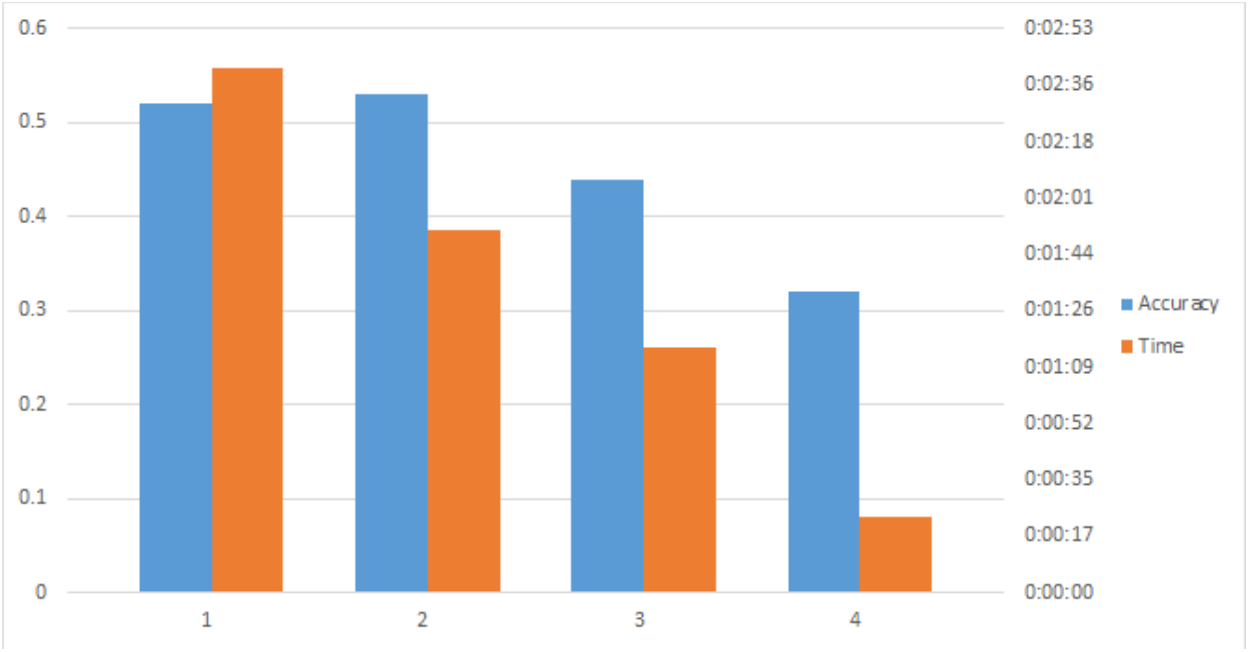

Figure 10. Accuracy and time measurements on the data100 dataset.

Thus, the graph in **Figure 10** suggests that there are four runs on four different *data100* datasets, and the median accuracy is 48%, while the maximum accuracy is around 53%. Furthermore, the mean time is 1 minute and 33 seconds, and the time varies between runs, but it is almost always directly proportional to the values of the accuracy. The last result is somewhat odd relative to the other results, and it is probably caused by applying the split operations on the training and test datasets, and also by the consideration of the MLP training.

Relative to **Figure 11**, two of the best time and accuracy measurements may be observed for the datasets *data100*, *data1k*, and *data10k*. It can be observed that the accuracy, but also the training time increase for larger datasets, which is somehow obvious. Considering an empirical point of view, the accuracy increases logarithmically, while the time increases exponentially. The figure presents the actual increase with a solid line, and the forecast with a dotted line.

The memory constraints measurements for *data100k* could not be conducted, but the forecast suggests that the accuracy could have reached nearly 70%, while the training time could have ranged between 19 and 28 hours, with an average value of 24 hours.

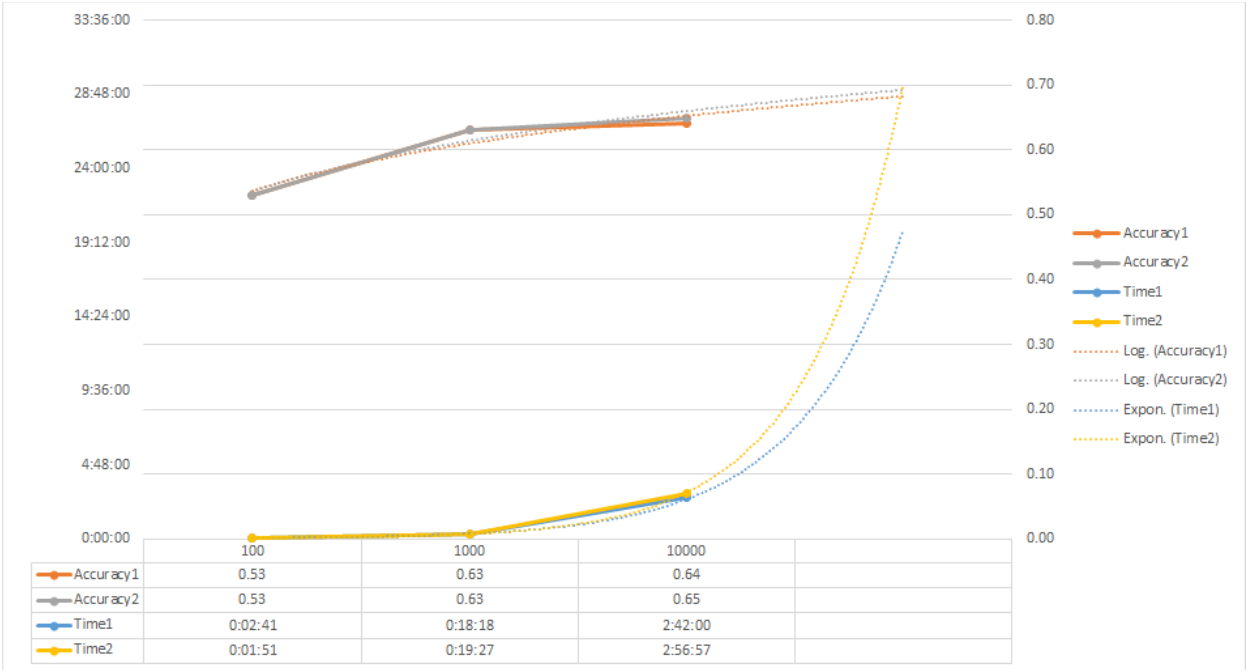

Figure 11. Assessment and forecasts for accuracy and time on larger datasets.

The Scikit-learn experiments provided a solid base of comparison with Spark, although they also brought to surface a notable limitation in the form of the memory constraints. Consequently, this approach is not scalable. Considering a project that requires scaling, it can be asserted that a cloud solution could be a better choice. Therefore, the switch to Spark appears as an optimal research direction.

Considering the measurements and comparisons that are applied on the classification jobs, the outcomes of the conceptual and experimental evaluation activities suggested that it is worth to use Apache Spark for the large-scale data processing operations, since it provides a full stack of libraries, including the Apache Spark DataFrame API to preprocess the data, and the Spark ML API for the construction of machine learning pipelines that are built on top of the DataFrame API, all of which can be programmed using Python. Furthermore, Spark provides the possibility to be run in standalone or cluster mode. The additional boost of the input-output operations, and the actual data processing speed, is determined by the consideration of the Apache Arrow.

The tests that consider Spark used the following runtime environment: Java 1.8.0 181, Spark 3.0.0-preview2, PyArrow 0.16, and

Python 3.8.2. It is relevant to note that running in cluster mode, the Python code should be submitted with the same environment as the master and workers use, even when using the Jupyter Notebook.

The usage of Spark in cluster mode considers a master/slave architecture. This means that the work needs to be submitted to the master, which will distribute the proper jobs to the workers. The process that submits the work is called a driver. Each of these three processes must benefit of the same runtime environment setup. Furthermore, the workers also represent a variation of manager processes, as they span executors in accordance with the driver request. The number of executors, as well as the number of cores and the amount of memory that an executor is able to use can be configured from the driver, usually through a *SparkSession* object. Thus, a *SparkSession* example can be seen in the source code, which is part of the reported research results.

Before outlining the results of the tests from the current study, there are some aspects to consider regarding how the Spark version of the Multilayer Perceptron (MultilayerPerceptronClassifier) was used. The considered network has the same architecture in the Scikit-learn tests. Spark requires specifying also the number of neurons of the input and output layer, besides the hidden layer. The number of input neurons is equal to the number of features in the training set minus one, as obviously, we don't use the defect type column as input. The number of output neurons is four, as there are four possible defect types (outcome classes). The maximum number of iterations again 500, as for the Scikit-learn tests. Additionally, while the rest of the hyperparameters were left at their default values, it is worth mentioning a few of them. Thus, the solver was *l-bfgs*, while the learning rate, designated as *stepSize*, was constant throughout the training and had a value of 0.03. Additionally, the tolerance value was 1e-6. Some of these hyperparameters were presented solely for easier reference purposes, while some other ones will be important later in the presentation, relative to the comparative analysis with the Scikit-learn implementation.

The training process of the model involves that 80% of the dataset was used, and the rest of the 20% is reserved for testing, while data

shuffling was considered during the split operations. The assessment of the model's performance involves the usage of proper accuracy metrics.

Thus, several scenarios are described, analyzed, discussed and compared with the Scikit-learn baseline from a computational performance perspective. The validity of the comparison is ensured by the usage of one computer with Apache Spark deployed in cluster mode. This means that the driver, master and worker are deployed on the same computer. The input files are on the local filesystem. The scenarios cascade from one to another because of certain limitations that are encountered along the way, but the results get better with each scenario, until they reach an optimal configuration.

Considering the first experimental scenario, Spark was configured to run one executor, with one core, and with 30 GB of associated memory. Although the amount of allocated memory seems to be overestimated, previous research findings prov that it is a good practice to allocate all the available memory [90]. The input was one parquet file for each of the datasets *data100*, *data1k*, *data10k*, and *data100k*. The import of the data from a parquet file into a Spark DataFrame does not imply that all the data are loaded into memory, but only the subsets that are needed considering an on-demand model.

The relevant scenario tried to replicate the Scikit-learn setup as closely as possible, assuming that the Scikit-learn algorithms ran on a single thread. Considering the two selected samples, as they are presented in **Table 9**, it can be observed that there is a big discrepancy in runtime between this scenario and the one that pertains to the Scikit-learn implementation, which led to certain investigations. It turned out that Scikit-learn's implementation of MLP uses multithreading for most of its computation. Therefore, in order to be aligned to the next scenario, multiple cores will be allocated to the Apache Spark executor.

Table 9. Comparison of Scikit-learn and Spark in Scenario 1.

	data100		data1k		data10k		data100k	
Variant	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
Scikit-learn	0.53	0:02:41	0.63	0:18:18	0.64	2:42:00	N/A	N/A
Spark S1	0.35	0:10:53	0.35	1:40:16	0.32	18:18:19	0.355	181:19:35

Considering the second experimental scenario, Spark was configured to run with a similar configuration as in Scenario 1, with the only difference that instead of one core, the executor was allowed to use all the twelve cores available on the machine.

Table 10. Comparison of Spark Scenario 1 with Spark Scenario 2.

	data100		data1k		data10k		data100k	
Variant	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
Spark S1	0.35	0:10:53	0.35	1:40:16	0.32	18:18:19	0.355	181:19:35
Spark S2	0.36	0:11:00	0.33	1:34:02	0.35	17:03:22	0.35	176:29:48

The expectation from this scenario was to obtain similar results as compared to the Scikit-learn implementation. Nevertheless, the results that are displayed in **Table 10** suggest an odd behaviour. Thus, the results are nearly the same as in the case of Scenario 1. Moreover, the inspection of the processor usage while the tests are conducted, confirms that only one core was actively running. Consequently, further research and investigation efforts defined the third scenario, which aimed to find a proper solution that effectively makes use of the available multiple CPU processing cores.

It is significant to analyze the multiple test measurements that are done in the context of both Scenario 1 and Scenario 2, as it can be observed in **Figure 12**. It is immediate to observe that the time required to run a test grows proportionally with the size of the data. Thus, let us note that the bubble sizes grow proportionally with the time, too.

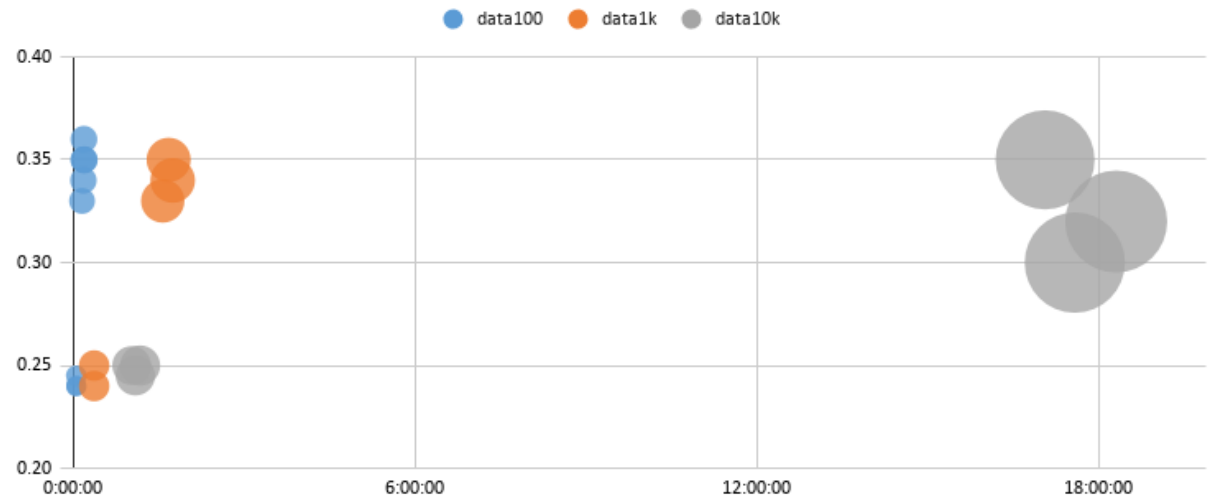

Figure 12. Comparative experimental overview regarding the Spark tests on one CPU core in Scenario 1 and Scenario 2.

Nevertheless, the same cannot be inferred about the accuracy, which seems to be in the same interval, regardless of how much data are used, more specifically around 35% and 25%. It is important to note that in the case of the 25%, one could say that learning didn't even occur, as it is the same as the random choice of a class out of the four that are available. Moreover, in the case when the accuracy is around 25%, it can be observed that the run time is significantly less compared to the 35% case. This is probably a consequence of an early termination of the neural network's training process. The design of a valid comparison with Scikit-learn implies that this aspect was considered during the selection process of the samples, which have already been described, using the results from the 35% accuracy pool.

Considering the third experimental scenario, Spark was configured to run with the same configuration as in Scenario 2, but instead of one parquet file, multiple parquet files were used as input. This resulted in Spark running jobs on multiple cores. While it may seem atypical, initially it was just a trial-and-error process, which later turned out into the realization that Spark cannot run jobs in parallel on a monolithic parquet file. The author of article [91] also asserts that this behavior only happens when the data is on the local system and not on a HDFS (Hadoop Distributed File System). Due to this shortcoming, some exploration was done on splitting data, in the following ways.

- Based on the defect_type feature, which resulted in 4 splits, which in turn allowed only 4 parallel jobs at a time, while the available computation cores were 12.
- By grouping the speed in discrete values, and split by speed, with the disadvantage of losing relevant information regarding speed.
- Introducing a new column, which would contain the remainder of dividing the index by 12, then split based on this new column, with the disadvantage of obtaining an extra column.
- Using the *maxRecordsPerFile* parameter like in the following code listing, which is supported starting with Spark version 2.2.

```
sp_df:DataFrame = spark.read.load(data_path)
count = sp_df.count()
sp_df.write.format("parquet")
   . option("maxRecordsPerFile",count//12)
   . save(r"data\split")
```

Considering also the cases where more tasks would access the same file, the last solution was used for splitting the data, into not 12, but 20 chunks to avoid concurrent file access.

The experimental results that were obtained are outlined in **Table 11**, and suggest that Spark now indeed runs in a multi-core manner, outperforming the speed of Spark Scenario 1 five times, and also matching the speed of Scikit-learn. The noticeable difference in accuracy between Scikit-learn and Spark is due to the different solvers used by the two, but it is the closest comparison to what we can get. It is necessary to note that the levels of the accuracy, which are displayed in Table 11, are justified by the consideration of the training time optimization over the accuracy in the current version of the data analytics system.

As it has already been proved, Spark can run as fast as Scikit-learn on a single machine, utilizing the full power of the CPU. Naturally, considering the next experimental step, out of scientific curiosity, but also for scalability purposes, is the attempt to obtain a new speed record in terms of training times. Along with this, another target is to break the barrier of how much data can be processed. Both of these objectives could theoretically be achieved by growing the processing

power of Spark, with the addition of new data computing nodes to the cluster. Considering the setup mentioned in the previous section, switching to a multi-node cluster setup is straightforward as Spark is already running in cluster mode.

Table 11. Comparison of Spark Scenario 2, Spark Scenario 3 and Scikit-learn.

	data100		**data1k**		**data10k**		**data100k**	
Variant	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
Spark S2	0.36	0:11:00	0.33	1:34:02	0.35	17:03:22	0.35	176:29:48
Spark S3	0.354	0:02:36	0.354	0:18:37	0.308	3:43:26	0.352	36:43:53
Scikit-learn	0.53	0:02:41	0.63	0:18:18	0.64	2:42:00	N/A	N/A

Running Spark on multiple computers can be done in a few simple steps. On each computer, the same Java and Spark versions should be installed, and in the case of running PySpark, Python should be also available on the systems. Spark needs to know the path to the Java and Python installations either through the system path variable, or other specific environment variables. Next, it is recommended to run Spark on one of the computers as master and worker on the remaining computing nodes. The Spark driver can be on the same computer as the master. The dataset is stored on each of the computing nodes.

Moreover, the Spark driver must be in the same location, because the driver cannot indicate to each worker individually, where to search for the data, it will pinpoint a single location, be it a local folder, a shared folder on the network, a path on a distributed filesystem, etc. While the most scalable solution would normally have been to store the data in a distributed filesystem like the Hadoop Distributed File System (HDFS), this would have introduced an additional overhead to the network, thus biasing the speed results, which constitute the main objective of these tests. For testing the performance of Spark, the input dataset consisted of the initially mentioned 336 parquet files.

It is relevant to note that that Spark natively handles the cases when a worker drops by whatever reasons, by assigning the unfinished tasks to other workers. With the default setup, the driver or master are still single point of failures (SPOF), but Spark can be configured to have a temporary replacement for the master or the driver, so undesirable fails in either of them can be handled without stopping the processing or losing the already computed results. The cluster for the following tests was deployed considering seven workstations with the same hardware configuration. Thus, they use hard disk drives for the local storage, 16 GB of RAM memory, eight CPU cores, which are connected physically to a switch with a 100 Mbps link speed. One computer was configured as the master and driver host, and the rest as worker machines. Each worker was configured to spawn two executors with 3 GB of RAM memory, and 4 CPU cores.

The experimental results suggest that, considering this new setup, Spark proves to be slower relative to small datasets than the setup prepared for Scenario 3. Nevertheless, the computational process gets progressively better as the size of the data grows, considering a logarithmic curve. The experimental results also suggest that the training on the whole dataset is possible using the integrated data analytics system on multiple machines, and the entire process takes approximately 45 hours.

The contribution that is presented in this paper is significant in several respects. Thus, it describes an integrated data analytics system, which is capable to fully process the large datasets that are generated during the bearings manufacturing processes using a multinode cluster setup. This achievement is significant considering that most of the existing similar approaches are able to process only subsets of the existing large datasets. Moreover, while the data analytics system that is presented in this paper is equally efficient to other similar approaches on small datasets of up to 10 MB, it is more efficient on larger datasets that are able to use the efficient machine learning-based data processing core, and the multinode cluster infrastructure.

The results of the performance assessment stage of the research that is reported in this paper, and also the relevant contributions that are reported in the existing literature demonstrate that Spark exhibits a great potential to scale. Thus, the design decision to consider Spark for the processing core of the data analytics system is completely justified. Additionally, the availability of the Hadoop Distributed File System (HDFS) optimizes the creation of the storage resources, and it also provides very high aggregate bandwidth across the multinode cluster infrastructure. Furthermore, the utilization of Spark alleviates the memory problems that are present on similar data analytics systems, which do not consider a multinode cluster infrastructure.

The proper setup of the data analytics system, which uses Spark at its processing core, requires sufficient knowledge regarding the configuration of the networking infrastructure, the creation of the adequate task schedulers, and the proper design and deployment of the required hardware and software computing architecture. Consequently, it is immediate to assert that the described data analytics system is naturally designed for the efficient and timely processing of very large datasets. The current version of the system is designed in order to optimize the training phase of the data analytics process. Furthermore, it is important to note that the architecture of the data analytics system allows for relevant data processing routines to be re-engineered. This allows for the future optimized iterations of the system to be implemented in an efficient way, so that useful improvements, such as the stronger consideration of the accuracy, will become available. The software system that was considered in order to conduct the research is publicly available at this address: https://github.com/akerestely/hpc-hadoop-spark .

The work on the data analytics system is planned to be continued in several respects. Thus, the accuracy will be considered with a higher importance during the analysis processes. Additionally, interesting hypotheses will be checked, such as whether the processing of larger amounts of data may imply the increase of the accuracy levels. Thus, several research pathways will be considered. Thus, other training and processing models will be designed and tested, which consider more efficient hyperparameters that use grid search and cross validation.

Additionally, the confusion matrix will be thoroughly studied, so that the classes that produce the prediction errors are identified. Furthermore, it is possible that the lower levels of the accuracy may be generated by an early stop of the relevant processes, which is possibly determined by the backpropagation that reached a local minimum. This hypothesis will also be attentively analyzed in the realm of future relevant research efforts.

Chapter 5. Advanced data privacy and security models

The relevant scientific contributions literally open new avenues concerning the data privacy and security models, which are immune to brute force attacks, including the attacks designed and conducted relative to existing and future quantum computers. Thus, selected peer reviewed papers relate to bibliographic references [92], [93], [94], [95], [96], [97], [98], and [99].

Thus, paper [93] defines fundamental concepts for the implementation of quantum resistant data privacy models, through the presentation of a linear layer architecture based on cyclic shift operations, together with exclusive OR (XOR) operations and logic gates. Furthermore, a semantically related contribution was reported in paper [94], which describes a novel quantum random number generator, which features an algorithmically and computationally enhanced certification method, while paper [95] describes an integrated fully functional post-quantum cryptosystem, which features a hybrid quantum random number generator.

Moreover, significant conceptual foundations are defined through paper [96], which relates to a novel hybrid method, which is relevant for randomness extraction processes. It is relevant to note that papers [98] and [99] further refine the integrated post-quantum digital signature and data privacy model, which is validated through a real-world use case scenario. This represents, to the best of our knowledge, the first scientific contribution that reports such a significant accomplishment. Furthermore, paper [97] thoroughly describes a real-world integrated solution, which relates to a post-quantum secure e-Health system that processes and manages

personal health data. It is relevant to note that post-quantum cryptography (PQC), sometimes referred to as quantum-proof, quantum-safe, or quantum-resistant, is the development of cryptographic algorithms (usually public-key algorithms), which are perceived to be secure against a cryptanalytic attack by a quantum computer. The contributions that we reported are pioneering, considering that the relevant scientific literature reports very few conceptual approaches, and virtually no practical or otherwise effectively tested solution in real-world use case scenarios. The following sections of this chapter discuss on a fundamental component of the proposed post-quantum data privacy and digital signature model, which is based on the usage of Verkle Trees and Lattices. The relevant contribution was reported in paper [92].

Research on quantum computers has advanced significantly in recent years. If humanity ever creates an effective quantum computer, many of the present public key cryptosystems may be compromised. These cryptosystems are currently found in many commercial products. We have reviewed existing solutions that seem to protect us from quantum attacks, but they are unsafe and inefficient for use in everyday life. Considering the contribution reported in this paper, hash-based digital signature techniques are analyzed. Thus, a Merkle tree-based digital signature is evaluated. Using a Verkle tree and proper vector commitments, the paper explores novel ideas. The authors of this article present a unique technology for developing a post-quantum digital signature system using state-of-the-art Verkle tree technology. A Verkle tree, vector commitments, and vector commitments based on lattices for post-quantum features are used for this purpose. The necessary concepts concerning a post-quantum digital signature design utilizing a Verkle tree are also provided in the paper.

In the future, quantum computing will become more common. Quantum encryption, a technique devised for regular computers, may protect against attacks from quantum computers. It is also called post-quantum cryptography. Quantum computers can do complex calculations much faster than current computers by using the special properties of quantum physics. For example, a quantum computer

could complete tasks that take a regular computer several years in just a short time [100].

Quantum computers will probably break most, if not all, of the standard cryptosystems currently used in practice. RSA-based systems are widely used today and they are at risk of being hacked by quantum computers. Many commercial products and applications rely on the RSA encryption system because it is one of the most commonly used public key cryptosystems, especially relative to advanced encryption technologies [101].

There have been a number of suggested alternatives to RSA systems, but none of them can be utilized in practice because of security or performance difficulties. Hash-based signature schemes are one of several that have been suggested. Since random numbers are employed as the starting random sequence of systems, their security depends on the hash function's ability to resist collisions [102]. Designing and putting into practice secure and effective post-quantum cryptosystems takes a lot of work.

There have been numerous suggestions for RSA system substitutes, but none of them can be implemented in real life because of performance or security issues. The hash-based signature approach is one of the many that have been suggested. Since random numbers are used to create systems' initial random sequences, the hash function's ability to withstand collisions is crucial to their security. Developing and implementing safe and effective post-quantum cryptosystems is a time-consuming process. When quantum computing takes over, RSA and other asymmetric algorithms will no longer be able to secure our private data. Because of this, we are aiming to create post-quantum systems [103][104].

In reality, attacks from quantum computers can compromise conventional digital signature technologies. Our objective is to create RSA substitutes that can withstand attacks from quantum computers. Hash-based digital signature schemes represent one of the choices. The cryptographic hash function is used by these schemes. These digital signature methods are secure because the hash algorithms they employ have low collision rates. The safety of these systems is determined by the security of their cryptographic hash functions.

We reviewed hash-based one-time signature schemes that make use of Merkle trees. These schemes are post-quantum and can resist quantum attacks. The problem of these schemes is a very large size of the signature. NIST has accepted hash-based digital signature SPHINCS+, but it still has the efficiency problems. SPHINCS+ is a bit larger and slower compared to the other two NIST standards, but it serves as a valuable backup for a key reason: it relies on a different mathematical approach than the three selections made by NIST.

Nevertheless, there are Verkle trees, which are powerful upgrades to Merkle trees, which are more effective and offer more efficient verification procedures by retaining only essential information. This cuts down essential space required for storage purposes. Therefore, replacing Merkle can greatly reduce the size of the signature. In the paper, we discus Verkle tree and vector commitments, which are implicit for Verkle trees.

We present a model of the novel post-quantum digital signature using Verkle trees, which substantially extends and improves the algorithmic and computational capabilities of the very few existing similar approaches. Additionally, lattice-based vector commitments are taken into consideration with regard to post-quantum properties.

Current encryption methods are easily broken by quantum computers. As a result, attacks enabled by quantum computers can now be carried out successfully. Digital signature methods that can withstand attacks from quantum computers are presented in article [100]. Paper [101] also covers approaches for one-time signatures and one-way functions. The work that was described in article [102] provides an in-depth analysis of the state of cryptanalyses, as well as the implementation of the McEliece public-key encryption system with algorithmic and parameter options.

According to article [103], the authors are interested in quantum computers. Cryptosystems that rely on the integer factoring problem are susceptible to breach by quantum computing. It implies that the RSA system, one of the most well-known public-key cryptosystems, is vulnerable to attack by quantum computers. Numerous Quantum Random Number Generation integration methods are provided in

article [104]. Different quantum number generator-based and hash-based digital signature schemes are discussed by the authors of papers [105–107]. In article [108], the Merkle plan is described in detail. Considering papers [109–111], the application of vector commitment is described. Additionally, articles [112] and [113] describe the fundamental concepts, which pertain to Verkle trees. Moreover, paper [114] relates to a Merkle tree-like construction based on the SIS lattice problem, which defines a stateless updatable VC scheme.

Hash-based signature schemes represent a type of cryptographic signature scheme that creates digital signatures using the properties of cryptographic hash functions. The basic idea behind hash-based signatures is to hash a message and then use some transformation of the hash value as the signature. Unlike traditional digital signature schemes based on public-key cryptography (such as RSA or ECDSA), hash-based signatures do not rely on the mathematical difficulty of certain problems like factoring large numbers or solving elliptic curve discrete logarithms. Therefore, these schemes can be used in a post-quantum epoch.

Considering the scope of digital signatures, crucial for identity verification in digital transactions and remote document signing requests, NIST has chosen three algorithms: CRYSTALS-Dilithium, FALCON, and SPHINCS+. Thus, SPHINCS+ is a member of the hash-based digital signatures family.

The following is how hash-based one-time signature methods operate. The creation of keys must come first. Next comes signature creation, and lastly comes signature verification. The private key for the signature scheme is created by randomly generating a secret key. The secret key must be kept private. The message is subjected to repeated application of the secret key and hash function to generate a signature for that specific communication.

The signature is the result of the hash function after every iteration. Using the same hash function and the message they received, the receiver of the signature confirms its legitimacy. The message is concatenated with the public key (obtained from the secret key), and then the hash function is applied repeatedly. If the outcome

corresponds with what was sent, then the signature is considered legitimate.

Hash-based one-time signature methods show great potential for the post-quantum era. We focus on signature schemes that rely entirely on the collision resistance of cryptographic hash functions for their security. An example of such a scheme is the Lamport–Diffie one-time signature (LDOTS) system [105]. Assuming computers have access to a constant supply of truly random bits, essentially a series of impartial and independent coin flips, it may be asserted that this approach is necessary when creating randomized algorithms and protocols. Relative to real-world applications, a sample that produces this sequence is obtained from a "source of randomness" [106].

We consider the Lamport–Diffie one-time signature's security parameter *n* to be an integer. LDOTS generates an LDOTS key pair using the following one-way function, and the related cryptographic hash function.

$$f : \{0,1\}^n \rightarrow \{0,1\}^n$$

$$g : \{0,1\}^n \rightarrow \{0,1\}^n$$

The goal of these functions is to generate a Lamport–Diffie one-time signature key pair. The following expression states that the string of *2n* bits of length *n*, which makes up the LDOTS signature key *X*, is selected at random.

$$X = (x_{n-1}[0], x_{n-1}[1], \ldots, x_1[0], x_1[1], x_0[0], x_0[1]) \epsilon R \{0,1\}^{(n,2n)}$$

The Lamport–Diffie one-time signature verification key is *Y*, which is calculated according to the following expression.

$$Y = (y_{n-1}[0], y_{n-1}[1], \ldots, y_1[0], y_1[1], y_0[0], y_0[1]) \epsilon \{0,1\}^{(n,2n)}$$

The calculation of the key is conducted using the one-way function *f*, which is represented by the following function.

$$y_i[j] = f(x_i[j]),\ 0 \le i \le n-1,\ j = 0,1.$$

The Lamport–Diffie one-time signature key generation therefore requires *2n* evaluations of function *f*. Thus, *2n*-bit strings of length *n* make up the signature and verification keys. If an LDOTS signature is generated, then the document $M \in \{0,1\}^*$ is signed using LDOTS with signature key *X*. The message digest of *M* is $g(M) = d = (d_{n-1}, \ldots, d_0)$. The LDOTS signature is defined through the following expression.

$$sign = (x_{n-1}[d_{n-1}], \ldots, x_1[d_1], x_0[d_0]) \in \{0,1\}^{(n,n)}$$

A series of *n*-bit strings is used to construct this signature. They are selected as function *d* for the message digest. A common way to measure how many cryptographic operations a CPU can execute at once is to look at hashes per second [107].

The *i*-th bit string of this signature is $x_i[0]$, but if the *i*-th bit in *d* is 0, the *i*-th bit string of this signature is $x_i[1]$. The signature can be obtained without the evaluation of *f*. The signature is n^2 in length.

Considering an instance of the LDOTS verification process, the verification of the signature attached to message *M* is conducted relative to the following expression.

$$sign = (sign_{n-1}, \ldots, sign_0)$$

Thus, the verifier produces the message digest $d = (d_{n-1}, \ldots, d_0)$. Consequently, the mathematical correctness of the process is ensured through the consideration of the following expression.

$$(f(sign_{n-1}), \ldots, f(sign_0)) = (y_{n-1}[d_{n-1}], \ldots, y_0[d_0])$$

The LDOTS generates keys and signatures fairly quickly, even with the large size of the signature. It is advised to use the Winternitz

one-time signature scheme (WOTS) to lower the quantity of signatures. The idea is to sign several bits in a message digest with a single string, or to use a single string in a one-time signature key. Similar to LDOTS, WOTS employs a cryptographic hash function and a one-way function.

A crucial characteristic of hash-based one-time signature structures is that the secret key is only ever utilized to produce a single signature. This makes sure that an attacker cannot produce additional signatures, if the secret key is compromised. There is a major security benefit to using the secret key in this particular way. We use a hash-based method only once to ensure the accuracy and legitimacy of digital signatures. This substantially enhances the computational efficiency of the proposed model, and fundamentally individualizes the proposed approach compared to the other similar solutions, which were reviewed.

Considering most real-world scenarios, one-time signature approaches are ineffective since a key pair can only be used once to create a signature. Ralph Merkle offered a solution to this issue. He recommends employing a complete binary hash tree. Using a full binary hash tree, the goal is to limit the authenticity of an arbitrary, but fixed, number of one-time verification keys to one public key, which serves as the hash tree's root.

In practice, it is difficult to use one-time signature schemes because each message requires a different key pair. The problem is that this requires storing numerous digests (n), which is impractical for everyday use. Ideally, we want a method where we can save a consistently sized digest regardless of the number of files. An idea to deal with this problem was the Merkle tree. It substitutes a single public key for numerous verification keys using a binary tree structure. Using a one-time Lamport or Winternitz signature scheme, this system integrates a cryptographic hashing function. Any of these functions, and any one-time signature scheme, can be utilized with the flexible Merkle signature scheme (MSS).

Considering this adaptability, users can select the hash function and signature scheme that best meet their requirements, and quality assurance levels. In this case, we imagine the existence of a cryptographic hash function, abbreviated as $g : \{0,1\}^* \rightarrow \{0,1\}^n$. This

essentially converts binary strings of any length to binary strings of a fixed length *n*. It is relevant to note that the Merkle system uses the hash function *g* as a fundamental building block to produce safe and trustworthy signatures. Additionally, by providing the essential mechanisms for generating one-time signatures that offer the necessary security features, the already selected one-time signature scheme accomplishes the Merkle scheme.

Relative to the person that initiates the signing request, the following element is selected: $H \in \mathbb{N}$, where $H \geq 2$. This essentially creates the MSS key pair. Consequently, a key pair is generated. This will make it possible to sign and validate 2^H documents. It should be noted that this differs significantly from signature protocols like RSA and ECDSA, where a single key pair may be used to sign/verify a large number of documents. Nevertheless, in practice, this figure is also limited by the instruments utilized to create the signature or by particular local laws [108].

Considering every $0 \leq j < 2^H$, the signing party will produce 2^H unique key pairs $(X_j, Y_j), 0 \leq j < 2^H$, where X_j is the signature key, and Y_j represents the verification key. The Merkle tree's leaves are $g(Y_j)$, where $0 \leq j < 2^H$. The following formula determines a Merkle tree's internal nodes: a parent node's hash value is equal to the sum of its left and right children. The Merkle tree's base is the MSS public key. A series of 2^H signature keys make up the MSS secret key [109].

One-time signing keys are successfully used by MSS to generate signatures. The *n*-bit $d = g(M)$ must first be computed in order to sign a message on *M*. Next, using the *s*-th onetime signature key X_s, $s \in \{0, \ldots, 2^H - 1\}$, the signer creates a one-time signature, $sign_{OTS}$. This one-time signature and the matching one-time verification key Y_s are contained in a Merkle signature.

In order to validate Y_s, the signer additionally appends the authentication path and index *s* to the verification key Y_s. There are two steps in the verification of Merkle's signature. First, the verifier employs the matching one-time signature scheme verification

algorithm to verify *d*'s signature $sign_{OTS}$ using the one-time verification key Y_S. The verifier assesses the one-time verification key Y_S trustworthiness in the second step.

Merkle trees are quick to compute. Thus, considering an *O(n)* time, a Merkle tree with *n* nodes can be created. Merkle proofs of a Merkle tree with a large number of nodes may then be unreasonably large. The tree's height needs to be *n* in order to sign 2^n messages. Our local storage may be severely and expensively taxed by the Merkle proof itself.

Strongly superior to Merkle trees, Verkle trees enable much smaller verifications and are more effective [110]. The ability of Verkle trees to reduce compute and storage costs while preserving high security makes them indispensable for post-quantum cryptography. Compared to standard Merkle trees, Verkle trees are more effective. As the amount of cryptographic data grows, Merkle trees require more processing and storage. Verkle trees provide a solution to this problem by reducing redundant data and the amount of storage space required by intermediate nodes, considering scenarios when speed and efficiency are essential relative to applications with limited resources. Verkle trees retain only the data that are required, resulting in verification procedures that are more efficient.

The primary assertion of the Verkle tree is that vector commitments, rather than cryptographic hashing functions, can be used to create a Merkle tree. We first choose how many pieces to divide our tree into (*k* pieces). After that, let us compute a Verkle tree using the files f_0, f_1, ..., f_n. After splitting our files into *k* sub-groups, we also compute a vector commitment over each of the portions of files. Additionally, for each file f_i in the subset, we determine whether each membership of the vector commitment proves PR_i with relation to VC. Then, along the tree, until we calculate the root commitment, we compute vector commitments across previously computed commitments [111].

There are nine files and a branching factor of three in the Verkle tree represented in **Figure 13**. Specific sets of size *k=3* are created from the files, and membership proofs and a vector commitment are calculated for each group. We are now left with the obligations VC_1,

VC_2, and VC_3. In addition to computing the membership proofs PR_9, PR_{10}, and PR_{11} for the commitments VC_1, VC_2, and VC_3, with respect to the commitment VC_4, we also compute the vector commitment VC_4 over these three commitments. The root commitment, in this case, is VC_4.

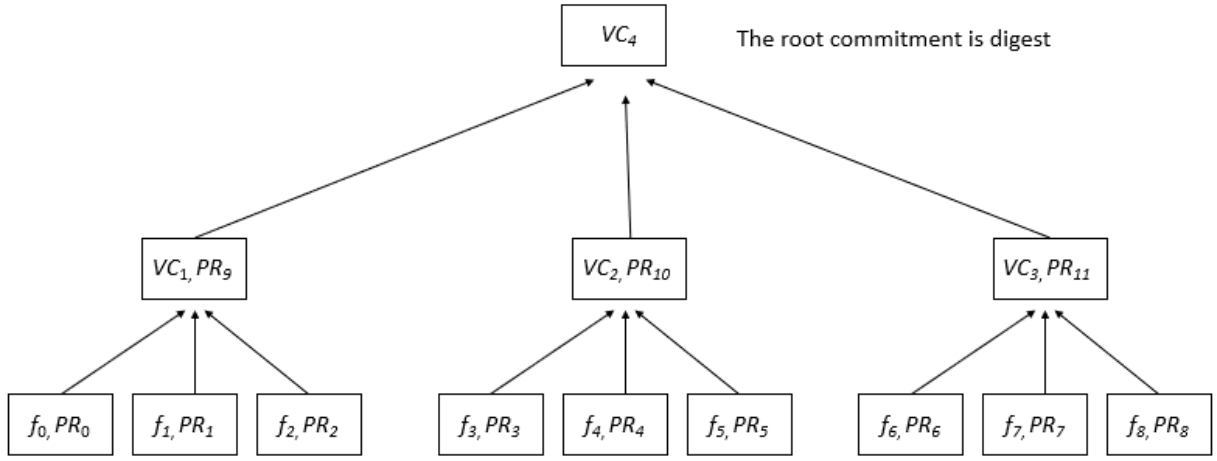

Figure 13. A Verke tree with k=3.

Each of the Verkle constructs have specific characteristics when offering Merkle and Verkle proofs, and the Verkle tree is an improved form of the Merkle tree. It is necessary to consider all of the sister nodes in the tree, whose parent has a relationship with the node it is necessary to check, if it is required to prove a value in a Merkle tree. This indicates that the proof must contain all nodes, which can take a lot of time. When it comes to providing proof, the Verkle tree requires a different approach. It relies on "batching nodes" to verify multiple pathways at once, significantly reducing the amount of evidence needed to establish a value. As a result, while proving values, the Verkle tree is quicker and more computationally effective than the Merkle tree.

Verkle trees do not even need sibling nodes, unlike Merkle trees, which only need the path and a small amount of extra information as proof. Consequently, a wider width is advantageous to Verkle trees but not to Merkle Patricia trees. Both scenarios yield shorter routes with a wider tree; however, in a Merkle Patricia tree, this advantage is lessened by the additional expense of proving each width -1 sister node at each level. This cost is absent in a Verkle tree.

A Verkle tree computes an inner node from its descendant using a hash algorithm different from a conventional hash. A vector commitment is used instead. This small parameter will help us demonstrate our point. The primary statement of the Verkle tree is that a Merkle tree can be produced by replacing the cryptographic hash functions with vector commitments. The same objective is achieved with a Verkle tree as with a Merkle tree. The main difference is that they are much more efficient in terms of size in bytes.

Commitment schemes are cryptographic fundamentals that enable a value to be hidden and later exposed. Two essential features of commitment systems are hiding, which reveals only the most essential aspects of the value, and binding, which limits access to other values.

Commitments are expanded to incorporate ordered value sequences in the vector commitment (VC) schemes. Enabling commitment to a vector and then opening at any preferred indices' binding is one of the goals of VC schemes, along with potential attribute hiding. This makes it challenging to open relative to different values simultaneously.

With vector commitment, users can commit to a vector, which represents, in this context, an ordered list of q values, instead of to individual messages (VC). This is carried out in order to enable the commitment to be opened in the future with regard to particular locations (e.g., to prove that m_i is the i-th committed message). More specifically, vector commitments are required to define position bounds. An adversary should not be able to openly commit to two different values at the same time, according to the idea of position binding. The length of the commitment string and the size of each opening must be independent of the vector length in order to meet our conciseness criteria [112].

Vector commitments may also need to have security properties, such as, hiding property, which stipulates that it should be hard to identify whether a commitment was made to the vectors $(m_1, \ldots, m_q)$, or $(m'_1, \ldots, m'_q)$. Thus, the commitment should not divulge

any information about the members of the vector, such as their values or order. The implementation of vector commitments, on the other hand, is not heavily reliant on the hiding attribute.

Furthermore, the ability to update vector commitments is required. Thus, two algorithms are provided to update the commitment and the associated openings. Considering the amendment of a commitment, *Com*, the committer can obtain a *Com'*, a modified commitment, containing the revised message by changing the *i*-th message from m_i to m_i'. Holders of a message opening at position *j* with respect to *Com* may amend their evidence using the second approach to make it legitimate with respect to the new *Com'*.

Considering our vector commitment system, multiple techniques are used for committing to, and verifying vector messages. Depending on the configuration options, the scheme uses the message space *M*, commitment space *Com*, and proof space *Pr*.

The following algorithms are provided with special programming interfaces by the proposed scheme.

- $\text{Setup}\left(1^{\gamma}, 1^{d}\right)$ - This algorithm takes security parameter γ, and a value *d* as input, and generates public committer parameters (cp), and verifier parameters (vp).
- $\text{Commit}\left(\text{cp}, m \in \text{M}^{d}\right)$ - Given the committer parameters (cp) and a message *m* from the message space M^d, this algorithm outputs a commitment $\text{c} \in \text{Com}$, and a committer state (st).
- $\text{Open}(\text{cp}, \text{st}, i \in [d])$ - Given the committer parameters (cp), committer state (st), and an index *i* from the range *d*, this algorithm produces a proof pr_i for the *i*-th entry of the committed message associated with *st*.
- $\text{Verify}(\text{vp}, c \in \text{Com}, i \in [d], m \in \text{M}, \text{pr} \in \text{Pr})$ - This algorithm takes the verifier parameters (vp), commitment (c), index (i), message (m), and proof (pr) as input, and determines whether the proof is valid or not.

The correctness condition of the scheme ensures that for any polynomial value *d* and message $m \in \text{M}^{d}$, and for a given setup

(cp, vp), commitment (c, st) obtained from Commit, and proof pr_i obtained from Open, the Verify algorithm accepts with high probability, indicating that the commitment and proof are valid. Furthermore, if the scheme offers a collection of algorithms with the following interfaces, it can be categorized as updatable. Thus, **Figure 14** provides more details in this respect.

PrepareUpdates $\left(\text{cp}, \text{st}, j \in [d], m'_j \in \text{M}\right)$—This algorithm takes the committer parameters (cp), committer state (st), index (j), and a new message entry (m'_j), and produces a commitment update (σ_c), proof update (σ_{pr}), and state update (σ_s) required to modify the i-th entry point of the committed message vector.
UpdateC $(\text{vp}, c \in \text{Com}, \sigma_c)$—Given the verifier parameters (vp), commitment (c), and commitment update (σ_c), this algorithm deterministically produces an updated commitment (c').
UpdateP $(\text{vp}, i \in [d], \text{pr}_i \in \text{Pr}, \sigma_{\text{pr}})$—Given the verifier parameters (vp), index (i), proof (pr_i), and proof update (σ_{pr}), this algorithm deterministically generates an updated proof (pr'_i).
UpdateS $(\text{cp}, \text{st}, \sigma_s)$—Given the committer parameters (cp), committer state (st), and state update (σ_s), this algorithm deterministically produces an updated committer state (st').

Figure 14. Algorithms that define the proposed updatable scheme.

Considering that the scheme satisfies the interfaces that are presented in Figure 14, it can be stated that it allows for the implementation of modifications to the committed message vector by generating appropriate commitment, proof, and state updates. The scheme can further be classified as stateless updatable if PrepareUpdates can be implemented without requiring the committer state st as an input. In this case, PrepareUpdates$^{\text{nost}}$ is used, which takes the committer parameters *cp*, index *j*, and old and new message entries m_j and m'_j as inputs.

Furthermore, the scheme can be considered differentially updatable if PrepareUpdates$^{\text{nost}}$ (and thus PrepareUpdates) can be used to implement PrepareUpdates$^{\text{diff}}$. This alternative algorithm takes the committer parameters *cp*, index *j*, and a "difference" $\sigma = m'_j - m_j$ as inputs, where the operation denotes an abstract operation on the

message space *M*. This allows for more compact representations of the updates.

The updatable scheme's correctness condition guarantees that the outputs of two experiments are statistically identical for any polynomial value *d*, committer and verifier parameters (*cp*, *vp*) supplied from *Setup*, and messages *m* and *m'* that differ in at most the *j*-th coordinate. While the second experiment creates a new commitment and proof on the revised message vector, the first experiment focuses on committing, opening, and updating the commitment, proof, and state.

Updating a commitment and proof should effectively provide results that are comparable to creating a new commitment and proof on the altered message vector. The incorporation of state information into the results enables composability, allowing for numerous updates within exponential bounds.

Compact and efficient solutions that significantly outperform earlier research in terms of the "quality" of the fundamental assumption, the effectiveness of the generated solutions, or both are made possible by the specification of proper vector commitments [113].

However, it is crucial that the approaches that emerge protect us from quantum computer challenges. Unfortunately, quantum computers can currently break vector commitments based on RSA and other popular asymmetrical systems. We are enhancing the framework to make it more effective and safer in this part. While we employ lattices to construct vector commitments, we are working on developing signature systems that will utilize Verkle trees. We build our schemes on post-quantum assumptions.

It is possible to commit to an ordered series of values succinctly using vector commitment (VC) methods, so that the values at needed points can then be demonstrated succinctly. Additionally, a vector can be stateless updatable, meaning that commitments and proofs can be updated to reflect changes to individual entries while only being aware of those changes and not the vector as a whole.

Numerous cryptographic uses have been discovered for vector commitments. Vector commitments have received significant uses for

cryptocurrencies, cryptographic accumulators, and verified external databases. They are helpful for databases that are efficiently updated and are publicly verifiable.

On the other hand, very little research has been performed on post-quantum vector commitment schemes. More precisely, these are schemes that are conceivably safe from quantum attacks. Merkle trees built with a post-quantum hash function can be employed, but they are impacted by updates that are required and relatively inefficient. We present a stateless, updatable VC scheme directly from a Merkle tree-like construction based on the SIS lattice problem [114].

We give constructions of post-quantum vector commitments based on the traditional Short Integer Solution lattice problem, suitably defined in this work. Compared to the only prior post-quantum stateless updatable construction, we present new stateless updatable vector commitments that are more efficient and have substantially shorter proofs. Considering the proposed private-key configuration, public parameters are generated by a central authority prior to its downtime. This is another aspect that specifically individualizes the reported scheme, relative to other similar approaches.

It is also relevant to note that the proposed algorithms guarantee the accuracy and security of the scheme, which enables secure commitment, opening of proofs, verification, and modifications relative to the committed message vectors.

Since an individual key pair must be used to sign each message, one-time signature algorithms are especially challenging to implement. These systems' drawback is that they require the saving of n digests, which makes them prohibitively expensive for frequent use. Therefore, we would require an approach that allows us to save a digest of the same size regardless of the number of files we have. That problem was suggested to be solved with the Merkle tree. With that approach, multiple verification keys can be replaced with a single public key by using a binary tree as the root.

Merkle trees are quick to compute. Thus, it takes *O(n)* time to build a tree with *n* nodes. A multi-node Merkle tree can be used to generate large Merkle proofs. As an example, to sign two messages, the

tree's height needs to be 2^n. The Merkle proof alone may put a significant and expensive load on the local storage resources.

Verkle trees, which allow for substantially smaller proof sizes, can greatly enhance Merkle proofs. The verifier only needs to submit a single proof that demonstrates all parent–child relationships between all commitments along the paths from each leaf node to the root, as opposed to having to submit all "brother nodes" at every level. In comparison to ideal Merkle trees, proof sizes can be reduced by a factor of 6–8, relative to Merkle Patricia trees, by a factor of 20–30 or more.

We employ the Verkle tree in place of the Merkle tree. The person signing chooses $H \in \mathbb{N}, H \geq 2$, during the key pair formation process. The key pair is then generated after that. They will make it feasible to sign and validate 2^H documents. The signer will generate 2^H unique key pairs $(X_j, Y_j), 0 \leq j < 2^H$. In this instance, the signature key is X_j, and the verification key is Y_j. It is relevant to note that these are both bit strings. The Verkle tree's leaves are $g(Y_j), 0 \leq j < 2^H$. As the leaves of the tree, they are computed and used, and every node is a hash value formed by joining the hashes of its descendants. The root commitment in the Verkle cryptography scheme is the public key. A computation of 2^H pairs of keys is required to produce a public key.

We can create signatures using one-time signature key generation. Before we can sign a message relative to *M*, we have to compute the *n*-bit digest *d=g(M)* . A message of size *n* is first created by converting a random size message of size m using the hash function. The document's signature will be created by combining the root commitment, one-time signature, one-time verification key, and finally, the proof's index *s*.

Verkle's signature verification works as follows: the one-time signature of *sign* should be validated with Y_s. If this is true, the VC_i commitments are validated. The signature is confirmed, if the root of the tree equals the root commitment. Considering a Verkle tree, the root commitment is *d*.

Merkle trees are very fast and are determined by a linear complexity of *O(n)*. Nevertheless, their proof size of $O(\log_2 n)$ is

relatively large, and may require significant computational resources. The size of their proofs $O(w\log_w n)$ is actually larger than Merkle trees when using greater width trees. Utilizing a vector commitment scheme reduces the proof size to a fixed value, *O(1)*. Nevertheless, the vector commitment construction is very costly and labor-intensive, with a complexity of $O(n^2)$. Nevertheless, the comparative algorithmic complexity analysis that is presented in **Figure 15** suggests that, overall, Verkle trees provide a proper tradeoff between algorithmic features, and computational complexity.

Scheme	Construction	Update	Proof Size
Merkle Tree	$O(n)$	$O(\log_2 n)$	$O(\log_2 n)$
Merkle Tree (w-ary)	$O(n)$	$O(w\log_w n)$	$O(w\log_w n)$
Vector Commitment	$O(n^2)$	$O(n)$	$O(1)$
Verkle Tree	$O(\mathrm{w}n)$	$O(w\log_w n)$	$O(\log_w n)$

Figure 15. Comparative algorithmic complexity analysis.

There has not been a lot of research performed on post-quantum vector commitment schemes, or ones that might be secure against quantum attacks. This paper demonstrates that Merkle trees that were constructed with a post-quantum hash function, feature updates that are relatively costly and inherently stateful. Relative to the SIS lattice problem, it is possible to obtain Merkle tree-like construction that directly produces a stateless updatable VC scheme.

Relative to the purposes of the presented contribution, it is important that resulting methods are safe against quantum computer attacks. Quantum computers could break our earlier vector commitments based on RSA. The proposed signature techniques employ Verkle trees, and vector commitments constructed using lattices. There are other Merkle algorithms, which are post-quantum, such as the Fractal Merkle algorithm [114]. Nevertheless, the experimental evaluation that was conducted suggests that the proposed solution produces the optimal computational behaviour.

Thus, the classical algorithm results are the following: Key generation time—0.049351; Signature time—0.0002425; Verification time—0.0038651. Moreover, the thread-based algorithm generates the

following computational performance parameters: Key generation time—0.013841; Signature time—0.0002425; Verification time—0.0038651.

Based on the post-quantum Short Integer Solution lattice problem, we developed a new construction of vector commitment. The protocol enables vector message verification and secure commitments. They start with a stateless updatable "base" VC construction. It is especially suitable for a relatively large d because of the quadratic dependence of the public parameters on d.

We provide a specialized tree transformation (unlike generic Merkle trees) of our SIS-based VC for larger dimensions d^h that preserves stateless updates. The proofs of construction are d-factor-concise, as the transformation relies on a VC instead of a hash function.

Our method's primary advantage is its theoretical security against quantum attacks. This has the drawback that commitment and proof sizes in the vector dimension d are logarithmic rather than constant. Despite the inherent trade-off of logarithmic commitment and proof sizes in vector dimension d, our lattice-based construction was rigorously tested against classical algorithms. The results, although slower, showcase a smaller digital signature compared to the Merkle tree version, emphasizing the practical advantages of our proposed scheme.

We tested our algorithm on the same machine, where we tested the digital signature based on the Merkle tree. The values of the resulting performance metrics are the following: Key generation time—0.049351; Signature time—0.00001520; Verification time—0.00048250. The lattice-based construction is naturally slower, but the proposed digital signature model is much better than the reference Merkle version. This is a unique scientific result, as compared to the reviewed literature.

Thus, the proposed solution represents a compelling alternative, especially in scenarios where quantum security is paramount. The trade-offs in commitment and proof sizes are carefully balanced, and the empirical results underscore the practical benefits of our approach. Consequently, the paper reported a pioneering contribution, which

considers Verkle Trees in order to implement a computationally efficient digital signature scheme, which was validation through a complex experimental process.

Chapter 6. Sensitive data privacy solutions

The papers, which relate to this scientific research scope, are determined by the following bibliographic references: [116], [117], [118], and [119]. Thus, paper [116] addresses the stringent topic of data privacy in the context of next generation data networks. Thus, it describes an integrated personal health metrics data management system relative to symmetric 5G data channels. Furthermore, paper [118], reported a secure e-Health system for the integrated management of personal health data collected by IoT devices. This proposed unique algorithmic, design, implementation, and deployment solutions, according to the thorough literature review process, which was conducted.

It is relevant to note that paper [119] fully describes a secure distributed e-Health system for the management of personal health metrics data. The proposed model considers a complex homomorphic encryption-based scheme, which allows for mathematical operations to be conducted directly over the encrypted data. This effectively implements a full data privacy model relative to the entire data collection, transmission, processing, and storage pipeline, even in the case when brute force hacking attempts are successful. This is one of the very few papers, which reports a fully functional and computationally effective homomorphic encryption model, which was thoroughly considering a real-world field trial. Therefore, the following paragraphs discuss on the problematic approached in paper [117], which constitutes the reference contribution concerning the specification of the fully homomorphic encryption model, and also its real-world validation.

The pervasiveness of personal mobile devices fundamentally changes the way personal health information is collected. These devices usually feature a comprehensive set of sensors, which include various biomedical sensing components. This indicates that the

individual subjects' physiological parameters may be conveniently monitored, while the personal health information is collected and is used for medical and other related purposes. This process generates large amounts of personal health information. Because of the limited storage and computational capabilities of these devices, the local processing of the collected data is not suitable. Therefore, the data must be stored and processed on external systems. Thus, the sensitive nature of the medical data requires safe and anonymous handling. Here, the safety encompasses two perspectives. First, the communication channel that connects the user-side mobile device to the processing and storage backend should transport the data in a secure fashion. Second, the backend should process the collected data without having access to the individual's identity or personal information. The system we describe in this paper uses a comprehensive homomorphic encryption model to preserve the data privacy. The homomorphic encryption allows for computations to be performed on encrypted data without exposing the raw data. The results of the computations are also encrypted and completely preserve the bit values of the plaintext data [120], and when they are decrypted, they match the results of the same operations performed on plaintext.

This kind of privacy-preserving data processing is useful in the context of the system that is described in this reference paper. We also suggest that this is a necessary approach when working with personal health and physiological information in the field of mobile sensing and eHealth applications.

The minutiose scientific survey, which was conducted, suggested that the existing similar approaches may often be unsuitable, as they are, for the construction of an efficient system like the one that is reported in this paper (SafeBioMetrics), while considering all the four major perspectives: the biomedical data collection at the user's end, its transfer to the storage and processing backend, the proper and secure storage of this data, and its privacy preserving processing. The SafeBioMetrics system is one of the few personal health information collection frameworks that combine the clear separation between the long-term data storage and data processing paths with the possibility to

easily attach a variety of medical sensors and data collection devices at the client side. Additionally, the backend component is able to make use of cloud storage and processing services that ensure the system's scalability in the future. Therefore, the following sections present the system considering all the features that differentiate it from most existing contributions in the field.

The standard encryption schemes, for example AES (Advanced Encryption Standard), do not allow for arithmetic operations to occur over the encrypted data. In this case, the only allowed operation is the decryption through the usage of the secret decryption key. Thus, the standard encryption schemes define an environment that securely stores the data, but it is not able to compute it.

The fully homomorphic encryption (FHE) schemes offer the possibility to perform computation operations over the encrypted data, without considering the actual plain text significance of the computed data. The SafeBioMetrics system is based on the utilization of the fully homomorphic encryption schemes, which ensure that the personal health information (PHI) is safely collected and analyzed. The personal data is processed by the backend in its encrypted form. Thus, the level of privacy is „optimal", and the overall performance, as it is perceived by the end user, is not significantly affected.

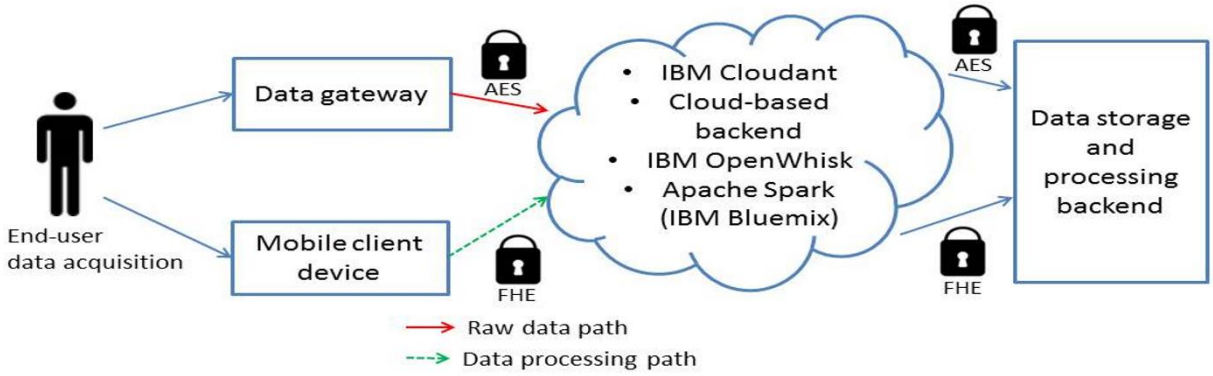

Figure 16. The general homomorphic encryption system architecture.

The system architecture is presented in **Figure 16**. The data privacy is considered during the four main stages that define the data transmission pipeline. The first stage is represented by the data

collection through each individual's wearable or portable device. Then, the second stage involves the data being safely transmitted to the data storage and processing backend. The third stage is defined by the actual storage of the patient data, while the last stage implies the safe data processing using the fully homomorphic encryption-based approach. The data storage and processing backend is deployed inside the IBM Cloud [121] infrastructure. Thus, the collected data are efficiently stored using an IBM Cloudant-based [122] storage module. Furthermore, the necessary fully homomorphic encryption computations are performed using the Apache Spark platform, which is also included within the IBM Cloud infrastructure. The processing events are intercepted and the proper actions triggered using the IBM OpenWhisk programming service [123]. The following sections offer more details on this storage and processing infrastructure.

The data transmission pipeline is intimately related to the cloud-based infrastructure considering the last two stages: the data storage and the safe data processing. The data processing results are sent back to the client devices on request. It is essential to note that the results' transmission is conducted with the data in a fully homomorphic encrypted format. This system model essentially changes the way most of the existent similar systems approach the personal health information collection and processing. Thus, the existing systems only encrypt the data at the storage backend, using a standard encryption scheme such as AES. Furthermore, the communication channels between the client devices and the server-side backend are secured, at best, with a standard encryption scheme.

Therefore, it is impossible to ensure the long-term useful storage of the data, as no data processing can be safely performed while protecting the personal health information. In other words, the SafeBioMetrics system proposes personal health monitoring options and data processing capabilities that have not been available with existing systems that rely on standard data encryption schemes.

The homomorphic encryption routines usually increase the amount of the processed data by a few orders of magnitude compared to the plain text original data. The architecture of the SafeBioMetrics system includes a data pipeline that is formed of two distinct data

buses. In Figure 16, the top data bus is intended for storage purposes, while the bottom data bus is intended for the data transfers, which support the fully homomorphic encryption operations. Thus, the data processing path is used during the fully homomorphic encryption computations, while the raw data path is used during normal patient data retrieval. Consequently, this data transmission topology ensures that health status reports can be efficiently obtained by requesting only the necessary personal health data through the storage data bus. Then, only the relevant data chunk is processed and secured using the fully homomorphic encryption mechanism, while the results are safely transmitted to the requesting mobile client device.

The fully homomorphic encryption model that SafeBioMetrics considers was extensively presented in [124]. It is called the Brakerski-Gentry-Vaikuntanathan (BGV) fully homomorphic encryption (FHE) scheme. We have thoroughly evaluated and tested the existing FHE schemes considering simulated test settings. We have found that most of the FHE schemes are prohibitively resource intensive, even in the case of capable hardware, especially because of the involved noise elimination (recrypt) operations, which are conducted after each multiplication operations [120][124]. We have found the BGV to be the only feasible solution in the context of the SafeBioMetrics system. This is because the BGV scheme defines a leveled FHE scheme, which disregards the noise elimination operation. This approach considers a better noise management algorithm, which is called modulus-switching. This optimization was explained in [125]. It allows for cascaded homomorphic multiplications (X_h) to be performed, while avoiding the risk to encounter decryption errors. This type of processing problem would be catastrophic for a system like SafeBioMetrics, as inaccurate personal health assumptions could be inferred, or even the mapping between the personal health information (PHI) and the respective individual could be rendered impossible. The following paragraphs describe the four types of FHE operations that are supported by the SafeBioMetrics system.

In essence, the system introduces a parameter *L* (the *Level*), which must be determined before starting any computation instruction. The

level *L* is calibrated considering the depth of the multiplication operations to be performed in the given computational context.

The first type of FHE operation that SafeBioMetrics supports is homomorphic addition ($+_h$). This operation takes as operands two ciphertexts that correspond to a slotwise *XOR* operation of the related plain text elements. The second type of FHE operation that SafeBioMetrics supports is the homomorphic multiplication (X_h). This operation takes as operands two ciphertexts that correspond to a slotwise *AND* operation of the related plain text elements. The multiplication increments by 1 the level *L* of the operation. Thus, the depth of the multiplication operations determines the calibrated value of the level L. The third type of operation is rotate (*<<<h, >>>h*), which essentially provides the possibility to rotate the data elements' slots. The concept of slots refers to the storage bits that determine the data elements processed by the rotate operation. The fourth operation, which is *select* (sel_{mask}), has the role to correct the potentially altered slots (bits) of the data elements after the rotate operation. Therefore, the select operation has the role to preserve the data consistency during the fully homomorphic encryption process.

The SafeBioMetrics system relies on the efficient usage on the data storage and processing backend, which must safely process the personal health information. The efficient incorporation of the fully homomorphic encryption primitives into the SafeBioMetrics system relies on the utilization of the communication data path, which is illustrated in **Figure 17**.

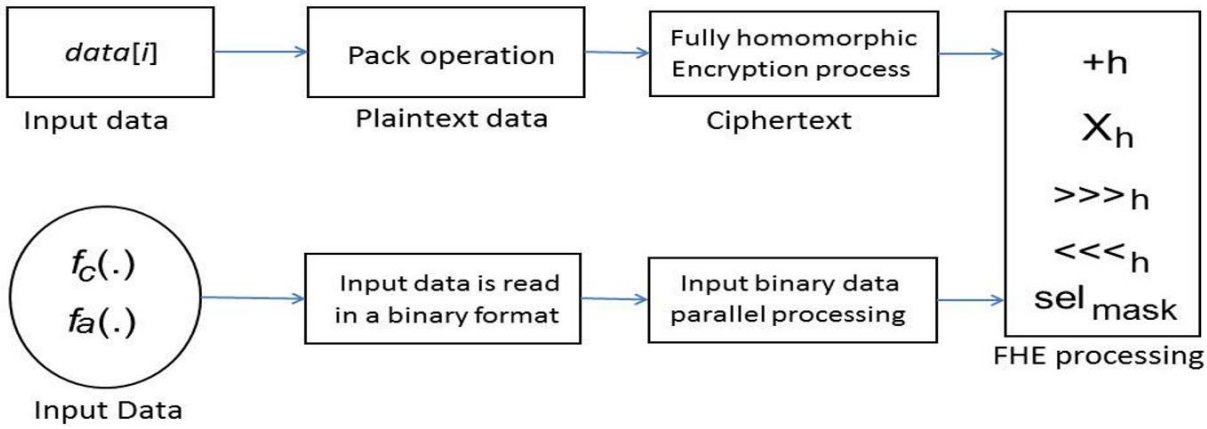

Figure 17. The logical structure of secure data flow.

Thus, Figure 17 suggests that each bit of the plaintext data is properly packed into the respective plaintext message. The ciphertext is generated through a fully homomorphic encryption model considering the steps contained by the top data path. The ability to process the encrypted data is the essential feature of this safe computational model. More precisely, the bottom data processing path, which is represented in Figure 17, suggests that the input data are translated into a binary format, which is efficiently understood by the central processing unit. This is achieved using the computation ($f_c(.)$), and aggregation ($f_a(.)$) functions that are represented as the first elements of the bottom data processing path. Following this, the binary data is optimally processed using a parallel single instruction, multiple data (SIMD) model. The data processing is performed considering all the four types of operations that have already been introduced.

The system architecture is sufficiently flexible to accommodate any use case scenario that requires the client data collection through a certain mobile device, and its safe transportation, storage and processing at the backend. The system can be configured to accommodate various existing and future mobile devices that perform the health data collection at the user's end. In order to demonstrate the validity and appropriateness for the intended scope of the SafeBioMetrics system, we focus on the collection of the cardiac rhythm data in the current version of the system. The data storage and processing backend has the ability to store the cardiac rhythm data and provide it on request to the entitled end user using the raw data path that is presented in Figure 16. Furthermore, the system has the ability to detect the delayed repolarization of the heart syndrome (DRHS) [126], and consequently report this condition.

The practical computational performance of the SafeBioMetrics system, which relies on the usage of the optimized fully homomorphic encryption scheme, depends on two factors. First, the value of the level L, which determines the operation of the FHE scheme, is an important performance factor. Second, the system performance depends on the number of multiplication and rotation operations, which are computationally expensive. The multiplication operation is also

relevant considering that it influences the calibration of the level *L*. The SafeBioMetrics system incorporates a series of improvements that pertain to the reduction of the level *L*, and it also induces the minimization of the number of FHE operations. The system is designed to compute the necessary level *L*. This operation considers a number of N_{CT} ciphertexts, which have the role to encrypt an array, with *n* bits, that stores cardiac rhythm data.

The computation of the average heart rate is based on the storage of the encrypted values in N_{CT} ciphertexts. The implementation optimization that is included in the SafeBioMetrics system considers two main types of improvements. The first one refers to the reduction of the computationally expensive multiplication operations. The second one pertains to the reduction of the computation operations depth, so that the level *L* is calibrated at the optimal level. Considering the existing few similar approaches, the proposed approach is unique both from an algorithmic and implementation perspective.

The addition operation is optimized considering two mechanisms. In the context of the SafeBioMetrics system, these two mechanisms are called the *additive compression* and the *prefixed parallel addition*. The additive compression transforms three data inputs (*H*, *M*, and *F*), each of them composed of *n* bits, into two outputs. These are represented by the A_R (addition result), and L_{OVER} (leftover). The $A_R = H\Delta M\Delta F$, and $L_{OVER} = [(H \times M)\nabla(H \times F)\nabla(M \times F)] \ll 1$. Here, Δ represents an additive single instruction multiple data (SIMD) operation, while the nabla operand (∇) also denotes a SIMD operation, which is performed on all *n* bits of the input data in a parallel fashion.

The computation of the average heart rate considers the N_{CT} ciphertexts, which encrypt the input messages that are represented on *n* bits. Thus, the first step of this data flow involves the usage of the additive compression in order to transform N_{CT} ciphertexts into two ciphertexts. Following this, the resulting two ciphertexts are added through the prefixed parallel addition operation. We will report the system evaluation results, which prove that this approach is efficient in the context of the SafeBioMetrics system.

The discussion concerning the detection of the delayed repolarization of the heart syndrome is relevant. Thus, the algorithmic

model that implements the detection of this abnormal cardiac condition is based on the usage of a certain mathematical apparatus. First, let us consider the following mathematical expressions.

$$\frac{T_{\mathrm{QT}}}{\sqrt{T_{\mathrm{RR}}}} > 475 \text{ ms} \Rightarrow {T_{\mathrm{QT}}}^2 > T_{\mathrm{RR}} \times 225,625,$$

$$\Rightarrow T_{\mathrm{QTH}} > T_{\mathrm{RRH}}.$$

Here, the expressions ${T_{\mathrm{QT}}}^2 = T_{\mathrm{QTH}}$ and $T_{\mathrm{RR}} \times 225,625 = T_{\mathrm{RRH}}$ are computed using the frontend, client-side devices that are represented in Figure 16. Thus, T_{QT} and T_{RR} represent the time intervals that are measured and recorded during any electrocardiogram test. Essentially, T_{QT} represents the time taken for ventricular depolarization and repolarization to occur, while T_{RR} measures the variability in the timing of the heartbeats. The subscript *H* denotes the homomorphic nature of the comparison, which detects the existence of the DRHS condition. We have applied an extensive set of calibration tests in order to design the optimized version of the above mathematical expression. Thus, the mathematical expression is optimized regarding the accuracy of the detection and the efficient usage of the computational resources. In the present form, the equation ensures that the SafeBioMetrics system accurately detects the DRHS condition with virtually no false positives, while running only the absolute necessary FHE operations. The data storage and processing backend aggregates the results of the individual comparisons. The client device simply requires a report from the backend considering a certain period of time. The client device decrypts the received result and checks for the presence of at least one bit that is equal to 1. If at least one such bit is found, then this is enough proof that during the given time period the comparison $T_{\mathrm{QTH}} > T_{\mathrm{RRH}}$ was true at least once. Consequently, the DRHS condition occurred with a significant probability at least once.

The detection of minimum and maximum heartbeat rates is a functional requirement of the SafeBioMetrics system and is implemented considering the $f_c(.)$ function, which is graphically represented and put in context in Figure 17. This function has the role of converting the input data into a binary format, which is efficiently processed by the SafeBioMetrics system. It has already been shown that the comparison of two numbers, which are defined by *n* bits, produces a result that is also defined by *n* bits. If the first number is greater than the other number, then the result will contain a single bit of 1, and *n-1* bits with a value of 0. If the first number is less than the other number, then the result contains only bits with a value of 0. Furthermore, the SafeBioMetrics system triggers a succession of rotate and select operations. The output of this subroutine is represented by a succession of *n* bits, each with a value of 1.

The problem of determining the minimum and the maximum values for the cardiac rate is reduced to the problem of determining the minimum and the maximum values from among N_{CT} ciphertexts, which encrypt an array of messages that are composed of *n* bits. Consequently, the correct computation of the minimum and maximum values for the cardiac rate is based on the successive application of the following functions: min($f_c(.)$) and max($f_c(.)$). In this context, the initial calibrated level *L* of the fully homomorphic encryption computation is calculated according to the following reference formula.

$$L > (\log_2 n + 2) \times \log_2 N_{\mathrm{CT}}.$$

Let us recall that the SafeBioMetrics system architecture is graphically displayed in Figure 16. The system is able to accommodate any kind of mobile data collection device, provided that it is technically capable of gathering the required personal health information. The structural versatility and stability of the system, which is also suggested in Figure 16, is determined by the fact that only the client-side data collection devices may vary. Thus, any technically suitable client-side device is able to communicate with the system and send the data to

the data storage and processing backend, without any hardware topology changes.

The client software module, which is installed on the user's mobile device, is capable of sending the collected data to the backend in real time. If the data connection is not available, then the collected data is stored locally, and immediately transferred to the backend as soon as a working data connection becomes available.

We have tested a variety of personal cardiac rate sensors, and we determined that the most accurate device is the Polar H7 [127]. Thus, it has been decided to use this device in order to collect the cardiac rate data. The personal health information, which is required to test the system's ability to detect the delayed repolarization of the heart syndrome (DRHS), is provided by a medical data set that includes 500 patients. The Polar H7 sensor has been applied on an experimental population sample, which is composed of 45 individuals. The Polar H7 device was used in order to assess the system's ability to properly collect, process, and store the data, while the dataset of 500 patients was used in order to assess the system's ability to detect the DRHS medical condition. It is relevant to note that in paper [116], which describes a functionally extended version of the reference integrated system, the whole range of available data processing operations is applied to a set of 750 enrolled patients, the personal health data are transmitted over virtualized secure 5G data channels, while the biosensors are represented by Polar H10 Heart Rate devices [128].

The system architecture is composed of the following software and hardware components. The cardiac rate data are collected by the Polar H7 personal sensor. The collected data are sent to each person's Android smartphone. The SafeBioMetrics client application is installed on the smartphone. It collects the data, which are transmitted by the personal sensor, properly encrypts it, and sends it to the data storage and processing backend, which is stored inside the IBM Bluemix infrastructure.

The software component of the backend is implemented the optimizations that have been described. The backend is deployed to the IBM Cloud infrastructure using a proper buildpack. The Apache Spark service is considered in order to optimize the data access layer of

the backend. The data that are collected from the client software modules are stored using the IBM Cloudant platform. This is a non-relational database engine, which proves to be suitable for the large amounts of data that the SafeBioMetrics system generates. The arrival of new health data in the cloud is detected by the IBM OpenWhisk programming service. Following this, the proper event handlers are triggered, so that the newly arrived data are stored by the IBM Cloudant platform. Additionally, any data request that comes from the client devices is processed by the backend considering the algorithms and data flows that are presented in the previous paragraphs.

The system's performance assessment considers three relevant metrics. The first performance metric is represented by the network capacity that is used in order to transfer the data between the client software modules and the backend, in both directions. This metric is particularly relevant in the case of fully homomorphic encryption-enabled systems because of the large amount of data that must be transmitted over the network. Let us define two performance indicators in this context. Thus, the $XFER_{IN}$ represents the amount of data that is transferred from the client devices to the backend, while the $XFER_{OUT}$ denotes the amount of data that is transferred from the backend to the client devices.

The second performance metric is represented by the *storage ratio* (S_R). This assesses the amount of storage that is necessary to store one byte of plaintext data in a fully homomorphic encryption format. As an example, if S_R=500, then it is clear that for one byte of plaintext data, there are a necessary 500 B in order to store the fully homomorphic encrypted byte.

The third performance metric is determined by the processing speed (P_S). This metric is defined through the following ratio.

$$P_S = P_{TO}/P_{IN}.$$

Here, the numerator represents the amount of time to send the data from the client device to the backend, while the denominator is

the amount of time that is required by the backend to process the received data.

Let us recall that the practical performance assessment of the initial reference integrated system was conducted considering the dataset of 500 patients in order to detect the delayed heart repolarization condition. Additionally, the Polar H7 devices were applied to 45 individuals over a period of one month.

The values of the performance metrics recorded during the detection of the heart rates are presented in **Figure 18**. The displayed table columns are structured in such a way so that each of them offers essential information regarding the state of the system's basic parameters. Thus, the table columns present, in this order, the following set of system parameters and performance metrics values: the time period that is considered when reading the client-side input data, the number of the ciphertexts N_{CT}, the value of the calibrated level L, the amount of data that are transferred to the backend, the amount of data that are transferred from the backend, the values of the storage ratio parameter, and the values of the processing speed parameter.

Data reading interval	N_{CT}	*Level L*	$XFER_{IN}$ (GB)	$XFER_{OUT}$ (GB)	S_R	P_S
One min	2	12	4.8	2,886.3	32.1	0.54
Five min	12	15	5.9	1,147.8	39.4	0.24
Fifteen min	40	18	6.4	608.2	47.5	0.23
Thirty min	44	20	9.7	1,003.5	88.3	0.36
One hr	86	21	7.4	592.8	91.4	0.35
Three hr	258	24	8.9	201.6	101.2	0.37
Six hr	519	25	10.1	98.9	108.5	0.36
Twelve hr	1,021	26	11.2	42.7	117.4	0.39
One day	2,099	28	14.3	24.6	128.1	0.42

Figure 18. The values of the performance metrics.

The performance results prove that the SafeBioMetrics system functions in a more efficient manner than existing similar approaches. The similarity with other systems only pertains to the fully homomorphic primitives, as the SafeBioMetrics system is one of the few that offers this platform for the personal health information collection in a perfectly safe and private manner. Additionally, it is

worth noting that the well-balanced (the amount of resources used is proportional to the amount of processed data) values of the performance metrics suggest that the system is scalable.

The system's performance metrics values, which pertain to the detection of the delayed heart repolarization medical condition, are presented in table displayed in **Figure 19**.

Data reading interval	N_{CT}	*Level L*	$XFER_{IN}$ (GB)	$XFER_{OUT}$ (GB)	S_R	P_S
One min	2	5	1.1	1,102.3	10.1	0.06
Five min	8	6	1.9	314.8	12.7	0.07
Fifteen min	22	8	2.4	108.5	14.5	0.10
Thirty min	41	10	2.8	83.5	16.3	0.11
One hr	85	11	3.1	69.8	29.4	0.32
Three hr	256	12	3.6	61.8	37.3	0.28
Six hr	517	14	4.8	30.2	42.5	0.29
Twelve hr	1,023	15	6.2	17.7	49.4	0.33
One day	2,079	17	7.3	9.6	53.6	0.35

Figure 19. The performance assessment regarding the detection of the DRHS condition.

It is relevant to mention that the values of the $XFER_{IN}$ and $XFER_{OUT}$ performance metrics demonstrate the suitability of the system's deployment in the cloud environment, which the data storage and processing backend uses. Thus, the cloud service providers usually charge for the uploaded ($XFER_{OUT}$) data stream, while the downloaded data ($XFER_{IN}$) is usually not monitored regarding the amount of the transferred data. Furthermore, the number of the ciphertexts (N_{CT}) is maintained at the minimum possible level, while the value of the level L is also computed in an optimal fashion. Additionally, it is relevant to note that the finer the time period granularity is, the greater the amount of the uploaded data becomes. Nevertheless, this performance metric's value increases according to an arithmetic model, and it is perfectly balanced relative to the quantity of the encrypted personal health information, which the backend provides as response to the client software module's requests.

The efficient collection of personal health data has been increasingly important during the past 15 years. As a consequence of

the technological advancements, it has relatively recently become possible to collect the personal health data considering a continuous and unobtrusive monitoring process. Thus, the amount of the collected personal data is significant and poses numerous administrative and legal challenges. The main administrative challenge is connected to the necessity to efficiently extract relevant medical knowledge out of the vast amount of stored personal health information. The legal constraints principally pertain to the imperative requirement to safely collect, transfer, store, and process the personal health information.

The reference described the SafeBioMetrics system, which addresses the entire palette of requirements that have been mentioned. Considering its flexible and decoupled architecture, the system is capable of accommodating most of the existing and, with a high probability, future client-side data collection devices. In this context, the term decoupled architecture refers to the functional autonomy of the system's components. The system's validity and efficiency are tested considering real personal health information and a real-world use case scenario. The outcome of this assessment demonstrates that the SafeBioMetrics system is capable of sustaining a perfectly functional and secure data flow between the client data collection devices and the data storage and processing backend, in both directions. This result is worth mentioning because this is one of the few integrated systems that offer the full range of personal health information collection, storage, and processing functional capabilities. Furthermore, it is significant to mention that this contribution proves that the fully homomorphic encryption can be used in order to secure a complex system like the SafeBioMetrics. In this context, the complexity especially denotes the data buses and the related data processing modules, which are in charge of delivering and processing a large amount of data. Thus, the system proves to be perfectly usable considering real-world use case scenarios. Additionally, the backend system uses the processing and storage capabilities that are offered by the IBM Cloud ecosystem, but it can be also deployed on other cloud platforms that offer similar data storage and processing services.

In conclusion to the presentation of this reference paper, it may be asserted that the reported contribution represents one of the few

computationally feasible full privacy preserving data processing integrated systems, which was fully validated through a real-world field trial. The subsequent extensions of this initial reference model, which are partly described in paper [116], prove the algorithmic and functional reliability of the proposed solution. Additionally, it also demonstrates that the system is sufficiently scalable to process large amounts of structurally complex personal health data.

Chapter 7. Software solutions deployed over next generation infrastructures

Some of the contributions that belong to this category relate to the papers determined by the following reference numbers: [14], [15], [116], [129], [130], [131], and [132]. These papers pertain several topics such as the specification and management of health parameters monitoring through an integrated multilayer digital twins architecture, a comprehensive scientific survey regarding the significance of deep learning and privacy techniques for data-driven soft sensors, or an analytically constructive review concerning the significance of 5G networks for the Internet of Things. Additionally, the following paragraphs expand on the contribution that is reported in paper [129], which approaches relevant cybersecurity aspects of IoT microservices architectures deployed over next-generation mobile networks.

The concept of Microservices Architectures (MSA) essentially relates to the larger agile developer professional community. It is interesting to note that both industry and academia have contributed to the development of the related research field [133], in an attempt to extend the capabilities of the classical Service-Oriented Architecture (SOA). Nevertheless, the effective relationship between these two architectural paradigms remains a matter of debate [134]. Thus, some proponents of MSA assert that it represents a new architectural philosophy, while some advocates of SOA almost postulate that MSA merely represents an implementational variation of SOA. The details of this intellectual debate are complemented by a sufficient understanding of Newman’s architectural principles [135].

In principle, MSA proposes the functional and logical decomposition of the application into several services, which are featured by a smaller footprint. They communicate through efficient mechanisms such as Representational State Transfer (RESTful) application programming interfaces (API) or stream-based data transmission models [133]. Usual principles and concepts [134] in the realm of MSA are “componentization via services”, which enhances modifiability, scalability, and deployability, “organized around business capabilities”, which ensures that the code is easy to read, follow and maintain “infrastructure automation”, which determines the continuous delivery processes and supports the activities of Software Development IT Operations (DevOps), and also “Decentralized governance and data management”, which favors flexibility and suitability. It is worth mentioning that relevant business conglomerates, such as Netflix, Amazon, and eBay, have chosen MSA in order to implement their sensible IT infrastructures.

Through this paper, theoretical and applied research results are used in the proposed architecture and its implementation by considering safety aspects for the development and deployment of a microservices ecosystem and these microservices’ internal structure.

Among other important advantages of a microservices-based architectural design, we can mention bounded contexts, fault tolerance, and governance flexibility [136]. Thus, bounded contexts imply that microservices are independent of one another, which means that they are interconnected only using the exposed endpoints of other microservices. Fault tolerance results from their capability to be dynamically loaded and multiplied based on the process load and can be achieved through the usage of circuit breakers patterns. These allow for the flow to continue even if errors or availability problems occur.

Microservices are deployed and afterwards operated on platforms that include hardware, software, and various types of services. The operation phase of the microservices brings multiple advantages such as independence, containerization, fault isolation, scalability, reliability, and updateability. While the monolithic application can be scaled only through a full redeployment of the whole application, the microservices can be scaled individually based on operational needs. Considering the

cloud-native development style of the microservices, the application operates in cloud or outside of it while being portable and scalable. The application services can scale based on the load. Nonetheless, using this pattern, the application can also be run in a single data center. Even if the microservices architecture and cloud-native development pattern are closely related to each other, it is possible to use them separately. That means that it is possible to create cloud-native applications without using microservices or microservice-based applications that are not built for cloud-native platforms.

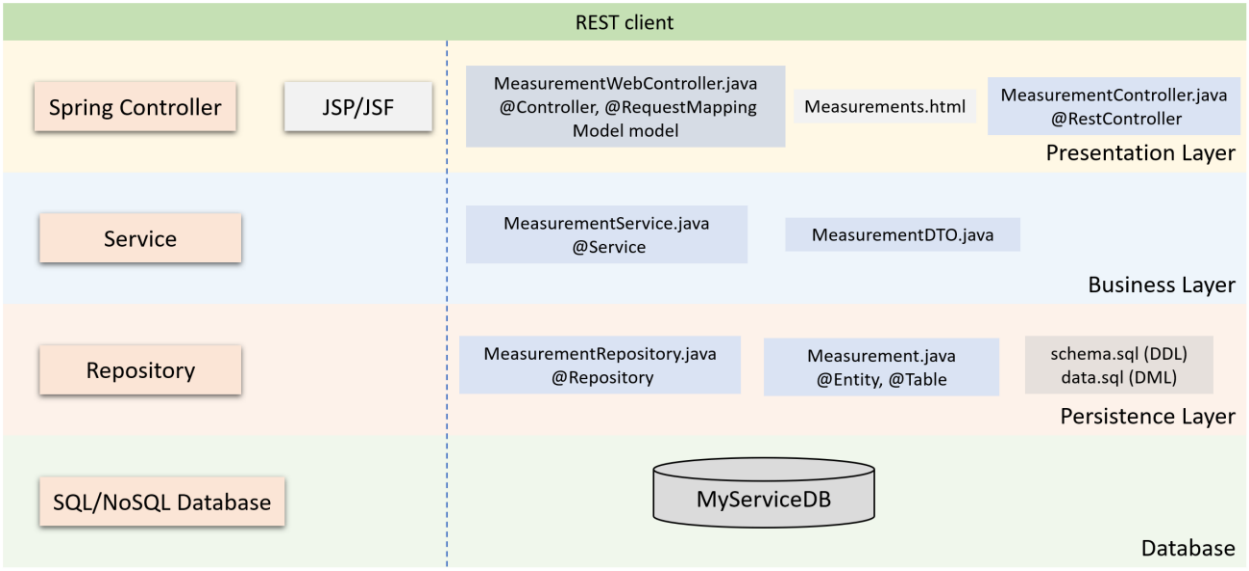

Figure 20. Multilayered architecture for shared data service.

Identity providers are a good solution for creating and managing access tokens. They provide permission management, profile management, roles, permission delegation, registration, and credential management. Additionally, they are compatible with existing security standards like OAuth [137]. Some open-source solutions can be customized and integrated into the application and used as gateway microservices in order to implement the authentication mechanisms and increase overall security.

The contribution that is described in this paper considers microservices as basic pillars of the respective system's architecture, which is concerned with the efficient and secure management of tens of thousands of electricity meters. The microservices use standard TCP data channels in order to exchange data between them and also with other system components. Nevertheless, for the TCP-based traffic,

some use-case scenarios require to be translated to DNP3-based data traffic, which is compatible with the specific Supervisory Control and Data Acquisition (SCADA) architectural components of the system.

Furthermore, let us discuss about the practical steps that can be considered in order to create a data microservice. The development and testing of microservices consider several software platforms, as follows: the Java programming language software development kit. There are multiple Java variants available that can be used. The Java versions are implementations of the JSR 390 as specified in the Java Community Process. For example, for this sample, the OpenJDK service distribution was used; the Maven build tool that helps the development process to easily manage the project files and the dependencies, which a project component uses; a suitable development environment, such as Eclipse or IntelliJ; a relational database management system, such as PostgreSQL.

The demo service application concept that is reported is a simple web application that is based on Spring Boot, which implements the microservice architecture, uses the Java Persistence API (JPA) specification, and also considers the server-side UI framework Thymeleaf. The Thymeleaf template engine allows the design and manipulation of a graphical user interface controls and events. It should be noted that Spring Boot can be used in order to develop serverless/independent applications or web applications, which are managed and run with the help of application servers.

Thus, **Figure 20** presents the internal architecture of a single microservice. The internal architecture of a microservice is organized on multiple tiers, each tier with its own responsibilities. The consideration of the separation of concern principle implies that the microservice can be easily designed, coded, and improved, if required.

Being a simple service, a graphical user interface is also created next to the service, even if, in the production environments, this is not mandatory since the views are independently created and managed, and they are often based on different technologies. The service publishes a single endpoint only be accessed using the exposed endpoint. The creation of a data wrapper microservice, which is represented in **Figure 21**, that is compatible with the system that is

described in this paper or with other similar software systems relates to the following steps: Generation of the Spring Boot project using the online tool; Import the skeleton project for further development into the Eclipse; Populate the database with data; Define an entity (Measurement entity class); Create a repository (services for working with entities); Create a business logic service; Implement the controller for using the service from previous step; Creation of a simple UI for direct service access using the Java-based Thymeleaf templates engine; Embedded server port configurations. Modify the application configurations by changing the connection port using the application.properties file; Define a REST controller for the service in order to work with it without a graphical interface.

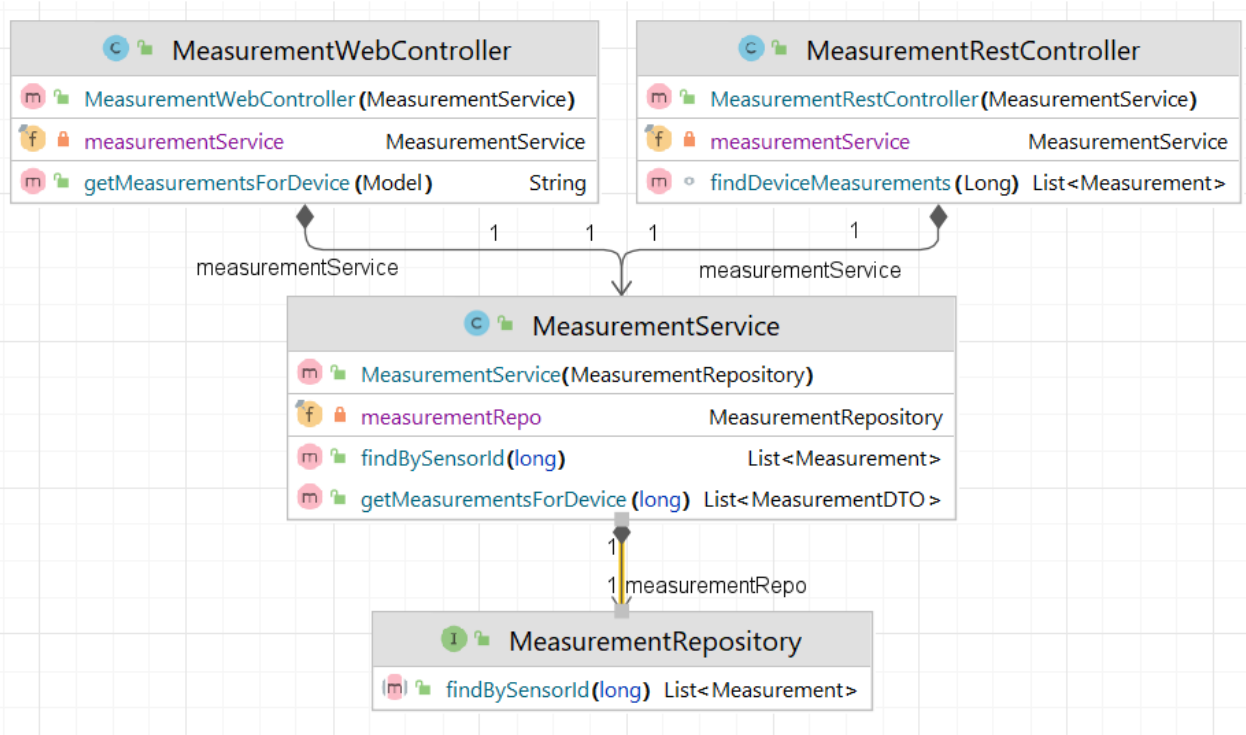

Figure 21. Class diagram for data service.

The authentication mechanism, in the case of legacy applications, is based on the server, which keeps the secrets and, when it is queried, the provided credentials are validated, and the access authorization is provided. The server keeps track of login and session information in order to prevent relevant cybersecurity problems, such as the replay attack. Often based on the clients' needs, the provided tokens are stored during their working sessions (e.g., as cookies). The microservice

authentication process implements different mechanisms due to their distribution and asynchronous model of action. A widespread method that is used for authentication is JSONWeb Token (JWT). The method enables the delegation of authentication to external services in a stateless and space-efficient manner. JWT implements other standards, such as JSON Web Signature (JWS) or JSON Web Encryption (JWE), for message authentication and encryption.

A JWT consists of three parts: header, payload, and signature. Let us note the example provided in the RFC specification and listed in the table displayed in **Figure 22**. The header section indicates that the media type is application/jwt and specifies the algorithm used for digitally signing the token, in this case, HMAC SHA-256.

Section	Payload
Header	{"typ":"JWT", "alg":"HS256"}
Payload	{"iss":"hpcc", "exp":1672444800, "http://cs.unitbv.ro/is_root":true}
Signature	Iuu5xStItPU5at/Nkvme4V6IfWUWZmTl0bX0EimXYxo

Figure 22. Anatomy of a JWT.

The payload section is used for storing the claims. Claim Names are case-sensitive and have to be unique. Otherwise, applications can reject the JWT or use the lexical last claim name as a fallback.

For interoperability purposes, some names are registered in the IANA *JSON Web Token Claims* registry. These include the issuing entity (*iss*), the issue date and time (*iat*), as well as the expiration date (*exp*). The last claim from the payload section in Figure 22 represents a custom claim, which signifies that the user successfully authenticated as an administrator on the specified domain, and services can use it for authorization purposes.

To build a complete JWT, the contents of the different sections will be encoded with base64url and concatenated with separating dots, as it is suggested by the table displayed in **Figure 23**.

JWT Token Section	
eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9	Header
.	
eyJpc3MiOiJocGNjIiwgImV4cCI6MTY3MjQ0NDgwMCwgImh0dHA6Ly9jcy51bml0YnYucm8vaXNfcm9vdCI6dHJ1ZX0	Payload
.	
Iuu5xStItPU5at/Nkvme4V6IfWUWZmTl0bX0EimXYxo	Signature

Figure 23. Sample base64 encoded JWT section.

According to the specification, the JWT implementations only have to include the signature algorithms HMAC SHA-256, the other fields being optional. Considering an application that implements microservices-based distributed architectures, a microservice could handle the login process and provide a JWT that would be stored on the client side. The validity of the JWT could then be checked either on the API Gateway or by each microservice that receives a request.

When connecting to the user interface, the application detects that the user possesses no valid authentication token and redirects them to the login page. After entering the credentials, a request is sent to the */user/login* path, which contains the email and password in the form of JSON records. The ingress controller identifies all URLs starting with */user* as belonging to the authentication service and consequently routes them accordingly. The authentication service checks the credentials against the ones stored in the directory (or database), and if valid, it issues a pair of authentication and refresh tokens.

Using the valid tokens, the user is considered to be logged into the whole application, and can use the other microservices, as it is also suggested by **Figure 24**. Firstly, the user is forwarded to a dashboard application, where they can access the other functionalities using the specific menu and options. A request containing the newly acquired JWT in the authorization header is sent to the accessed service at the location /serviceroute. Before sending the response, the service must first check the validity of the authentication token. In the background, the JWT is sent to the authentication service for verification. If the token is valid, the service proceeds to reply with the requested data using a JSON format.

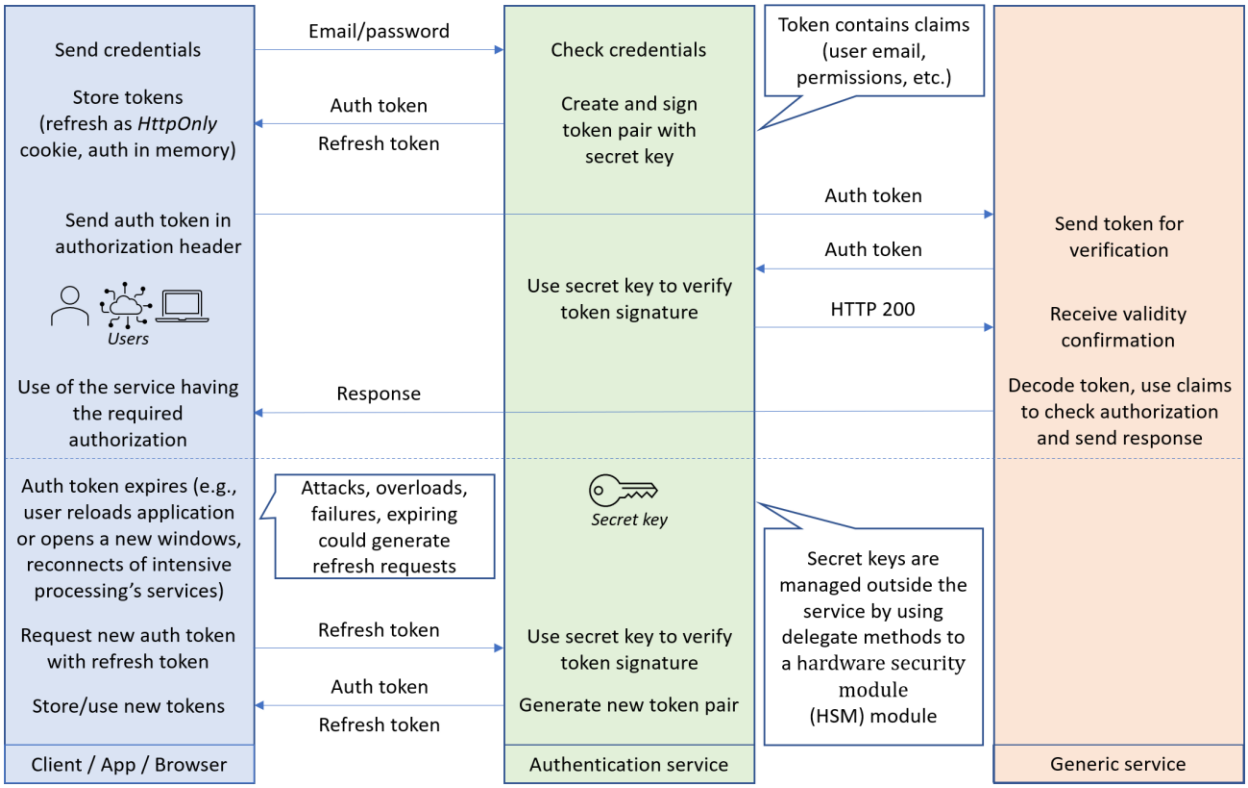

Figure 24. Authentication token sequence.

The design and development of microservices that also wrap up the use of IoT devices and their communication with complex architectures were discussed, considering different optimization aspects in the existing literature [138]. When properly implemented and used, such features offer more performance and increased reliability to any IoT-enabled platform. Despite their importance, some of these optimizations are hidden in the implementation of the considered software frameworks, which are represented by context and dependency injection mechanisms, related concepts and benefits. The reusability of instances relates to relevant safety aspects and can further be connected to the cybersecurity aspects of the relevant architectures and platforms. For the specific case of an IoT system, appropriate research methods, for example, learning-based methods [139], have to be applied to treat the difficulties faced by IoT devices after the occurrence of a cyberattack.

The case study for the architecture experimented on by the authors and presented in this paper was the use of smart metering

devices, which are related to energy consumption. In simple terms, the scenario implies that the clients of such a platform have to access their consumption in real-time using the proper web interfaces, and at the same time, the energy provider can do estimations concerning the consumption for its entire network of clients, while respecting all the regulations that pertain to the tariffs. Nevertheless, the energy provider can be different from the smart meter provider and also from the transport company in such a way that a heterogeneous combination of stakeholders can read or write information into the systems. It implies that the security specifications and requirements are clearly defined concerning the smart metering PKI, the encryption algorithms, security modules, protection profiles, and administration, together with the respective operations and processes [140].

Thus, **Figure 25** represents the context architecture for the whole ecosystem of smart meter gateways. Thus, it considers the proper microservices-based architecture, which is represented in Figure 20. This is oriented on a single data source service. The operation of a smart meter implies a multitude of related microservices. Essentially, a smart meter gateway is placed in front of a single consumer and a single electricity meter. Even so, the stakeholders that need access to the measurement data and also to the relevant reading context are different.

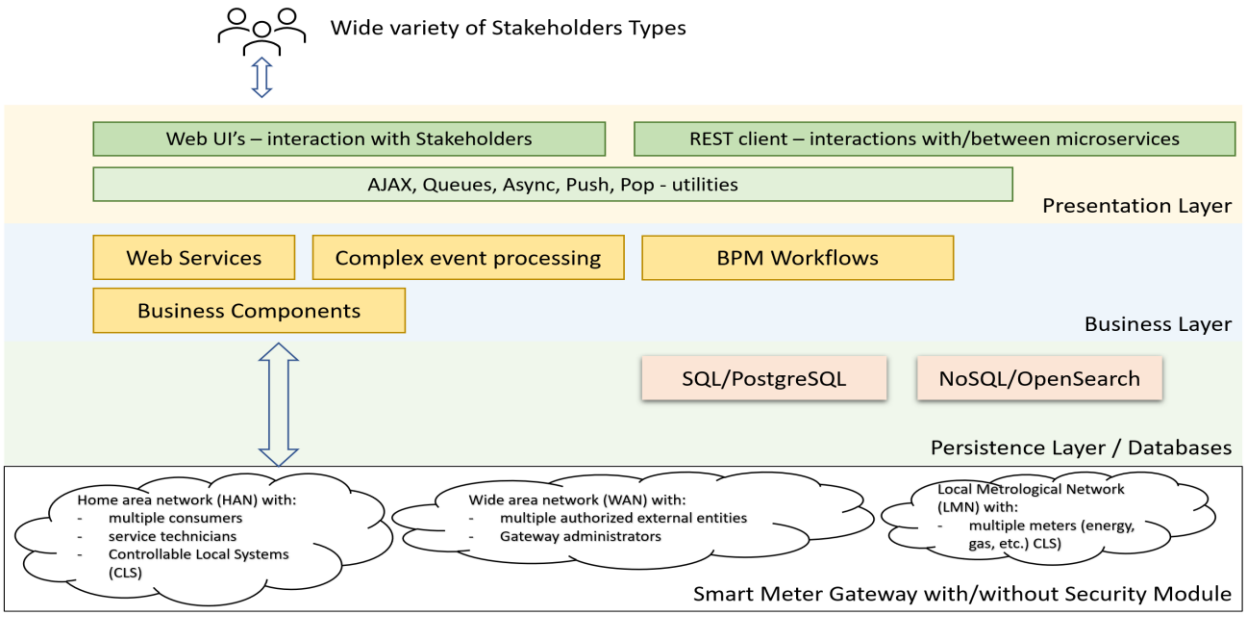

Figure 25. Smart meter architectural context.

The data that are collected by the enrolled electricity meters are transmitted in a fully encrypted form over insecure data transmission channels.

Similar to other types of intelligent devices, the updates represent a usual process which must be secured. The updates must not affect the actual electricity meters data collection processes. First, the new system has to be uploaded on the meter, and then the updated firmware should be triggered considering the chosen time intervals.

One of the simplest optimal ways to deal with the smart meter gateway-related processes is to encapsulate its functionalities into separate microservices. As an example, considering the readings a microservice performs, the collected data are packed into a specific telegram that is valid within the platform. The telegram is published using the platform queues and consumed by other microservices as needed in order to provide their functionalities. The telegram is encrypted using the certificates installed on the smart gateway and agreed with the platform. Each time, when it is required, the exchanged telegram between platform components can be verified by the certification authority.

The effective real-world demand, not only for smart meter types such as intelligent thermostats and electricity meters, but also for other kinds of smart meters, is rapidly growing on the market. Together with this objective market increase, the theoretical and empirical research results determine the related hardware and software progress. For example, in Germany, one in ten (10%) households use smart meters, which amounts to 3.7 million households out of the overall 40.7 million [140]. Thus, private homes need to manage and control their smart metering systems through online channels in order to save energy. The increasing energy prices and the need to avoid shortages place this market on an ascending trend, as it is demonstrated by the tabular data displayed in **Figure 26**.

Smart Device Type	% Net Income under 2500	% Net Income 2500 to under 18,000
Smart TV	46	71
Smart Speakers	9	21
Smart household appliance	8	18
Smart energy management	5	14
Smart security system	7	12

Figure 26. German households equipped with smart devices and systems, 1 January 2022.

Smart Grids spread across countries need to be reliable in terms of offering stable and scalable services and need to be secured against increasing security threats and attacks. Let us recall that smart meters also measure gas and water consumption, in addition to electricity. The cumulative number of smart meters is increasing, and these IoT devices need to be managed, controlled, and queried using intelligent software architectures. The system architecture that is proposed in this paper represents one of the few integrated solutions which fulfils all the necessary logical and functional constraints.

The (auto)scaling of the microservices and concurrent consuming the information provided by the contained IoT devices is an undoubted advantage. To keep the services available, further security aspects, such as throttling (avoiding capacity being exhausted by denying some traffic to avoid denial of service attacks, manage quotas) and rate limit (avoiding bursts of traffic which can cause degradation or the outage of service), were considered.

The microservices-based approach succeeds at pointing out the boundaries between different components. A software architect or even developer would have few or no problems identifying what the functionalities offered by a service are. Thus, the separation of concern was defined by properly designing the services' responsibilities. Additionally, fault tolerance represents another advantage.

Architectural loose coupling of the microservice containers allows their replacement and restart without negatively impacting the other services. The poorly engineered services are easily replaceable during

development, while the rest of the application continued to operate as expected. Deploying changes can be as easy as pushing a commit to the master branch if Kubernetes is set up to check for newer versions of the Docker images automatically. Another advantage comes from reusability. The authentication service could easily be used in a different application without further modifications. Implied data are also isolated at the database level and can be used or backed up independently. Containerization means that the services can be easily deployed using commercial cloud infrastructures, as well. Adding features might prove to be simpler than working with monolithic applications while requiring only the development of a new microservice instead of modifying the existing ones and taking the risk of introducing new bugs. Moreover, the web interface can be multiple-sided, covering different components of the ecosystem that manage data readings, end clients, smart meters, and so on. Under normal circumstances, each team that develops a microservice should be responsible for the entire developed microservices-based architecture. This approach is also recommended for complex applications. Microservices-based architectures are better suited for complex applications, which require a high degree of flexibility and scalability. Indeed, it is often recommended to start with a monolithic structure and consequently break it up into microservices only when and if the need arises.

Designing a system around microservices might prove to be a more difficult architectural task than designing standard applications. New and exciting technologies, such as containers and orchestration layers, help the design, development, and deployment processes.

Multiple token-based standards were developed, such as OAuth, OpenID Connect, and JSON Web Token. The token-based authentication models cover the two main phases of the service requests, namely, the initial authentication and also the refresh (update) of the token. According to the architectural considerations and principles introduced in this paper, the services are separated, and they can be easily replaced when updated libraries are published, or new versions of security token standards are made available.

Identity providers may be represented by self-developed services, which manage their own credential databases and tokens, or they can be open-source identity stores, such as Okta, Keycloak, LDAP, or a combination of these. Their use is wrapped into customized services, or their own RESTful API can be used in order to secure the respective microservices-based architectures. Web redirect mechanisms are used to forward system requests to the authentication service using secure communication channels.

There exists a large pool of client types that are connected to microservices-based architectures, such as other services, phones, IoT devices, browsers, etc. Some of the clients are owned by particular organizations, while other clients are represented by external entities. Limited access to external clients must be granted through firewalls. Each type of client must be able to access the identity service in order to retrieve the needed token or to access the application functionalities. The access rights to a group of microservices cannot always be granted through a reverse proxy microservice. The trusted network is difficult to isolate. The management and validation of tokens should be accessible in the case of each microservice, which is part of the overall software system.

The tested access scenario is based on the usage of the API gateway that considers the Identity and Access Management (IAM) platform, with the possibility to customize the gateway or extend the IAM functionalities.

A reverse proxy pattern acts like a single-entry point towards a particular resource pool. Defined by an API gateway, it allows for the required security controls to be defined. Using certain policies, such as client quota, an API gateway can trace requests and monitor relevant performance parameters. An IAM platform provides capabilities that overlap with those offered by API gateways. One of the main advantages of the IAM platforms is represented by the existing implementations of multiple security standards, such as OAuth Primer.

Although some of the scientific contributions that have been reviewed propose valuable algorithmic, architectural, and practical features, they miss certain functional features, which may be considered in order to describe robust security models that evolve and

may be partially used in order to impose proper access policies in the particular scope of microservices-based architectures. This paper surveys relevant research articles, which highlight valuable ideas and concepts, but also relevant drawbacks, which were analyzed during the design and development of the approach that is proposed in this paper. It analytically presents and discusses security aspects concerning an integrated microservices-based system, which offers the necessary functional features and security mechanisms.

The system is evaluated considering a real-world use case, which relates to the management of tens of thousands of electricity meters. This dimension determines a complex use case scenario, which presumes the efficient collection of the customers' data, its secure transmission to the microservices-based components, as well as the optimized timely processing of the collected data using the relevant microservices-based software modules. The thorough real-world performance assessment demonstrates that the proposed microservices-based architecture is capable of properly managing the enrolled electricity meters, and it also offers the required scalability, which would allow the enrolment of additional customers to occur without significant practical issues.

It is relevant to note that the authors of this paper are members of the "**High Performance and Cloud Computing**" research group, which is part of Transilvania University of Brasov, Romania. This group is concerned with scientific research topics that pertain to distributed systems, cybersecurity, and IoT. Therefore, the continued development of the proposed microservices-based system is among the conceptual and practical scientific priorities of this research group.

Chapter 8. Autonomous driving solutions

It is important to note that the contributions, which are reported in papers [77] and [78], present the results of a fruitful cooperation with two major car manufacturers. Thus, a fully functional prototype was designed, implemented, and validated through an extensive field trial. Therefore, the reported solution holds the distinction of being one of the few similar solutions, which features the necessary algorithmic

and computational advancements that allow for autonomous driving scenarios to occur through an efficient detection of the environmental 3D objects.

The relatively complex task of detecting 3D objects is essential in the realm of autonomous driving. The related algorithmic processes generally produce an output that consists of a series of 3D bounding boxes that are placed around specific objects of interest. The related scientific literature usually suggests that the data that are generated by different sensors or data acquisition devices are combined in order to work around inherent limitations that are determined by the consideration of singular devices. Nevertheless, there are practical issues that cannot be addressed reliably and efficiently through this strategy, such as the limited field-of-view, and the low-point density of acquired data. This paper reports a contribution that analyzes the possibility of efficiently and effectively using 3D object detection in a cooperative fashion. The evaluation of the described approach is performed through the consideration of driving data that is collected through a partnership with several car manufacturers. Considering their real-world relevance, two driving contexts are analyzed: a roundabout, and a T-junction. The evaluation shows that cooperative perception is able to isolate more than 90% of the 3D entities, as compared to approximately 25% in the case when singular sensing devices are used. The experimental setup that generated the data that this paper describes, and the related 3D object detection system, are currently actively used by the respective car manufacturers' research groups in order to fine tune and improve their autonomous cars' driving modules.

The detection of 3D objects is placed at the core of the autonomous driving components. This process involves the estimation of 3D bounding boxes that define the objects' spatial position and orientation, and identifying the category to which the objects belong in the environment. The detection of 3D objects is usually performed using machine learning techniques, and the assessment may consider either real-world or synthetic data. As an example, there are well known datasets, such as KITTI [141], which include images gathered by frontal cameras, and also data that define the 3D boxes annotations. It

can be stated that rich texture images, which are generated by the cameras, are essential for the proper and efficient classification of 3D objects. Nevertheless, light detection and ranging laser-based cameras (Lidar), along with depth cameras, may provide useful data that can be used in order to determine the objects' spatial position and orientation. In this context, it can be asserted that many detection schemes consider data that is gathered from multiple sensors, in order to increase the actual detection performance and accuracy.

Nevertheless, the consolidated sensor data may prove vulnerable to certain functional problems. This category of problems includes occlusion, restricted perception horizon that is related to a limited field of view, and also the low-point density relative to distant regions. Thus, a solution to these kinds of problems may be represented by cooperative data collection that includes various sensors. Existing reported contributions demonstrate achievements, although relatively limited, concerning lane selection, maneuver coordination, and automated intersection crossing. This paper considers the goal to detect 3D objects, which pertains to the data that is cooperatively acquired from multiple sensors. Thus, instead of using the disparate data that is collected from individual sensors, the solution is to combine the collected data (cooperative perception). This determines significant advantages, such as an increased perception of the horizon, and a reduced occurrence of noisy data.

The 3D object detection, which is based on cooperative perception, may be accomplished through two particular modalities, late fusion (LF) and early fusion (EF). The modalities are named in regard to temporal sequence, as the fusion may occur before or after the 3D object detection stage. In late fusion, each data acquisition is regarded and processed in an independent manner. The detected 3D boxes are combined (fused) into a resulting 3D entity. Moreover, early fusion processes the acquired raw data and combines (fuses) it prior to the detection stage. Consequently, late fusion schemes are classified as high-level detection schemes, while early fusion belongs to the category of low-level detection schemes. While both approaches are essentially capable to enhance the perception horizon and field-of-view, only the early detection scheme is able to efficiently use the raw

image data. Thus, let us consider the typical example of a vehicle, which is partially covered (occluded) as it is perceived from two different detection positions. In such a scenario, each particular sensor is determined by a certain occlusion model, which produces an essentially unreliable or even unsuccessful detection. Consequently, the combined data may provide the possibility to successfully detect the 3D object.

Considering the aspects presented above, this paper proposes the detection of 3D objects through both schemes, the late and early detection schemes. This essentially results in a detection scheme that supports the late and early detection models, which can be used as required at the level of the central data processing components. The contribution that is presented in this paper considers a system, which aims to produce a precise construction of roads, regardless of their complexity. This includes problematic road components, such as roundabouts and T-junctions. The system was tested using the data obtained from experimental sensors that are installed at road level. This approach is capable of providing the necessary data to allow safe autonomous driving scenarios to occur in virtually any scenario. Consequently, safe autonomous driving can be conducted on even the most complex road segments.

This paper reports on the following contributions. Two cooperative schemes for the detection of 3D objects are described, which consider the mentioned fused detection schemes, and a custom training model. The algorithmic routines are optimized through a comprehensive experimental process. This allows for the implemented software components to perform better than existing similar approaches. The dataset that is used for the assessment procedure is obtained through cooperation with automotive industry partners. The late and early fusion schemes are comprehensively assessed using real-world data. Additionally, the importance of the data acquisition sensors' deployment is analyzed and discussed, with significant results concerning the deployment of such a system in a real-world scenario.

The proposed model, which represents an extension and algorithmic optimization of the Frustum PointNet (F-PointNet) fusion-based method [142], considers *n* data acquisition sensors. Each of them is capable of providing depth sensing capabilities, which

include laser-based Lidar, and also depth assessment functions. Furthermore, each data acquisition component features a local data processing unit (processor). Let us recall that we proposed a centralized data acquisition model for the detection of 3D objects. This implies that the dispersed data acquisition sensors send the collected data to the central processing components using wired or wireless data links. Moreover, it can be stated that the sensors that were used in order to collect the data are properly calibrated. This means that their precise spatial position and orientation are communicated to the central data processing components. The original F-PointNet fusion based model usually processes the front view images. This constraint implies that this approach has difficulties in properly detecting 3D objects in certain regular environmental conditions, for example at night time. Although F-PointNet is regarded as one of the most reliable fusion-based methods, this drawback, that manifests often in practice, it is unacceptable for our automotive industry partners. Therefore, we have extended and improved the original F-PointNet fusion based method, in such a way that the model that we propose is able to accurately detect 3D objects in virtually any environmental conditions. The following sections describe and assess the proposed model.

The central data processing components possess enough computational power in order to process and combine the data that is received from the data acquisition sensors. Additionally, the central data processing components are able to synchronize and send useful data to nearby vehicles using a radio system. This implies that the vehicles that are used possess the necessary hardware components in order to receive the radio signals from the central data processing components. Furthermore, the enrolled vehicles possess the necessary computational capabilities in order to locally process relevant data, which pertains to the implied autonomous driving processes. This implies that data, which relate to environmental perception and trajectory control, are processed. It is relevant to note that the central data processing components are not responsible for the transmission of actual trajectory control signals for the enrolled vehicles. Thus, the respective vehicles use the data that are supplied to them by the central data processing components in order to perform the necessary

data processing and make their own decisions concerning the proper control of their trajectories. The rationale behind the presence of the central data processing components, as an architectural component of the proposed approach, is to offload computationally expensive data acquisition and processing operations from the actual vehicles, and allow them to make more precise decisions.

The 3D objects detection model that is described in this paper processes point cloud data, which can be provided by laser-based Lidar devices or depth cameras. The laser-based Lidar devices are able to generate point clouds natively. The images that are generated by depth cameras should go through a post-processing phase in order to generate the required point clouds. It is necessary to note that each sensor, which is part of a certain hardware assembly, determines data points that pertain to its own coordinates system. Consequently, the respective acquired data must be translated to the global coordinates system, which can be properly handled by the central data processing components. This translation is based on the operation of rotation, which is followed by a proper translation phase. This effectively associates data points that are generated by the individual data acquisition sensors to the central coordinates system, which is determined by the inverse of the extrinsic matrix [143] that is associated to each data acquisition sensor. Let us consider the coordinates (a, b, c) of a 3D point in the coordinates system of sensor s. Consequently, the global reference point (a_r, b_r, c_r) may be determined considering the following formula.

$$\begin{pmatrix} a_r \\ b_r \\ c_r \\ 1 \end{pmatrix} = Mat_i^{-1} \begin{pmatrix} a \\ b \\ c \\ 1 \end{pmatrix} = [Rot_i | Trans_i] \begin{pmatrix} a \\ b \\ c \\ 1 \end{pmatrix}$$

Here, Mat_i represents the extrinsic matrix of sensor s. This can be decomposed into a rotation matrix Rot_i, and also into a translation vector $Trans_i$. The extrinsic matrix Mat of a sensor s, and consequently Rot and $Trans$ are adequately determined through a calibration process. This can prove to be a challenging process in practice,

considering that *Mat* depends on the spatial position and orientation of the implied data acquisition sensors.

Thus, the accuracy of the obtained result directly depends on the measured values of these variables. Therefore, considering that the sensors are placed on mobile data acquisition nodes, which are typically represented by a vehicle, any error that pertains to the localization data of the mobile node determines alignment issues concerning the fused point cloud. This may imply that non-existing 3D objects are detected, or existing 3D objects are not detected. Let us recall that the model that is described in this paper considers data that is collected by sensors that are placed at the side of the road segment, which is used in order to conduct the necessary experiments. The successful transformation of the point data that are collected by the individual data acquisition sensors into the coordinates system that is considered by the central data processing system, implies that any data which do not specifically pertain to the detected 3D object and its vicinity are removed.

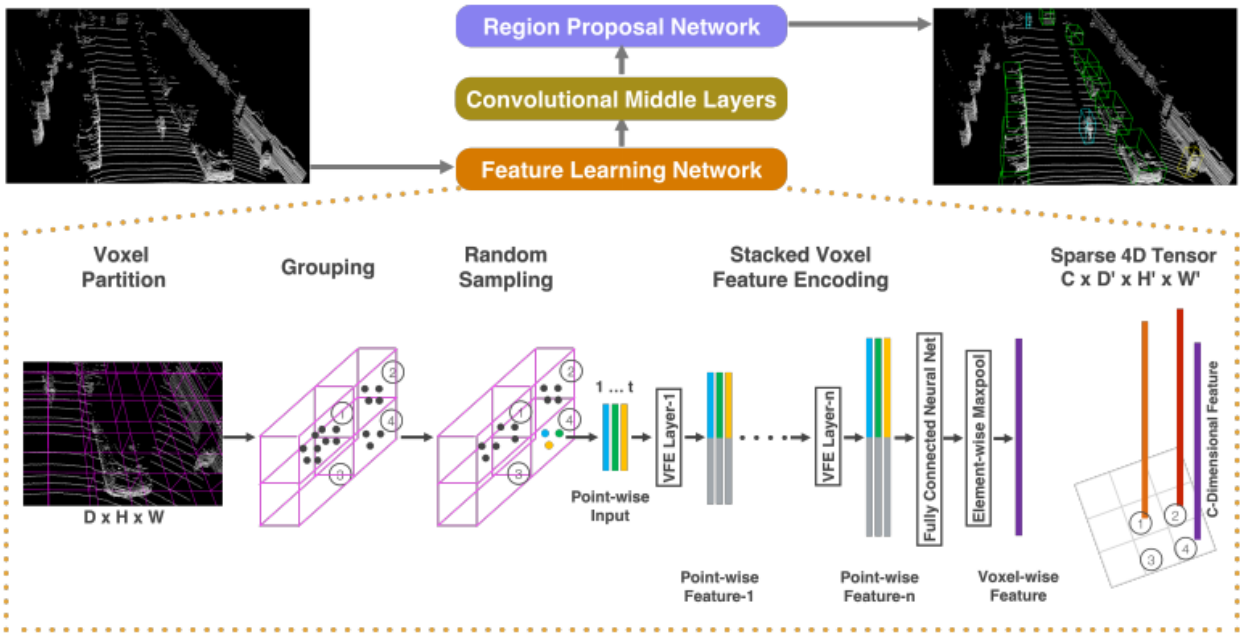

Figure 27. The early fusion scheme.

The basic architectural and logical structure of the early fusion scheme is illustrated in **Figure 27**. It can be immediately noticed that it is based on the fusion of the point clouds, which are generated by *n* data acquisition sensors. This allows for the data that pertains to the 3D

objects in the analyzed environment to go through an aggregation process. The experiments that we conducted show that this significantly increases the successful detection of the 3D objects that are affected by noise. The data processing workflow goes through a series of distinct phases. Thus, the preprocessing phase is the responsibility of each individual data acquisition sensor. This phase generates a cloud of *n* points that is referred in the coordinates system that is understood by the central data processing components.

Furthermore, each individual point is transmitted to the central data processing system, which aggregates the individual data points it receives into a single data point, and which is then sent as input to the 3D object detection component. This particular component produces, as an output, a list of objects, which defines the required 3D bounding boxes. Consequently, these data are sent over to the enrolled autonomous vehicles.

The objects detection component is made, from an architectural perspective, of three sub-components. Thus, it includes a feature learning network, multiple convolutional middle layers, and a Localized Proposal Network (LPN).

The feature learning network has the role of transforming the 3D point cloud data into a constant size representation, which can be efficiently processed by the convolutional layers of the neural network. Let us note that the original Voxelnet model considers the lasers-based reflection intensity channel, as well as the 3D spatial coordinates of the detected point, let us refer to them as (*a*, *b*, *c*). The 3D objects detection model that is described in this paper exclusively considers the spatial coordinates. This enables the data processing phase to efficiently produce the points that can be easily understood by the central data processing components. Thus, the input point cloud is separated into voxels of equal size, let us refer to them as (vox_a, vox_b, vox_c). Here, the components represent the width, length and height, respectively. Considering each voxel, a set of *t* points is selected in order to create a proper feature vector. Additionally, let us consider that *T* is the threshold concerning the maximum number of points for each individual voxel. If *t* is greater than *T*, a randomized sample of *T* points is generated. This has the role of decreasing the

computational load, and improving the balance regarding the points distribution between different voxels. Moreover, the coordinates of these points are transmitted to a chain of Voxel Feature Encoding (VFE) layers. Thus, each particular VFE layer is composed of fully connected layers, local aggregations, and also max-pooling operations [144]. The output of this neural network is represented by a 4D tensor, which is indexed considering the following features of the voxel: dimension, height, length and width.

The middle layers in the convolutional neural network are particularly important. In essence, they add to the data processing pipeline three additional stages relative to the 4D voxel tensor, which has been presented. These stages include spatial data from the voxels in the vicinity, which add the mandatory three-dimensional context to the considered features map.

The Localized Proposal Network (LPN) receives the tensor that is obtained. The LPN network is structured considering three stages of convolutional layers in the neural network. They are immediately processed through three supplementary transposed convolutional layers. Consequently, a high-resolution features map is obtained. The generated features map is considered in order to produce two output branches. The algorithmic model considers a confidence score, which indicates the probability for a 3D object to be present in a certain analyzed scene. The algorithmic model also uses a regression map, which determines the position, orientation, and size of the processed bounding box.

The late fusion scheme combines the list of 3D bounding boxes, which is generated by each data collection sensor. It is possible that an object is not properly detected as a consequence of the noise levels in the acquired image samples, or because of the occlusion. Consequently, the central data processing components become unable to properly process the collected data. The workflow starts with the preprocessing of each individual point cloud. The output is provided to each data collection sensor, which generates a list of 3D objects that are defined by their 3D bounding boxes. Consequently, the entire set of detected 3D objects is sent to the central data processing components, which combine them into a unitary set of objects. We have determined

that the unitary set of objects may contain multiple representations of the same object entities, as they are detected by several data acquisition sensors.

Therefore, we have developed an algorithm that determines the overlap between different bounding boxes. In the case when the overlap between any pair of bounding boxes is above a specific threshold, the bounding box that is featured by the lower value of the confidence score is removed from the set. Thus, the value of this score essentially represents the probability with which an object is present inside a particular bounding box. The experiments that we conducted demonstrate that this approach successfully eliminates the bounding boxes that overlap. Additionally, the supplementary processing that it requires does not induce a substantial overhead on the overall 3D objects detection process. It is essential to note that the central data processing components conduct the required data processing operations. Consequently, the list of detected 3D objects is transmitted to the autonomous vehicles that move in the neighbourhood, which use this data in order to make their autonomous driving decisions.

The experimental dataset contains data that were gathered by data acquisition sensors, which are statically placed at the edge of the road. The data acquisition sensors' placement is adequate in order to create the two basic scenarios, the roundabout and the T-junction. The data acquisition sensors are capable to capture depth and RGB image data at a resolution of 640x480 pixels, while the horizontal field of view is provided at a 90-degree angle. The area of the T-junction is monitored by ten data acquisition sensors, which are mounted on vertical masts with a height of 4.3 m. The balanced acquisition of data is ensured by the setup of the sensors, which implies that five of them target the incoming direction, while the other five point to the T-junction's opposite direction. Furthermore, the roundabout scenario considers twelve data acquisition sensors, which are placed on masts at a height of 6.1 m. Considering the roundabout data acquisition sensors, it can be stated that six of them scan the incoming road lanes, while the other six sensors monitor the outgoing road lanes. The placement of the data acquisition sensors, considering both scenarios, was optimized through an empirical onsite process.

The experimental dataset is composed of four independent sections. Thus, two data collections pertain to the T-junction scenario, while the other two are related to the roundabout scenario. Each data collection contains 24,000 training image samples, and 1000 test image samples. The image sample, as an aggregated entity, is defined as the entire set of RGB and depth images, which are acquired by all the installed data acquisition sensors at a particular instance of time. Furthermore, each image sample also includes data that relates to the detected 3D objects' spatial position and orientation, dimension and category.

The 3D objects that are defined and stored in the experimental dataset represent four main categories: pedestrians, cyclists, motorcyclists, and vehicles. These categories are represented in the data set with the weights 0.2, 0.2, 0.2, and 0.4, respectively. This ensures that the actual vehicles are assigned a proper weight, which corresponds to the real-world situation. The data curation phase ensures that the experimental dataset allows for each 3D object to manifest during eight image samples. This increases the structural diversity of the 3D objects that are stored in the experimental dataset, while allowing for a greater number of spatial positions and orientations to be stored. Additionally, it can be stated that the actual movement of the 3D objects considers the standard traffic rules.

The detection areas are determined by a rectangle with the dimensions of 100 by 50 m, in the case of the T-junction scenario. Additionally, in the context of the roundabout scenario, the detection areas are determined by squares with sides of 90 m. The area that is covered by the data acquisition sensors has a size of 4300 square meters in the case of the T-junction scenarios, and 12,200 square meters in the case of the roundabout scenarios. The algorithmic core of the proposed 3D objects detection model takes into consideration the mathematical model of a laser-based Lidar sensor, which is described in article [145].

The experimental dataset is processed through a training process considering the following steps. Thus, a particular instance of the 3D objects detection model is trained considering the data that is provided

by multiple data acquisition sensors, and considering the algorithmic process that has already been described. The training process considers a Stochastic Gradient Descent (SGD) optimization during 90 epochs. The learning rate is 10^{-4}, while the momentum is 0.8. Moreover, a loss function is considered, which penalizes the regression relative to the position, size and yaw angle.

Considering the voxel size as (vox_a, vox_b, vox_c), then the anchor stride that goes along the dimensions *X* and *Y*, in the case of the T-junction, is set to (0.3, 0.3,0.5) m and 0.5 m, respectively. The consideration of exactly the same hyperparameters in the case of the roundabout is not feasible because the covered area is approximately three times larger. This would provoke computational problems as a consequence of the impossibility to store all the features maps in the graphics processing unit's (GPU) memory. Therefore, the spatial coverage of the axis *X* and *Y* is reduced, in the case of the roundabout, through a voxel size of (0.4, 0.4, 0.4) m, and an anchor stride of 0.8 m.

The algorithmic core, which actually performs the 3D objects detection, is mostly designed to isolate vehicles from the processed image samples. The other three entity categories, pedestrian, cyclist, and motorcyclist, have the role of contributing to the prevention of the statistical phenomenon of overfitting, since they allow for the trained model to learn the necessary distinct features for the vehicles.

Additionally, the proposed algorithmic core is designed to apply rotations to the determined bounding boxes. The angle of these rotations is selected, considering a randomized model, from the interval [-26,26]. The rotation is applied in order to compute the rotation angle in an as general as possible way. Furthermore, the rotation also contributes to the prevention of the model reaching a state of overfitness.

The effective performance of the proposed 3D objects detection model is evaluated considering the two road scenarios, the T-junction and the roundabout. Additionally, the variation in the number of data acquisition sensors is assessed in connection to the accuracy of the 3D objects detection process itself.

The 3D objects detection process is assessed using four performance metrics. They are intersection relative to union (IRTU),

recall, precision, and the communication cost, which is defined by the average volume of data that is transmitted between a data acquisition sensor and the central data processing components relative to each image sample. The communication cost is measured in kilobits. Let us recall that the concept of image sampling has already been described in a previous section.

The intersection relative to union essentially measures the spatial similarity of a pair of bounding boxes, one which is normally selected from the set of estimated bounding boxes, while the other is selected from the ground-truth set. This is determined by the following formula.

$$IRTU(B_{gt}, B_e) = \frac{volume(B_{gt} \cap B_e)}{volume(B_{gt} \cup B_e)}$$

Here, B_{gt} and B_e represent the ground-truth and estimated bounding boxes, respectively. The set of estimated bounding boxes encompasses all the positive entities. These are the bounding boxes that are determined by the 3D objects detection algorithm considering a confidence score that is greater than a certain threshold, which is denoted by *Trs* in this paper. Let us recall that the metric *IRTU* also considers the location, size, and the yaw angle of both bounding boxes, which essentially represents the orientation. Thus, the value of this metric is 0, if the respective bounding boxes do not intersect, while it is 1 if the two bounding boxes are identical in terms of their size, orientation, and location. The value of the *Trs* has been calibrated through successive experimental trials. Thus, we have determined that a value of 0.75 generates an optimal balance between the quality of the 3D objects detection, and the utilized computational resources.

The precision is defined by the ratio between the number of estimated bounding boxes that are matched, considering the already mentioned definition, and the total number of bounding boxes that are part of the estimated set. Furthermore, the recall is determined by the ratio between the number of estimated bounding boxes that are matched relative to the overall number of bounding boxes that are part of the ground-truth set. It is natural to observe that precision and recall are, in essence, functions of *Trs*.

Considering the fusion schemes that are presented in the introductory part of this paper, the purpose of this experiment is to compare the performance of early fusion (EF) and late fusion (LF) schemes considering the effective detection performance, and also the communication cost and computational time. The evaluation considers both road topologies, the T-junction and the roundabout. Considering the late fusion scheme, the algorithm uses an *IRTU* threshold Trs with a value of 0.17. This value has been experimentally calibrated in order to prevent the same 3D object being detected multiple times. The values of the performance metrics can be studied in the table displayed in **Figure 28**.

Scheme and Topology	Communication Cost (Kilobits)	Computation Time (ms)
LF T-junction	0.39	217
EF T-junction	471	296
LF Roundabout	0.17	139
EF Roundabout	541	228

Figure 28. Comparative performance analysis between late and early fusion schemes.

Let us note that the communication cost is computed for each sensor, while the computation time is calculated for each image sample. Let us recall that the concept of image sample has already been defined in a previous section.

The experimental results that are presented in Figure 28 suggest that the more efficient 3D object detection in the case of the early fusion schemes comes at the expense of a greater computational cost. This behaviour is determined by the larger data volume that is transmitted in order to send the raw point clouds from the data acquisition sensors to the central processing components, as compared to the similar data transmission in the case of the late fusion. Furthermore, the experimental results that were obtained demonstrate that the early fusion variant exhibits a higher performance concerning the 3D objects detection in the case when the threshold *Trs* is assigned higher values. The best performance was obtained for Trs=0.92. Furthermore, considering a particular value of *Trs*, the 3D object detection performance is more efficient in the case of the T-junction

road topology, as compared to the roundabout topology. This behaviour is determined by the larger voxels that have to be computed in the roundabout scenario, as it has already been explained in a previous section.

Additionally, the results that are presented in Figure 28 suggest that the early fusion variant determines a higher communication cost. The explanation resides in the larger volume of data that has to be transmitted in order to accommodate the raw point clouds. Furthermore, it should be observed that the required capacity of the data transmission link is dependent on the frequency of the processed image samples. As an example, considering a frequency of ten image samples per second, which is common in the case of laser-based Lidar sensors, the required capacity of the data transmission link is 4.71 Mbps. This number is obtained through the multiplication of the communication cost value by 10, considering there are 10 image samples that are processed in a second. The field experiments that were conducted demonstrate that this transmission rate can be easily accommodated by the wireless or wired data connections that are available along the road that is monitored by the data acquisition sensors. Moreover, considering that the temporal sequence of the transmitted image samples is relevant, then it can be stated that the network latency may produce problems concerning the actual 3D object detection. The field studies that were conducted did not detect this problem, and we are able to make the assumption that this problem may occur only in cases when the latency of the data link is greater than a few tens of milliseconds.

The computational time that is necessary in order to process each image sample is greater in the context of the early fusion schemes, as there are more point clouds that have to be processed than in the case of the late fusion scheme. Furthermore, the computational time depends on the GPU that is used in order to deploy the central data processing components. The experiments that we conducted considered an Nvidia RTX 3090 GPU. The hardware components that sustain the function of the central data processing assembly are installed in a mobile van for the purpose of this experimental process. The data acquisition sensors are capable of sending the collected data

to the central data processing components using a wired or wireless data connection. The experimental road deployments that are reported in this paper consider wireless connections through the 802.11 family of standards. The system may be easily modified in order to support wireless standards that offer a longer range. Additionally, it is relevant to note that in the case of permanent road deployments, it is possible to configure and deploy wired data connections between the data acquisition sensors and the central data processing components. The wired data connections may use either standard electrical conductors, or fiber-optic cables. It is important to note that the hardware features of the data transmission infrastructure can be easily and transparently modified relative to the deployed 3D objects detection system.

The experimental evaluation process also considered the number and placement of the implied sensors. This stage of the experimental process considers the assessment of the impact that the number, spatial position and orientation of the data acquisition sensors have on the actual detection of the 3D objects. This evaluation considers the early and late fusion schemes. The same structure of the algorithmic core is considered for the actual 3D objects detection. Considering the table that is displayed in **Figure 29**, the detection performance is measured through the actual accuracy of the detected 3D objects. The accuracy quantifies the number of precise 3D object detections relative to the total number of 3D objects that are part of the experiment.

No. Sensors	T-Junction EF	T-Junction LF	Roundabout EF	Roundabout LF
1	0.212	0.186	0.196	0.174
2	0.243	0.205	0.207	0.198
3	0.324	0.308	0.305	0.289
4	0.439	0.413	0.424	0.402
5	0.547	0.532	0.536	0.514
6	0.657	0.635	0.638	0.618
7	0.789	0.768	0.778	0.769
8	0.858	0.837	0.849	0.828
9	0.957	0.946	0.948	0.939
10	0.992	0.989	0.978	0.968
11	–	–	0.989	0.985
12	–	–	0.994	0.992

Figure 29. Comparative performance analysis considering various numbers of the active sensors.

The values of the performance metrics, which are presented in Figure 29, demonstrate that the accuracy of the 3D objects detection improves directly proportional to the numbers of active data acquisition sensors. It is interesting to note that, in the case of the T-junction topology, the optimal detection accuracy is reached with ten active data acquisition sensors, while the optimal detection accuracy relative to the roundabout topology is achieved when twelve active data acquisition sensors are considered. The experiments that were performed prove that the optimal level of the accuracy is greater or equal than 0.98. This implies that the 3D objects are detected without any significant issues, and the autonomous driving process occurs in an adequate manner. Additionally, more data acquisition sensors are required, if the detection accuracy level is necessary to be maintained on a larger area that is monitored. Let us recall that the monitored surface area is 4300 square meters in the case of the T-junction scenario, while the roundabout scenario covers an area with a surface of 12,200 square meters. Moreover, it can be observed that the early fusion scheme determines a superior level of the accuracy considering all the evaluated cases. This is a direct consequence of the early fusion scheme's ability to use more data during the preprocessing stage, as compared to the late fusion detection model.

Furthermore, the experimental workflow considers the spatial position and orientation of the data acquisition sensors. This is particularly important in the case of the T-junction scenario, which benefits from groups of three data acquisition sensors that are deployed in order to monitor certain sub-sections of the overall detection area. The next paragraph discusses on the experimental results that were obtained during the assessment of the spatial diversity's impact on the actual 3D objects detection accuracy.

The experiments that were conducted demonstrate that the problem of the data acquisition sensors' spatial diversity is also important. Thus, the prevention or, at least, minimization of the multihop data acquisition links from the sensors to the central data processing components is required in order to obtain a high level of the 3D object detection accuracy. Thus, we have determined that, in the

case of the T-junction scenario, a group of two sensors outperforms the single most efficient data acquisition sensor by 57%, while the optimal cluster of three sensors, which monitor precise sub-sections of the overall area, determine an improvement of 96%, relative to the most efficient data acquisition sensor. Furthermore, the same performance gains are 46% and 82%, respectively, in the case of the roundabout scenario. The percentage of the improvement is calculated relative to the base value of the accuracy, as it is determined by an individual data acquisition sensor considering the roundabout and T-junction scenarios. This demonstrates that the clusters of sensors should provide optimal overlap between their members. Consequently, this further demonstrates that the early fusion scheme may reduce the occurrence of falsely detected 3D objects. Additionally, clusters of sensors with properly overlapped members contribute to the increase of the 3D object detection accuracy.

The fundamental real-world significance of the proposed integrated model is also demonstrated through a comparative performance evaluation relative to three of the most computationally efficient fusion based models, which are the reference F-PointNet model [142], and also the MV3D [146] and AVOD [147] models. We used the same dataset that has been generated by the experimental field setup that has already been described. Furthermore, we implemented the F-PointNet, MV3D, and AVOD fusion based models, as it is suggested in reference articles [142], [146], and [147], respectively. Following this, the detection performance is computed considering the same road scenarios and number of sensors, which are presented in Figure 29. Let us recall that the detection performance is measured through the actual accuracy of the detected 3D objects. The accuracy quantifies the number of precise 3D object detections relative to the total number of 3D objects that are part of the experiment. Thus, let us study in **Figure 30** the detection performance that is determined by the MV3D model.

No. Sensors	T-Junction EF	T-Junction LF	Roundabout EF	Roundabout LF
1	0.122	0.114	0.119	0.108
2	0.173	0.178	0.185	0.142
3	0.215	0.256	0.277	0.203
4	0.289	0.356	0.352	0.346
5	0.398	0.431	0.468	0.424
6	0.508	0.513	0.528	0.505
7	0.623	0.597	0.599	0.594
8	0.698	0.684	0.675	0.649
9	0.789	0.768	0.759	0.702
10	0.841	0.869	0.854	0.789
11	–	–	0.869	0.828
12	–	–	0.904	0.893

Figure 30. Comparative performance analysis considering the MV3D model and various numbers of the active sensors.

Following, let us study in **Figure 31** the detection performance that is determined by the AVOD model.

No. Sensors	T-Junction EF	T-Junction LF	Roundabout EF	Roundabout LF
1	0.137	0.132	0.133	0.119
2	0.192	0.188	0.194	0.158
3	0.228	0.269	0.290	0.213
4	0.304	0.372	0.371	0.358
5	0.407	0.441	0.479	0.443
6	0.517	0.531	0.540	0.520
7	0.639	0.604	0.608	0.608
8	0.708	0.697	0.688	0.668
9	0.804	0.779	0.767	0.717
10	0.861	0.879	0.867	0.799
11	–	–	0.878	0.845
12	–	–	0.924	0.902

Figure 31. Comparative performance analysis considering the AVOD model and various numbers of the active sensors.

Finally, let us observe in **Figure 32** the detection performance that is determined by the F-PointNet model.

No. Sensors	T-Junction EF	T-Junction LF	Roundabout EF	Roundabout LF
1	0.151	0.141	0.148	0.134
2	0.199	0.206	0.217	0.163
3	0.241	0.293	0.305	0.231
4	0.316	0.386	0.391	0.368
5	0.416	0.469	0.490	0.452
6	0.540	0.542	0.551	0.539
7	0.641	0.628	0.610	0.608
8	0.716	0.709	0.692	0.668
9	0.831	0.789	0.776	0.730
10	0.878	0.898	0.881	0.804
11	–	–	0.899	0.846
12	–	–	0.926	0.907

Figure 32. Comparative performance analysis considering the AVOD model and various numbers of the active sensors.

The comparative performance data shows that there are marginal differences regarding the detection accuracy between the three fusion based reference models. Furthermore, this comparative experimental evaluation demonstrates that the integrated 3D object detection system that is reported in this paper, which is featured by an improved 3D object detection core, and an original architectural structure of its software components, determines a superior 3D object detection accuracy compared to all the reference fusion-based schemes.

The outcomes of the experimental evaluation process also demonstrated that the system is not substantially influenced by the weather conditions. Furthermore, it was observed that the system provides acceptable detection performance in the case when less detection sensors are deployed. Thus, it was demonstrated that the system can be considered as a sufficiently economical solution for the implementation of an effective autonomous driving approach. The detection accuracy remains high considering only six enrolled data acquisition sensors.

In conclusion, it can be stated that the system is able to accurately process the image samples and detect the 3D objects in virtually all of the cases. Additionally, it is important to note that the actual hardware deployment of the system considers existing technologies and data transmission protocols. This minimizes the implementation costs, while the data sharing from the central data processing components to the

individual cars in the monitored area ensures that even the vehicles that are not equipped with the latest technologies may benefit from the most efficient autonomous driving experience. The integrated system's 3D object detection accuracy is assessed against the reference fusion-based models MV3D, AVOD and F-PointNet. The results prove that the enhanced detection core and architecture of the described integrated system generates a higher level of performance relative to all three reference fusion-based models. The field validation setup that generated the experimental data that we considered, which has been deployed with the support of our industry partners, is already supporting the efforts of the relevant car manufacturers' research teams that aim to improve the current autonomous driving approaches.

B-ii. Plans for career development

Chapter 1. Didactic activity

Considering the early stages of my didactic activity, I have attentively monitored the evolution of my students' reference skills. The assessment activities considered the efficiency of related didactic activities, including an evaluation of the manifested students' interest towards, seminary, laboratory, and main course activities.

Although the Computer Science study programmes have constantly attracted intellectually gifted students, I have been able to notice a constant decrease of their involvement and motivation relative to the respective teaching activities. Therefore, starting with the academic year 2010-2011, I have adopted a novel didactic model, which has consistently proved its efficiency relative to the intended teaching activities.

The principles of the novel didactic model are fundamentally related to the experience that I gathered while I was an invited professor at the National University of Ireland, Cork, where I taught several Computer Sciences courses.

Thus, it is necessary for the students to receive the required number of theoretical concepts, which are mandatory for a correct understanding of the discussed problematic. The didactic process should be fundamentally interactive. This way, the students become an integral part of the teaching process, which further mediates the assimilation of relevant knowledge and skills.

The assessment of the acquired knowledge, skills, and competencies is realized through properly designed laboratory work packages. These should slightly overcome the work capacity of medium level students, which would stimulate the creation and development of relevant computer science thinking patterns.

It is relevant to note that although imposing deadlines regarding the submission of laboratory work packages is generally recommended, extensions may be granted in well justified cases. Additionally, creativity may be stimulated through the formulation of open laboratory work packages, as long as this approach is possible relative to the assessed scientific problematic.

The stimulation of the students' involvement and proactivity is valued at the highest possible level considering both the traditional and online didactic environments. Thus, I am able to assert, with a certain professional pride, that the consideration of the described didactic model has substantially determined an active online teaching environment during the period of COVID-19 pandemic. More precisely, no discernible decrease of the students' proactive involvement in the didactic process was apparent during the online teaching activities.

Considering the presented didactic strategy, it can be asserted that, on average, 95% of the students obtain passing grades relative to the proposed laboratory work packages. Moreover, more than 90% of the enrolled students obtain passing grades concerning the overall course marks, while more than 50% of the enrolled students obtain at least 7. It is relevant to note that the Romanian grading system relates to a scale going from 1 to 10, where 10 represents the highest mark, and 1 is the lowest mark. Moreover, the passing grade is 5.

Additionally, it is important to note that more than 50% of the enrolled students generally obtain marks that belong to the

range [7,…,10], although the evaluation of the respective laboratory work packages was rigorous.

It is very important to assert that students have generally asked for my supervision relative to their BSc and MSc graduation theses. Thus, on overage, I supervise approximately 50 BSc and MSc students each year. This is fundamentally an unusually high number of supervised BSc and MSc students. Additionally, the list of supervised BSc and MSc students may be consulted at the following web page: https://www.razvanbocu.bocu.ro/?page_id=78 .

Moreover, it was empirically demonstrated that the proposed didactic approach optimized the intellectual and practical real-world performance of talented students, while the less gifted students, who would not have been able to obtain 5, actually achieved a passing score. Therefore, this empirically validated didactic strategy will be continuously enhanced and optimized with the latest proper modalities and techniques, which would sustain the efficient intellectual and professional development of the respective computer science students.

Chapter 2. Prospective development of scientific career

Considering the presentation that was conducted in **section B-i**, the diversity of the research interests and activities, the amount of highly relevant outcomes, together with their scientific and real-world practical relevance, are evident. Nevertheless, there are numerous complementary activities, which further enrich the presented scientific activity. These are enumerated below.

- I am part of the reviewers' board of numerous highly ranked and prestigious Web of Science/Clarivate journals. Thus, a brief selection includes "Journal of Network and Computer Applications", "IEEE Transactions on Dependable and Secure Computing", "IEEE Access", "International Journal of Computers Communications & Control".
- I have submitted scientific research projects proposals, and consequently won the respective financing, in the realm of significant project calls organized by important organizations. Thus, I have won, through a strictly competitive procedure, the

financing that sustained my PhD research activity. The funder was the Government of Ireland. Moreover, I was awarded the financing for an advanced scientific research project under the auspices of NATO. The project addresses a fundamental topic concerning the continued assurance of data security, even relative to the advent of quantum computers. The identification of this project is "**NATO SPS G7394 - Post-quantum Digital Signature using Verkle Trees**".

- I have been a team member of numerous scientific research projects, which were awarded, for example, by the European Union, the Romanian Government, the Government of Georgia.
- During the period October 2007-October 2010, I have been an invited Professor at the National University of Ireland, Cork, in the Department of Computer Science. There, I conducted didactic activities relative to courses and laboratories/practical work packages. It is relevant to note that this university is placed among the first 200 best universities in the world.
- I founded and I have supervised the activity of the research group "High Performance and Cloud Computing", which functions under the auspices of Transilvania University of Brasov, Romania. Thus, a brief description of this scientific research group is provided on the official University website: HPCC Scientific Research Group .
- This research group has conducted an intense scientific activity considering the defined research scope, which produced papers published in peer reviewed Web of Science/Clarivate journals, and also in prestigious conferences that are indexed in CORE (https://portal.core.edu.au/conf-ranks/).
- I have been a member of several doctoral evaluation committees, which assessed the quality of the reported PhD research works, and related PhD theses. The last such participation was in Lithuania, in the Department of Computer Science at Kaunas Institute of Technology.
- I have proposed and supervised a special issue in the realm of the Clarivate journal "Symmetry" (https://www.mdpi.com/journal/symmetry).
- I have continuously cultivated the links with the relevant industry partners. Thus, since 2010, I have collaborated with General

Magic Brasov, Siemens Corporate Technology, In-Tech Engineering Services, and Siemens Industry Software. Considering the scope of these collaborations, I have conducted relevant scientific research projects, with outcomes that were considered by the relevant industry actors, while the results were published in peer reviewed journals. The relevant papers are enumerated, and some of them are presented in **section B-i**.

- Additionally, in cooperation with these partner companies, I organized summer schools and other training activities, which further improved the relevant skills of my Computer Science students.

The relevant professional career will be continuously developed considering all the significant perspectives. Thus, the empirically validated didactic strategy will be continuously enhanced and optimized with the latest proper modalities and techniques, which would sustain the efficient intellectual and professional development of the respective computer science students. The entire palette of complementary activities will be approached and continuously cultivated, with a special focus on the activity of the supervised research group, “High Performance and Cloud Computing”. Thus, there are interesting research projects, which this scientific research group currently handles that relate to advanced digital twins architectures based on microservices, machine learning algorithmic models, and related architectures, which address complex real-world use case scenarios, and also interdisciplinary research activities related to bioengineering and biomedical settings. In this context, it is absolutely relevant to draw the readers’ attention to the complex interdisciplinary contribution that is reported in paper [148], which interests the dynamic monitoring of time-dependent evolution of biomolecules using quantum dots-based biosensors assemblies. Essentially, the presented solution allows any interested researcher to study the time-dependent dynamics of target biomolecules on the surfaces of the analyzed cells through the generation of high resolution still images and motion pictures.

It is also relevant to note that the relatively diverse scientific research activity will be further developed relative to all the perspectives that were presented. Nevertheless, considering the research projects that will be managed, and also the activity conducted in the realm of the "High Performance and Cloud Computing" research group, it can be asserted that the following research subjects will be considered over the following three years: machine learning and artificial intelligence, advanced data privacy and security techniques, efficient encryption and digital signature models, which include post-quantum approaches. Additionally, the envisioned research efforts will also consider applications and architectures deployed over next generation infrastructures and data networks.

B-iii. Bibliography

[1]. R. Bocu. **Efficient Algorithms for Interactome Networks**. Cork University Press, Cork, Ireland, 2011.

[2]. T. Tsuruo, M. Naito, A. Tomida, N. Fujita, T. Mashima, H. Sakamoto and Naomi Haga. **Molecular targeting therapy of cancer: drug resistance, apoptosis and survival signal**. In Cancer Science, Volume 94, Issue 1, pp. 15-21, 2005.

[3]. D. Bocu, R. Bocu. **Introducere in Universul Ingineriei Softului**. ISBN 978-606-250-8166, Editura Matrix ROM, Bucuresti, 2023.

[4]. D. Bocu, R. Bocu. **Incursiuni speculative in si dincolo de Ingineria Softului**. ISBN 978-606-250-661-2, Editura Matrix ROM, Bucuresti, 2021.

[5]. D. Bocu, R. Bocu. **The Role of the WEB Technologies in Connection to the Communication's Streamlining and Diversification Between the Actors of a Learning System**. *Chapter published in the book "Social Media in Higher Education: Teaching in Web 2.0"***,** DOI: 10.4018/978-1-4666-2970-7.ch011, 2013*.*

[6]. D. Bocu, R. Bocu. **The ICS Paradigm in Knowledge and Modelling**. *Broad Research in Artificial Intelligence and Neuroscience ISSN 2067-3957, Volume 9, Issue 3, pp. 5-16,* 2018*.*

[7]. R. Bocu, D. Bocu. **The Role of the Conceptual Invariants Regarding the Prevention of the Software Artefacts' Obsolescence**. *Broad*

Research in Artificial Intelligence and Neuroscience ISSN 2067-3957, Volume 7, Number 4, pp. 56-62, 2016*.*

[8]. D. Bocu, R. Bocu. **THE FUNDAMENTALS REGARDING THE USAGE OF THE CONCEPT OF INTERFACE FOR THE MODELING OF THE SOFTWARE ARTEFACTS**. *Broad Research in Artificial Intelligence and Neuroscience ISSN 2067-3957, Volume 7, Number 1, pp. 91-102,* 2016*.*

[9]. D. Bocu, R. Bocu. **Remarks on Interface Oriented Software Systems Modelling**. *International Journal of Computers, Communications & Control ISSN 1841-9836, ISSN-L 1841-9836, Volume 8, Issue 5, October*, *pp. 662-672,* 2013*.*

[10]. D. Bocu, R. Bocu. **Strongly Project-Oriented Learning Systems. Concepts and Fundamental Principles**. *Scientific Studies and Research, Series Mathematics and Informatics, volume 21, no. 1, pp. 51-60*, 2011.

[11]. D. Bocu, R. Bocu. **Abstractization – A Fundamental Instrument for Describing and Modeling Software Systems**. *International Journal of u- and e- Service, Science and Technology, Vol. 4, No. 3, pp. 33-48*, 2011.

[12]. D. Bocu, R. Bocu. **Conceptual Foundations of Code Rationalization Through a Case Study in Haskell**. Proceedings of the Conference „Advanced Information Networking and Applications", 2022.

[13]. R. Bocu, M. Iavich. **Enhanced detection of low-rate DDoS attack patterns using machine learning models**. In *Journal of Network and Computer Applications*, volume 227, 103903. https://doi.org/10.1016/j.jnca.2024.103903, 2024.

[14]. C.L. Aldea, R. Bocu, R.N. Solca. **Real-Time Monitoring and Management of Hardware and Software Resources in Heterogeneous Computer Networks through an Integrated System Architecture**. *Symmetry* **2023**, *15*(6), 1134; https://doi.org/10.3390/sym15061134*, 2023.*

[15]. R. Bocu, M. Iavich. **Real-Time Intrusion Detection and Prevention System for 5G and beyond Software-Defined Networks**. *Symmetry* **2023**, *15*(1), 110; https://doi.org/10.3390/sym15010110*, 2022.*

[16]. M. Iavich, G. Akhalaia, R. Bocu. **Device Tracking Threats in 5G Network**. Proceedings of the Conference „Advanced Information Networking and Applications", 2023.

[17]. R. Bocu, M. Iavich, S. Tabirca. **A Real-Time Intrusion Detection System for Software Defined 5G Networks**. Proceedings of the Conference „Advanced Information Networking and Applications", 2021.
[18]. R. Bocu, M. Iavich, G. Iashvili, R. Odarchenko, S., Gnatyuk. **Intrusion Detection System for 5G with a Focus on DOS/DDOS Attacks**. Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021.
[19]. C. Costache, O. Machidon, A. Mladin, F. Sandu, R. Bocu, **Software-Defined Networking of Linux Containers**. *IEEE Computer Society RoEduNet Conference*, 2014.
[20]. R. Bocu, **IP Pooling-Based Email Systems Reputation Assurance**. *Proceedings of The RoEduNet 2011 International Conference, pp. 98-103,* 2011, ISSN: 2247-5443 / IEEE eISSN: 2068-1038 / Print ISBN: 978-1-4577-1233-3.
[21]. Z. Liu, X. Yin, Y. Hu. **CPSS LR-DDoS detection and defense in edge computing utilizing DCNN Q-learning**. IEEE Access, vol. 8, 8, 42120–42130, 2020.
[22]. Fortinet. **Fortinet Predicts Highly Destructive and Self-Learning 'Swarm' Cyberattacks in 2018**. [Online, accessed on 06 July 2023]. Available: https://www.fortinet.com/cn/corporate/about-us/newsroom/pressreleases/2017/predicts-self-learning-swarm-cyberattacks-2018 .
[23]. Z. Liu, X. Yin. **LSTM-CGAN: Towards generating low-rate DDoS adversarial samples for blockchain-based wireless network detection models**. arXiv:2210.02089, 9, 22616–22625, 2021.
[24]. A. Madane, M.-d. Dilmi, F. Forest, H. Azzag, M. Lebbah, J. Lacaille. **Transformer-based conditional generative adversarial network for multivariate time series generation**. arXiv:2210.02089 2022.
[25]. Fortinet. **Blacknurse ICMP DoS Attack**, 2023. [Online, accessed on 06 July 2023]. Available: https://www.fortiguard.com/psirt/FG-IR-16-091 .
[26]. S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, Y. Jararweh. **Federated learning review: Fundamentals, enabling technologies, and**

future applications. Information processing and management, 59(6), 103061, 2022.
[27]. A. Abeshu, N. Chilamkurti. **Deep learning: The frontier for distributed attack detection in fog-to-things computing**. IEEE Communications Magazine, vol. 56, no. 2, 169–175, 2018.
[28]. M. Yue, L. Liu, Z. Wu, M. Wang. **Identifying LDoS attack traffic based on wavelet energy spectrum and combined neural network**. International Journal of Communication Systems, vol. 31, no. 2, e3449, 2018.
[29]. H.J. Hadi, Y. Cao, K. U. Nisa, A. M. Jamil, Q. Ni. **A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs**. Journal of Network and Computer Applications, vol. 213: 103607, 2023.
[30]. M. Y. Akhlaqi, Z.B.M. Hanapi. **Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions**. Journal of Network and Computer Applications, vol. 212: 103568, 2023.
[31]. NS-3 Consortium. **The NS-3 discrete-event network simulator, 2023**. [Online, accessed on 06 July 2023]. Available: https://www.nsnam.org/ .
[32]. K. Hong, Y. Kim, H. Choi, J. Park. **SDN-assisted slow HTTP DDoS attack defense method**. IEEE Communications Letters, vol. 22, no. 4, 688–691, 2018.
[33]. M. Yue, L. Liu, Z. Wu, M. Wang. **Identifying LDoS attack traffic based on wavelet energy spectrum and combined neural network**. International Journal of Communication Systems, vol. 31, no. 2, e3449, 2018.
[34]. J. Charlier, A. Singh, G. Ormazabal, R. State, H. Schulzrinne. **SynGAN: Towards generating synthetic network attacks using GANs**. arXiv:1908.09899, 2019.
[35]. M.A. Khan, N. Iqbal, H. Jamil, D.H. Kim. **An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detections**. Journal of Network and Computer Applications, vol. 212: 103560, 2023.

[36]. A.R. Kunduru. **Security concerns and solutions for enterprise cloud computing applications**. Asian Journal of Research in Computer Science, vol. 15(4), 24–33, 2023.
[37]. H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A.Y. Arcas. **Communication-efficient learning of deep networks from decentralized data**. arXiv:1602.05629, 2016.
[38]. S.A. Rahman, H. Tout, C. Talhi, A. Mourad. **Internet of Things intrusion detection: Centralized, on-device, or federated learning?**. IEEE Networks, vol. 34, no. 6, 310–317, 2020.
[39]. T.A. Al-Amiedy, M. Anbar, B. Belaton, A.A. Bahashwan, I.H. Hasbullah, M.A. Aladaileh, G.A. Mukhaini. **A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things**. Internet of Things, vol. 22, no. 100741, 2023.
[40]. R. Wang, C. Ma, P. Wu. **An intrusion detection method based on federated learning and convolutional neural networks**. Netinfo Security, vol. 20, no. 4, 47–54, 2020.
[41]. B. Li, Y. Wu, J. Song, R. Lu, T. Li, L. Zhao. **DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems**. IEEE Transactions on Industrial Informatics, vol. 17, no. 8, 5615–5624, 2021.
[42]. P. Khordadpour, S. Ahmadi. **FIDS: Fuzzy Intrusion Detection System for simultaneous detection of DoS/DDoS attacks in Cloud computing**. arXiv:2305.16389, 2023.
[43]. W. Sun, S. Guan, P. Wang, Q. Wu. **A hybrid deep learning model based low-rate DoS attack detection method for software defined network**. Transactions on Emerging Telecommunications Technologies, vol. 33, no. 5, e4443, 2022.
[44]. M. A. Salahuddin, V. Pourahmadi, H. A. Alameddine, M. F. Bari, R. Boutaba. **DeepFed: Chronos: DDoS attack detection using time-based autoencoder**. IEEE Transactions on Network and Service Management, vol. 19, no. 1, 627–641, 2022.
[45]. X. Zhou, Y. Hu, W. Liang, J. Ma, Q. Jin. **Variational LSTM enhanced anomaly detection for industrial big data**. IEEE Transactions on Industrial Informatics, vol. 17, no. 5, 3469–3477, 2021.

[46]. R. Zhao, Y. Yin, Y. Shi, Z. Xue. **Intelligent intrusion detection based on federated learning aided long short-term memory**. Physical Communication, vol. 42, Art. no. 101157, 2020.
[47]. Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang. **Differentially private asynchronous federated learning for mobile edge computing in urban informatics**. IEEE Transactions on Industrial Informatics, vol. 16, no. 3, 2134–2143, 2020.
[48]. N. Ahuja, G. Singal, D. Mukhopadhyay, N. Kumar. **Automated DDOS attack detection in software defined networking**. Journal of Network and Computer Applications, vol. 187, 103108, 2021.
[49]. Y. Cui, Q. Qian, C. Guo, G. Shen, Y. Tian, H. Xing, L. Yan. **Towards DDoS detection mechanisms in software-defined networking**. Journal of Network and Computer Applications, vol. 190, 103156, 2021.
[50]. G. Brauwers, F. Frasincar. **A general survey on attention mechanisms in deep learning**. IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 4, 3279–3298, 2023.
[51]. S. Lin, Y. Zeng, Y. Gong. **Learning of time-frequency attention mechanism for automatic modulation recognition**. IEEE Wireless Communications Letters, vol. 11, no. 4, 707–711, 2022.
[52]. J. Miao, W. Zhu. **Precision–recall curve (PRC) classification trees**. IEEE Evolutionary intelligence, vol. 15, no. 3, 1545–1569, 2022.
[53]. F. Cheema, R. Urner. **Precision Recall Cover: A Method For Assessing Generative Models**. Proceedings of International Conference on Artificial Intelligence and Statistics, 6571–6594, 2023.
[54]. C.F.G.D. Santos, J.P. Papa. **Avoiding overfitting: A survey on regularization methods for convolutional neural networks**. ACM Computing Surveys (CSUR), vol. 54, no. 10, 1–25, 2022.
[55]. Y. Suhail, M. Upadhyay, A. Chhibber. **Machine learning for the diagnosis of orthodontic extractions: a computational analysis using ensemble learning**. Bioengineering, vol. 7, no. 2, 2020.
[56]. S.R. Dubey, S.K. Singh, B.B. Chaudhuri. **Activation functions in deep learning: A comprehensive survey and benchmark**. Neurocomputing, vol. 503, 92–108, 2022.
[57]. M. Ye, J. Shen, X. Zhang, P.C. Yuen, S.F. Chang. **Augmentation invariant and instance spreading feature for softmax embedding**. IEEE

Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 2, 924–939, 2020.
[58]. W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone, X. Liu. **A novel sigmoid-function-based adaptive weighted particle swarm optimizer**. IEEE Transactions on Cybernetics, vol. 51, no. 2, 1085–1093, 2019.
[59]. P.A. Papp, K. Martinkus, L. Faber, R. Wattenhofer. **DropGNN: Random dropouts increase the expressiveness of graph neural networks**. Advances in Neural Information Processing Systems, vol. 34, 21997–22009, 2021.
[60]. S. Jadon. **A survey of loss functions for semantic segmentation**. Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 1–7, 2020.
[61]. P. Netrapalli. **Stochastic gradient descent and its variants in machine learning**. Journal of the Indian Institute of Science, vol. 99, no. 2, 201–213, 2019.
[62]. C.T. Chen, G.X. Gu. **Generative deep neural networks for inverse materials design using backpropagation and active learning**. Advanced Science, vol. 7, no. 5, 1902607, 2020.
[63]. Y. Zhou, M. Shi, Y. Tian, Q. Ye, J. Lv. **DeFTA: A Plug-and-Play Decentralized Replacement for FedAvg**. arXiv:2204.02632, 2022.
[64]. HashiCorp. **The Vagrant Virtualization Platform**. [Online, accessed on 16 July 2023]. Available: https://developer.hashicorp.com/vagrant/ .
[65]. E. Gordon-Rodriguez, G. Loaiza-Ganem, G. Pleiss, J.P. Cunningham. **Uses and abuses of the cross-entropy loss: Case studies in modern deep learning**. Proceedings of Machine Learning Research, vol. 137, 1–10, 2020.
[66]. D.M. Belete, M.D. Huchaiah. **Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results**. International Journal of Computers and Applications, vol. 44, no. 9, 875–886, 2022.
[67]. M.A. Khan, N. Kumar, S.A.H. Mohsan, W.U. Khan, M.M. Nasralla, M.H. Alsharif, J. Zywiolek, I. Ullah. **Swarm of UAVs for Network Management in 6G: A Technical Review**. IEEE Trans. Netw. Serv. Manag. 2022, 35, 9.
[68]. V. Ravi, R. Chaganti, M. Alazab. **Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent**

network intrusion detection system. Comput. Electr. Eng. 2022, 102(C), 108156.
[69]. D. May, A. Landwehr, T. Browning, C. Cotton, F. Kiamilev. Next Generation Data Link for IRSP Systems. In **Proceedings of the 2021 IEEE Research and Applications of Photonics in Defense Conference (RAPID)**, Miramar Beach, FL, USA, 2–4 August 2021; pp. 1–2.
[70]. J. Santos, T. Wauters, B. Volckaert, F. De Turck. **Towards Low-Latency Service Delivery in a Continuum of Virtual Resources: State-of-the-Art and Research Directions**. IEEE Commun. Surv. Tutorials 2021, 23, 2557–2589.
[71]. G.A. Akpakwu, B.J. Silva, G.P. Hancke, A.M. Abu-Mahfouz. **A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges**. IEEE Access 2017, 6, 3619–3647.
[72]. I. Parvez, A. Rahmati, I. Guvenc, A. Sarvat, H. Dai. **A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions**. IEEE Commun. Surv. Tutor. 2018, 20, 3098–3130.
[73]. M. Iavich, R. Bocu, A. Gagnidze. **Real Time Self-developing Cybersecurity Function for 5G**. In Advanced Information Networking and Applications. AINA 2022; Barolli, L., Hussain, F., Enokido, T., Eds.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2022; Volume 451.
[74]. S. Sekander, H. Tabassum, E. Hossain. **Multi-Tier Drone Architecture for 5G/B5G Cellular Networks: Challenges, Trends, and Prospects**. IEEE Communications Magazine, Volume 56, Issue 3, 96–103, 2018.
[75]. R. Bocu, A. Baicoianu, A. Kerestely, **An Extended Survey Concerning the Significance of Artificial Intelligence and Machine Learning Techniques for Bug Triage and Management**. *IEEE Access*, vol. 11, pp. 123924-123937, doi: 10.1109/ACCESS.2023.3329732, 2023.
[76]. R. Bocu, D., Bocu, M. Iavich. **An Extended Review Concerning the Relevance of Deep Learning and Privacy Techniques for Data-Driven Soft Sensors**. *Sensors* 2023, *23*(1), 294; https://doi.org/10.3390/s23010294*,* 2022*.*
[77]. R. Bocu, D. Bocu, M. Iavich. **Objects Detection Using Sensors Data Fusion in Autonomous Driving Scenarios**. *Electronics 2021, 10, 2903. https://doi.org/10.3390/electronics10232903,* 2021*.*

[78]. R. Bocu, M. Iavich. **Enhanced Autonomous Driving Through Improved 3D Objects Detection**. Proceedings of the Conference „Advanced Information Networking and Applications", 2022.
[79]. R. Bocu, A. Kerestely, A. Baicoianu. **A Research Study on Running Machine Learning Algorithms on Big Data with Spark**. Proceedings of the 14th International Conference on Knowledge Science, Engineering and Management (KSEM), 2021.
[80]. R. Bocu, M. Iavich, S. Gnatyuk, R. Odarchenko, S. Simonov. **The novel system of attacks detection in 5G**. Proceedings of the Conference „Advanced Information Networking and Applications", 2021.
[81]. M. Iavich, R. Bocu, S. Gnatyuk, G. Iashvili. **Novel Method of Hardware Security Problems Identification**. Proceedings of the International Conference „Problems of Infocommunications. Science and Technology", Kharkiv, 2020.
[82]. University, C.W.R. **The case western reserve university bearing data center website**. [Online, accessed on 17 July 2024]. Available: https://csegroups.case.edu/bearingdatacenter .
[83]. S. Zhang, S. Zhang, B. Wang, T.G. Habetler. **Machine learning and deep learning algorithms for bearing fault diagnostics-a comprehensive review**. arXiv preprints: arXiv:1901.08247, 2019.
[84]. C. Freitas, J. Cuenca, P. Morais, A. Ompusunggu, M. Sarrazin, K. Janssens. **Comparison of vibration and acoustic measurements for detection of bearing defects**. In: International Conference on Noise and Vibration Engineering 2016 and International Conference on Uncertainty in Structural Dynamics 2016. vol. 1, 2016.
[85]. M. Driscoll. **Winning with big data: Secrets of the successful data scientist**. [Online, accessed on 17 July 2024]. Available: https://conferences.oreilly.com/datascience/public/schedule/detail/15316 , 2010.
[86]. Databricks. **Parquet files**. [Online, accessed on 17 July 2024]. Available: https://docs.databricks.com/data/data-sources/read-parquet.html , 2024.
[87]. J. Nowacki. **Text analysis in pandas**. [Online, accessed on 17 July 2024]. Available: https://sigdelta.com/blog/text-analysis-in-pandas/ , 2024.

[88]. Apache Spark. **Pyspark usage guide for pandas with Apache Arrow**. [Online, accessed on 17 July 2024]. Available: https://spark.apache.org/docs , 2024.
[89]. A. Cachuan. **A gentle introduction to Apache Arrow with Apache Spark and pandas**. [Online, accessed on 17 July 2024]. Available: https://towardsdatascience.com , 2024.
[90]. R. Pedapatnam. **Understanding resource allocation configurations for a Spark application**. [Online, accessed on 17 July 2024]. Available: http://site.clairvoyantsoft.com/ , 2016.
[91]. C. Davis. **Big data on a laptop: Tools and strategies**. [Online, accessed on 17 July 2024]. Available: https://tech.popdata.org , 2018.
[92]. M. Iavich, T. Kuchukhidze, R. Bocu. **A Post-Quantum Digital Signature Using Verkle Trees and Lattices**. *Symmetry*. 2023; 15(12):2165, https://doi.org/10.3390/sym15122165, 2023.
[93]. L. Gorlov, M. Iavich, R. Bocu. **Linear Layer Architecture Based on Cyclic Shift and XOR**. *Symmetry*. 2023; 15(8):1496; https://doi.org/10.3390/sym15081496, 2023.
[94]. M. Iavich, R. Bocu, T. Kuchukhidze, G. Iashvili, S. Gnatyuk. **Novel Quantum Random Number Generator with the Improved Certification Method**. *International Journal of Mathematical Sciences and Computing ISSN 2310-9025, Volume 7, Number 3, pp. 41-53,* 2021.
[95]. M. Iavich, T. Kuchukhidze, R. Bocu. **A Post-quantum Cryptosystem with a Hybrid Quantum Random Number Generator**. Proceedings of the Conference „Advanced Information Networking and Applications", 2023.
[96]. R. Bocu, M. Iavich, T. Kuchukhidze, R. Odarchenko. **The novel hybrid method for the randomness extraction**. Proceedings of „International Conference on Next Generation Cybersecurity Systems and Applications" (NGSEC), Kiev, 2022.
[97]. R. Bocu, M. Iavich, G. Iashvili, R. Odarchenko. **A Post-Quantum Secure e-Health System for the Data Management**. Proceedings of the IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT), 2021.
[98]. M. Iavich, R. Bocu, G. Iashvili. **Post-Quantum Digital Signature Scheme for Personal Data Security in Communication Network**

Systems. Proceedings of the International Conference of Artificial Intelligence, Medical Engineering, Education, Moscow, 2020.
[99]. M. Iavich, R. Bocu, A. Arakelian, G. Iashvili. **Post-quantum Digital Signatures with Attenuated Pulse Generator**. Proceedings of the International Conference on Information Technology, Kaunas, 2020.
[100]. L. Chen, L. Chen, S. Jordan, Y.K. Liu, D. Moody, R. Peralta, R.A. Perlner, D. Smith-Tone. **Report on Post-Quantum Cryptography**. US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, Volume 12, 2016.
[101]. J. Buchmann, E. Dahmen, M. Szydlo. **Hash-based Digital Signature Schemes**. In Post-Quantum Cryptography; Bernstein, D.J., Buchmann, J., Dahmen, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2009.
[102]. B. Bhaskar, N. Sendrier. **McEliece cryptosystem implementation: Theory and practice.** In Post-Quantum Cryptography, Proceedings of the Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, 17–19 October 2008; Proceedings 2; Springer: Berlin/Heidelberg, Germany, 2008.
[103]. X. Yin, J. He, Y. Guo, D. Han, K.-C. Li, A. Castiglione. **An Efficient Two-Factor Authentication Scheme Based on the Merkle Tree**. Sensors 20, 5735, 2020.
[104]. Y.-C. Chen, Y.-P. Chou, Y.-C. Chou. **An Image Authentication Scheme Using Merkle Tree Mechanisms**. Future Internet, 11, 149, 2019.
[105]. L. Lamport. **Constructing Digital Signatures from a One Way Function**. [Online, accessed on 18 July 2024]. Available: https://www.microsoft.com/enus/research/publication/constructing-digital-signatures-one-way-function/ , 1979.
[106]. M. Iavich, R. Bocu, A. Arakelian, G. Iashvili. **Post-Quantum Digital Signatures with Attenuated Pulse Generator**. Volume 2698. 2020. Available online: https://www.researchgate.net/profile/Maksim-Iavich/publication/346971219_Post-Quantum_Digital_Signatures_with_Attenuated_Pulse_Generator/links/5fd63e2845851553a0b26923/Post-Quantum-Digital-Signatureswith-Attenuated-Pulse-Generator.pdf (accessed on 18 July 2024).

[107]. D. Koo, Y. Shin, J. Yun, J. Hur. **Improving Security and Reliability in Merkle Tree-Based Online Data Authentication with Leakage Resilience**. Appl. Sci., 8, 2532, 2018.
[108]. M. Sim, S. Eum, G. Song, Y. Yang, W. Kim, H. Seo. **K-XMSS and K-SPHINCS+: Enhancing Security in Next-Generation Mobile Communication and Internet Systems with Hash Based Signatures Using Korean Cryptography Algorithms**. Sensors, 23, 7558, 2023.
[109]. R.C. Merkle. **A Digital Signature Based on a Conventional Encryption Function**. In Advances in Cryptology—CRYPTO ’87. CRYPTO 1987; Pomerance, C., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, Volume 293, 1988.
[110]. H. Chen, D. Liang. **Adaptive Spatio-Temporal Query Strategies in Blockchain**. ISPRS Int. J. Geo-Inf. 11, 409, 2022.
[111]. W. Wang, A. Ulichney, C. Papamanthou. **BalanceProofs: Maintainable vector commitments with fast aggregation**. In Proceedings of the 32nd USENIX Conference on Security Symposium (SEC ‘23), Berkeley, CA, USA, 9–11 August 2023; USENIX Association: Berkeley, CA, USA. Article 247. pp. 4409–4426, 2023.
[112]. Kurosawa; Kaoru; Hanaoka, G. (Eds.). **Public-Key Cryptography PKC 2013**. In Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, 26 Feburary–1 March 2013; Springer: Berlin/Heidelberg, Germany; Volume 7778, 2013.
[113]. K. John. **Verkle Trees**. [Online, accessed on 18 July 2024]. Available: https://math.mit.edu/research/highschool/primes/materials/2018/Kuszmaul.pdf , 2018.
[114]. C. Papamanthou, E. Shi, R. Tamassia, K. Yi. **Streaming authenticated data structures**. In Advances in Cryptology–EUROCRYPT 2013, Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, 26–30 May 2013; Proceedings 32; Springer: Berlin/Heidelberg, Germany, 2013.
[115]. M. Iavich, S. Gnatyuk, A. Arakelian, G. Iashvili, Y. Polishchuk, D. Prysiazhnyy. **Improved Post-quantum Merkle Algorithm Based on Threads**. In Advances in Computer Science for Engineering and

Education III 3; Springer International Publishing: Berlin/Heidelberg, Germany; pp. 454–464, 2021.
[116]. R. Bocu, A. Vasilescu, D.M. Duca Iliescu. **Personal Health Metrics Data Management Using Symmetric 5G Data Channels**. *Symmetry,* 14(7):1387. https://doi.org/10.3390/sym14071387*,* 2022*.*
[117]. R. Bocu, C. Costache. **A Homomorphic Encryption-Based System for Securely Managing Personal Health Metrics Data**. *IBM Journal of Research and Development ISSN 0018-8646, Volume 62, Issue 1*, *pp. 1:1-1:10*, 2018*.*
[118]. R. Bocu, M. Iavich, S. Gnatyuk, D. Ospanova, Y. Sotnichenko. **Secure e-Health System for the Integrated Management of Personal Health Data Collected by IoT Devices**. Proceedings of the International Conference „Cybersecurity Providing in Information and Telecommunication Systems", Kiev, 2021.
[119]. R. Bocu. **A Secure Distributed e-Health System for the Management of Personal Health Metrics Data**. Proceedings of the Conference „Advanced Information Networking and Applications", 2020.
[120]. C. Gentry. **A Fully Homomorphic Encryption Scheme**. Stanford, CA, USA: Stanford Univ., 2009.
[121]. IBM Corporation. **IBM Cloud Infrastructure**. [Online, accessed on 22 July 2024]. Available: https://www.ibm.com/cloud , 2024.
[122]. IBM Corporation. **IBM Cloudant Storage Service**. [Online, accessed on 22 July 2024]. Available: https://www.ibm.com/products/cloudant , 2024.
[123]. Apache Software Foundation. **The OpenWhisk Programming Service**. [Online, accessed on 22 July 2024]. Available: https://openwhisk.apache.org/ , 2024.
[124]. Z. Brakerski, C. Gentry, V. Vaikuntanathan. **Fully homomorphic encryption without bootstrapping**. In Proc. Innov. Theor. Comput. Sci. Conf., pp. 309–325, 2012.
[125]. Z. Brakerski, V. Vaikuntanathan. **Efficient fully homomorphic encryption from (standard) LWE**. In Proc. Annu. Symp. Found. Comput. Sci., pp. 97–106, 2011.

[126]. S. A. Immanuel, A. Sadrieh, M. Baumert, et al. **T-wave morphology can distinguish healthy controls from LQTS patients**. Physiol. Meas., vol. 37, no. 9, pp. 1456–1473, 2016.
[127]. Polar. **Polar H7 Heart Rate Sensor**. [Online, accessed on 22 July 2024].Available:https://support.polar.com/e_manuals/H7_Heart_Rate_Sensor/Polar_H7_Heart_Rate_Sensor_accessory_manual_English.pdf , 2024.
[128]. Polar. **Polar H10 Heart Rate Sensor**. [Online, accessed on 22 July 2024]. Available: https://www.polar.com/en/sensors/h10-heart-rate-sensor , 2024.
[129]. C.L. Aldea, R. Bocu, A. Vasilescu. **Relevant Cybersecurity Aspects of IoT Microservices Architectures Deployed over Next-Generation Mobile Networks**. *Sensors 23*(1), 189; https://doi.org/10.3390/s23010189, 2022.
[130]. C.L. Aldea, R. Bocu, D.M. Duca Iliescu. **Health Parameters Monitoring Through an Integrated Multilayer Digital Twin Architecture**. Proceedings of the Conference „Advanced Information Networking and Applications", Springer, Cham. https://doi.org/10.1007/978-3-031-57840-3_27 , 2024.
[131]. R. Bocu, D. Bocu. **Next Generation Mobile Sensors: Review Regarding the Significance of Deep Learning and Privacy Techniques for Data-Driven Soft Sensors**. Proceedings of the Conference „Advanced Information Networking and Applications", 2023.
[132]. R. Bocu. **A Constructive Review Regarding the Significance of 5G Networks for the Internet of Things**. Proceedings of the Conference „Advanced Information Networking and Applications", 2020.
[133]. F. Febrero, C. Calero, M.A. Moraga. **Software reliability modeling based on ISO/IEC SQuaRE**. Inf. Softw. Technol., 70, 18–29, 2016.
[134]. T. Dybå, T. Dingsøyr. **Empirical studies of agile software development: A systematic review**. Inf. Softw. Technol., 50, 833–859, 2008.
[135]. H. Zhang, M.A. Babar, P. Tell. **Identifying relevant studies in software engineering**. Inf. Softw. Technol., 53, 625–637, 2011.
[136]. J. Soldani, D.A. Tamburri, W.J. Van Den Heuvel. **The pains and gains of microservices: A systematic grey literature review**. J. Syst. Softw., 146, 215–232, 2018.

[137]. T. Champaneria, S. Jardosh, A. Makwana. **Microservices in IoT Middleware Architectures: Architecture, Trends, and Challenges**. In IoT with Smart Systems. Smart Innovation, Systems and Technologies; Springer: Singapore, pp. 381–395, 2022.
[138]. M.A. Jarwar, M.G. Kibria, S. Ali, I. Chong. **Microservices in Web Objects Enabled IoT Environment for Enhancing Reusability**. Sensors, 18, 352, 2018.
[139]. U. Inayat, M.F. Zia, S. Mahmood, H.M. Khalid, M. Benbouzid. **Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis, and Future Prospects**. Electronics 11, 1502, 2022.
[140]. U. Inayat, M.F. Zia, S. Mahmood, T. Berghout, M. Benbouzid. **Cybersecurity Enhancement of Smart Grid: Attacks, Methods, and Prospects**. Electronics, 11, 3854, 2022.
[141]. A. Geiger, P. Lenz, R. Urtasun. **Are We Ready for Autonomous Driving? The KITTI vision benchmark suite**. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; Volume 2012; pp. 3354–3361, 2012.
[142]. C.R. Qi, W. Liu, C. Wu, H. Su, L.J. Guibas. **Frustum Pointnets for 3D object detection from RGB-D data**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918-927, 2018.
[143]. P. Ghamisi, B. Rasti, N. Yokoya, Q. Wang, B. Hofle, L. Bruzzone, F. Bovolo, M. Chi, K. Anders, R. Gloaguen. **Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art**. IEEE Geosci. Remote Sens. Mag. 7, 6–39, 2019.
[144]. S. Shi, X. Wang, H. Li. **Pointrcnn: 3D object proposal generation and detection from point cloud**. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June, 2019.
[145]. F. Castanedo. **A review of data fusion techniques**. Sci. World J., 142–149, 2013.
[146]. X. Chen, H. Ma, J. Wan, B. Li, T. Xia. **Multi-View 3D Object Detection Network for Autonomous Driving**. In Proceedings of the

2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
[147]. J. Ku, M. Mozifian, J. Lee, A. Harakeh, S. Waslander. **Joint 3D proposal generation and object detection from view aggregation**. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
[148]. R. Bocu. **Dynamic Monitoring of Time-Dependent Evolution of Biomolecules Using Quantum Dots-Based Biosensors Assemblies**. Biosensors **2024**, 14, 380. https://doi.org/10.3390/bios14080380, 2024.