



Comparative Study of Name Entity Recognition Models in Burkina Faso Context

Sibiri Tiemounou¹(✉), Wend Yam Serge Boris Ouédraogo^{1,2}, Moumouni Djibo², Yaya Traoré³, Ali Maïga⁴, Souleymane Zio¹, and François Zougmore²

¹ Institut du Génie Informatique et Télécommunications, École Polytechnique de Ouagadougou, Ouagadougou, Burkina Faso

sibiri.tiemounou@gmail.com

² Laboratoire Matériaux et Environnement, Université Joseph KI-ZERBO, Ouagadougou, Burkina Faso

³ Laboratoire de Mathématiques et Informatique, Université Joseph KI-ZERBO, Ouagadougou, Burkina Faso

⁴ FDI MATELEC, Paris, France

Abstract. Name Entity Recognition (NER) is an important core component of Natural Language Processing (NLP) systems for identifying entities like person names, locations, and organizations. Many NER models have been proposed in the literature those whose architecture are based on deep neural networks are the most efficient. Burkina Faso, a West African country, is a French-speaking country with its culture and its specificities in the use of the French language. Our work consists in building a user-friendly NER model that reliably identifies 8 entity types from Burkina Faso media information where the French language is the most spoken. To achieve this goal, we firstly build a dataset that has been labeled along these entity types. Then, in order to choose the best architecture, we assessed 6 multilanguage NER models namely Spacy (with its three pre-trained models), Flair, Stanza, and Camembert NER. This paper presents the performance evaluation of existing French NER models when applying to media news data of Burkina Faso. They have been assessed along 3 common entity types Person (PER), LOcation (LOC), and ORGanization (ORG). Results show that Stanza and Flair outperform all models under study with a percentage greater than 70%. They reliably identify a person's name and location entities. However, their performance is relatively fair in correctly extracting organization entities due to the Burkina Faso context.

Keywords: Name Entity Recognition · Natural language processing · Neural Network · NER models

1 Introduction

Name Entity Recognition (NER) is a subfield of Natural Language Processing (NLP) that consists in identifying named entities like person, location, organization, etc. based on text analysis. It stands for the core components of many NLP applications such as

14. Olenik, S., Lee, H.S., Güder, F.: The future of near-field communication-based wireless sensing. *Nat. Rev. Mater.* **6**, 286–288 (2021)
15. Hossain, M.I., Markendahl, J.I.: Comparison of LPWAN Technologies: Cost Structure and Scalability. *Wireless Pers. Commun.* **121**, 887–903 (2021)
16. Dangi, R., Lalwani, P., Choudhary, G., You, I., Pau, G.: Study and Investigation on 5G Technology: A Systematic Review. *Sensors* **22**(1), 26 (2022)
17. Akasiadis, C., Pitsilis, V., Spyropoulos, C.D.: A Multi-Protocol IoT Platform Based on Open-Source Frameworks. *Sensors* **19**(19), 4217 (2019)
18. Tariq, M.A., Khan, M., Raza Khan, M.T., Kim, D.: Enhancements and Challenges in CoAP—A Survey. *Sensors* **20**(21), 6391 (2020)
19. Biswajeeban, M., Attila, K.: The Use of MQTT in M2M and IoT Systems: A Survey. *IEEE Access* **8**, 201071–201086 (2020)
20. Ioana, A., Korodi, A.: DDS and OPC UA Protocol Coexistence Solution in Real-Time and Industry 4.0 Context Using Non-Ideal Infrastructure. *Sensors* **21**(22), 7760 (2021)
21. Li, F., Wang, C.: Artificial intelligence and edge computing for teaching quality evaluation based on 5G-enabled wireless communication technology. *J. Cloud Comput.* **12**(1), 45 (2023)
22. Merenda, M., Porcaro, C., Iero, D.: Edge Machine Learning for AI-Enabled IoT Devices: A Review. *Sensors* **20**(9), 2533 (2020)
23. Xu, Z., Liu, W., Huang, J., Yang, C., Lu, J., Tan, H.: Artificial intelligence for securing IoT services in edge computing: a survey. *Security and communication networks* **2020**, 1–13 (2020)
24. Adam, B., Kaveh, M.: The rise of artificial intelligence in healthcare applications. In: Adam, B., Kaveh, M. (eds.) *Artificial Intelligence in Healthcare*, pp 25–60. Academic Press, Cambridge (2020)
25. Bajwa, J., Munir, U., Nori, A., Williams, B.: Artificial intelligence in healthcare: transforming the practice of medicine. *Future healthcare journal* **8**(2), 188–194 (2021)
26. Kamruzzaman, M.M., Alrashdi, I., Alqazzaz, A.: New opportunities, challenges, and applications of Edge-AI for connected healthcare in Internet of Medical Things for smart cities. *J. Healthcare Eng.* **2022**(1), 2950699 (2022)
27. Motti, V.G., Berkovsky, S.: Healthcare Privacy. In: Knijnenburg, B.P., Page, X., Wisniewski, P., Lipford, H.R., Proferes, N., Romano, J. (eds.) *Modern Socio-Technical Perspectives on Privacy*, pp 203–231. Springer, Cham (2022)
28. Wang, C., Zhang, J., Lassi, N., Zhang, X.: Privacy protection in using artificial intelligence for healthcare: Chinese regulation in comparative perspective. In: *Healthcare*, vol. 10, no. 10, p. 1878. MDPI (2022)
29. Monteiro, A., Santos, S., Gonçalves, P.: Precision agriculture for crop and livestock farming—brief review. *Animals* **11**(8), 2345 (2021)
30. Debauche, O., Mahmoudi, S., Elmoulat, M., Mahmoudi, S.A., Manneback, P., Le-beau, F.: Edge AI-IoT pivot irrigation, plant diseases, and pests identification. *Procedia Computer Science* **177**, 40–48 (2020)
31. Neethirajan, S.: The role of sensors, big data and machine learning in modern animal farming. *Sensing and Bio-Sensing Research* **29**, 100367 (2020)
32. Sun, L., Jiang, X., Ren, H., Guo, Y.: Edge-cloud computing and artificial intelligence in internet of medical things: architecture, technology and application. *IEEE Access* **8**, 101079–101092 (2020)

6 Conclusion

In the ever-evolving landscape of technology, the fusion of Edge AI and Internet of Things (IoT) has emerged as a transformative force, revolutionizing the way intelligent systems are designed, deployed, and experienced. Throughout this research paper, we have delved into the intricate realms of these cutting-edge technologies, exploring their architectures, applications, and future perspectives.

In conclusion, the synergy between Edge AI and IoT is reshaping the landscape of intelligent systems, bringing us closer to a future where machines are not just smart but also contextually aware and responsive. Embracing the full potential of these technologies demands continuous research, innovation, and collaboration.

The fusion of Edge AI and IoT has given rise to sophisticated architectures that bridge the gap between the physical and digital worlds. Edge computing, with its decentralized approach, enables real-time data processing and analysis at the edge devices, reducing latency and enhancing efficiency. Incorporating Edge AI technologies into agriculture holds the promise of transforming the industry into a more sustainable, efficient, and resilient sector.

References

1. Rahmani, A.M., Bayramov, S., Kiani Kalejahi, B.: Internet of things applications: opportunities and threats. *Wireless Pers. Commun.* **122**(1), 451–476 (2022)
2. Pérez, J., Díaz, J., Berrocal, J., López-Viana, R., González-Prieto, Á.: Edge computing: A grounded theory study. *Computing* **104**(12), 2711–2747 (2022)
3. Sheikh, H., Prins, C., Schrijvers, E.: Artificial Intelligence: Definition and Background. In: Sheikh, H., Prins, C., Schrijvers, E. (eds.) *Mission AI: The New System Technology*, pp. 15–41. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-21448-6_2
4. Singh, R., Gill, S.S.: Edge AI: a survey. *Internet of Things Cyber-Phys. Syst.* **3**, 71–92 (2023)
5. Yu, W., et al.: A survey on the edge computing for the Internet of Things. *IEEE Access* **6**, 6900–6919 (2017)
6. Kang, K.D.: A Review of Efficient Real-Time Decision Making in the Internet of Things. *Technologies* **10**(1), 12 (2022)
7. Hamdan, S., Ayyash, M., Almajali, S.: Edge-computing architectures for Internet of Things applications: a survey. *Sensors* **20**(22), 6441 (2020)
8. Lina, L., Ruisheng, S., Bai, W., Lei, Z., Ning, J.: A Universal Complex Event Processing Mechanism Based on Edge Computing for Internet of Things Real-Time Monitoring. *IEEE Access* **7**, 101865–101878 (2019)
9. IEEE Innovation at work: <https://innovationatwork.ieee.org/real-life-edge-computing-use-cases/>, last accessed 2023/10/13
10. Filali, A., Abouaomar, A., Cherkaoui, S., Kobbane, A., Guizani, M.: Multi-access-edge computing: a survey. *IEEE Access* **8**, 197017–197046 (2020)
11. Haque, K.F., Ahmed, A., Kumar, Y.: Comprehensive Performance Analysis of Zigbee Communication: An Experimental Approach with XBee S2C Module. *Sensors* **22**(9), 3245 (2022)
12. Chendong, L., Yilin, Z., Huanyu, Z.: A Comprehensive Study of Bluetooth Low Energy. *J. Phys: Conf. Ser.* **2093**(1), 012021 (2021)
13. Pahlavan, K., Krishnamurthy, P.: Evolution and Impact of Wi-Fi Technology and Applications: A Historical Perspective. *Int. J. Wireless Inf. Networks* **28**, 3–19 (2021)

Overall, Edge-IA and IoT will revolutionize the agricultural sector. By enabling real-time agricultural data processing and decision-making at the source of collection, Edge-IA and IoT will help optimize crop yields, improve livestock management and promote sustainable farming practices. With a growing global population to feed, Edge-IA and IoT will be key to meeting agricultural challenges and ensuring a sustainable future for all.

5 Challenges and Future Direction

Edge computing is made up of peer-to-peer systems, wireless networks, etc. and the adoption of a complete system is required to manage the system as a whole. The security and confidentiality of a distributed system is a real challenge.

During data processing at the edge, user data can be processed, putting their sensitive information at risk. Authentication of gateway nodes is one of the security issues in edge computing. Each edge node is managed differently. It is therefore difficult to implement a common security method everywhere.

Another challenge is the security of data sharing and transmission processes. Multiple devices need to work together to perform many tasks, and data needs to be shared securely. Data storage is provided by various third-party suppliers, and their storage devices spread over several sites also increase the risk of attack, for two reasons. Firstly, the data is separated into several slices and stored in different places. It is therefore easy to store incorrect data or to lose it. This makes it difficult to guarantee data integrity. Secondly, data downloaded to storage can be modified by malicious users, leading to confidentiality problems and data leaks.

It would be preferable to use local pre-processing, which could mask private information and reduce the amount of data transmitted.

In some cases, encryption could be used to protect privacy. However, before learning or inference tasks can be performed, the encrypted data must be decrypted. This requires an increase in the amount of essential computation. To meet this challenge, the authors [32] propose the use of homomorphic encryption, which allows the training or inference task to be performed directly on encrypted data.

Looking forward, the future of Edge AI and IoT appears incredibly promising. The rapid advancements in hardware, such as powerful yet energy-efficient microprocessors, are driving the development of more complex AI models that can be deployed at the edge. Additionally, the proliferation of 5G networks will significantly enhance the capabilities of Edge AI and IoT, enabling seamless connectivity and faster data transmission.

Moving on the future, it is crucial to address the challenges associated with these technologies, such as security vulnerabilities, data privacy concerns, and ethical considerations. Collaboration between academia, industry, and policymakers will be vital in creating robust frameworks and standards that ensure the responsible development and deployment of Edge AI and IoT solutions.

As the amount of data generated by healthcare facilities increases, so does the risk of data breaches and unauthorized access. Edge AI is emerging as a potential solution to this problem, offering the potential to guarantee data confidentiality and security [26].

Edge intelligence is ideal for healthcare facilities because it enables data to be analyzed without leaving the facility's secure network. With data maintained on site, the risk of data breaches and unauthorized access can be reduced. In addition, it enables healthcare organizations to ensure that they comply with the requirements of various privacy laws, such as HIPAA (Health Insurance Portability and Accountability Act) [27]. Advanced AI can be beneficial in identifying and reporting privacy breaches [28]. This allows healthcare facilities to take steps or actions before a breach occurs.

The potential for advanced intelligence to improve data security in the healthcare sector is clear. As the healthcare sector continues to evolve, advanced AI can become an indispensable tool for protecting patient data.

4.3 Edge-AI and IoT for Intelligent Agriculture

Edge-AI and IoT-based smart farming refers to the use of advanced AI and the Internet of Things (IoT) to improve efficiency and productivity in the agricultural industry.

Today, the use of AI techniques is playing an important role in the development of precision agriculture [29]. Predictive crop models can be designed. They take into account different production factors such as: climate; soil condition; seed condition; water supply; disease risk, etc. By taking all these factors into account when designing the model, it is easier to adjust any of the elements to optimize yield.

In irrigation management [30], for example, IoT can be used to monitor and control crop irrigation in real time. Sensors could be placed in fields to measure soil moisture, temperature and other key factors, helping to determine the optimum time for irrigation and avoid wasting water. Advanced computing combined with artificial intelligence will enable the data collected by the sensors to be processed and analyzed locally in real time. Personalized irrigation strategies can then be recommended for each plot based on soil and crop characteristics. This enables efficient and sustainable farming practices to be put in place.

Another benefit of advanced AI in agriculture is its ability to improve animal welfare [31]. Using wearable sensors, farmers will be able to monitor the health and well-being of their livestock in real time. Edge-IA devices will analyze the data from these sensors to detect early signs of disease. This will allow farmers to intervene before problems arise or worsen. This will improve animal welfare, reduce the use of antibiotics and increase productivity.

The ability to optimize crop yields and reduce waste is another key benefit of using advanced AI in agriculture [29]. Indeed, the analysis of data from soil sensors, weather stations and satellite images by Edge-IA devices will be able to provide farmers with real-time information about the health and needs of their crops. This information can then be used to make informed decisions about irrigation, fertilization and pest control, so that crops receive the precise amount of resources they need to thrive. This can lead to increased crop yields, reduced resource consumption and minimized environmental impact.

- **The Intelligent-Edge layer:** this layer lies between the IoT device layer and the cloud layer. It consists of edge-intelligent devices for the collection, storage, artificial intelligence-based processing and real-time analysis of data from intelligent devices and its transmission to the cloud using wireless transmission technologies such as 5G, Sigfox, LoRa-WAN, NB-IoT [21, 22] etc. This layer also plays a role in securing data transmission. Indeed, features such as tamper detection, encryption and others can be implemented at the edge-intelligent device level, preventing malicious attacks against IoT devices and securing user data to the cloud [23].
- **The Cloud layer:** this layer comprises servers dedicated to storage and data processing. It therefore hosts the data from the edge-intelligent layer, for in-depth processing. The Cloud layer gives IoT application programmers the means to work with heterogeneous objects without having to rely on a specific hardware platform. It is also responsible for service management (Fig. 2).

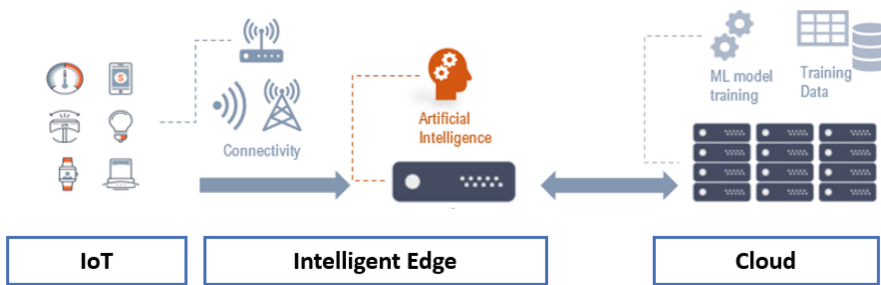


Fig. 2. Intelligent edge architecture.

4.2 Edge-AI and IoT for Connected Health

When healthcare professionals are faced with life-and-death decisions, the quality of the information available to them is critical. More accurate information and real-time access mean better decisions. Thanks to the power of the Internet of Things (IoT) and advanced AI, connected healthcare systems will be able to deliver real-time data to providers and enable more efficient care delivery.

Edge intelligence allows data to be processed and analyzed in real time without having to send it to a central server such as the cloud [4]. This can be very useful in healthcare facilities, where data needs to be analyzed quickly and securely.

One of the benefits of advanced AI in medical data is its ability to detect anomalies [24]. This can help healthcare staff to quickly identify potential health problems, take appropriate action and make more informed decisions about patient care.

The use of advanced AI can also improve the efficiency of healthcare systems. Real-time analysis of data can enable healthcare providers to reduce waiting times and improve patient outcomes [25]. Edge intelligence can also be used to automate the scheduling of appointments or the management of medical records.

4 Advanced Edge-AI for Convergent Services and Applications

4.1 Edge-AI Concepts and Opportunities

Concepts: intelligent Edge refers to the combination of Edge Computing and Artificial Intelligence techniques. In simple terms, the Intelligent Edge will enable AI-based processing algorithms to be run at the heart of the devices that are part of the Edge Computing network, using a reasonable internet connection. Everything happens in the device: collection, storage and AI-based processing. This results in extremely short response times of a few milliseconds. The information arrives in real time with unparalleled accuracy, since the algorithms will have refined the raw data. Even today, the majority of heavy computing is done in the cloud and requires large computing capacities. The Intelligent Edge will move some of the computational processing flow directly into the devices, to reduce the use of the cloud significantly. With this type of device, all data can be stored before being sent to a remote site for further analysis, if required. These intelligent devices, thanks to the availability of sensor data in the field, can interact autonomously without interaction with the central site.

Opportunities. Edge AI has huge potential. Here is a non-exhaustive list of the advantages that can be derived from this technology:

- **Ease of management:** Edge-AI enabled devices are easy to use because the objects are completely autonomous.
- **Reduced latency:** The AI will not be subject to congestion problems because it is as close as possible to the objects.
- **Deployment “in the field”:** Edge-AI lends itself by definition very well to IoT uses, but also in all mobile devices, starting with autonomous cars.
- **Reduced costs:** Bandwidth and data exchange costs are reduced because the amount of data transmitted is minimal.
- **Security and confidentiality:** Edge-AI makes it possible to filter the data to be uploaded to the cloud if necessary. Only the desired information is transmitted, after being anonymized for example. Data confidentiality and the protection of privacy are two major issues to which Edge-AI can provide a solution, since this approach completely reverses the paradigm compared with AI in the Cloud, which requires systematic transmission of all the data before it is analyzed by the AI.

The following Fig. 2 shows an example of a three-layer intelligent systems architecture assisted by IoT and Edge-IA. The layers of the architecture are identical to those of the Edge computing architecture with the exception of the edge-intelligent layer where we will find the artificial intelligence algorithms. Here, IoT devices represent the end users of the architecture.

- **The IoT layer:** this is the first layer and includes sensors, RFID devices and actuators for collecting information such as temperature, humidity, location, air quality and movement. Once the information has been collected, it is transmitted to the Edge layer via a secure wireless connection. This layer requires the use of standardized plug-and-play mechanisms for configuring heterogeneous objects. IoT mega-data is created at this layer.

and transmission power. These objects are connected by networking and communication protocols, allowing them to communicate and cooperate with each other to share information. These protocols represent the backbone of the IoT, they specify the data exchange format, data encoding, object addressing schemes as well as data packet routing [10]. Sequence control, flow control and retransmission of lost packets are also part of their functionality. In the Edge-IoT system, several protocols of different types are used. However, the best technology for our IoT use cases means that we need to accurately weigh the criteria in terms of range, bandwidth, QoS, security, power consumption, and network management.

The Tables 1 and 2 provide an overview of the most commonly used networking and messaging protocols.

Table 1. Overview of the most commonly networking protocols

Protocols	Frequency	Data rate	Range	Topology	Service
Zigbee [11]	2,4 GHz	250 kbps	10-100 m	Mesh, Star	Home automation
BLE [12]	2,4 GHz	1 Mbps	100 m	Star, Bus	Small appliances
WIFI [13]	5 GHz	1 Gb	100 m	Star, Mesh	Home
NFC [14]	13,56 MHz	420 kbps	20 cm	P2P	Payment system
LoRaWAN [15]	125 kHz	50 kbps	5-20 km	Star	Smart city
5G [16]	30-300 GHz	50 Gbps	Several kms	Star	Virtual reality applications

Table 2. Overview of the most commonly messaging protocols

Application protocols	Restful	Trans. Layer protocol	Architecture	Security	QoS
AMQP [17]	No	TCP	Pub-Sub	SSL	Yes
CoAP [18]	Yes	UDP	Pub-Sub Req-Res	DTLS	Yes
MQTT [19]	No	TCP	Pub-Sub	SSL	Yes
DDS [20]	Yes	UDP/ TCP	Pub-Sub Req-Res	DTLS, TLS, DDS Security	Yes

affect transmission time are reduced. Then considerably, the network performance is improved to meet the latency sensitive IoT applications.

Computation: in the Edge-IoT architecture, the computation follows the Edge cloud combination mode. The computation will be offloaded to the Edge servers thus relieving the cloud servers. Therefore, the speed and efficiency of the network in terms of resource utilization will be guaranteed.

Storage: the number of devices connected to the IoT is increasing and producing huge amounts of data. Previously, to be processed, all this data was at centralized cloud servers. This severely affected the user experience. The migration of all data collected by IoT devices to Edge storage servers solves this problem. Edge computing uses load balancing technology across different nodes, which would improve the quality of service.

The following Fig. 1 shows an example of a three-layer intelligent systems architecture assisted by IoT and Edge computing. However, note that the Edge-IoT architecture can change depending on the problem it wants to solve for an IoT application [7].

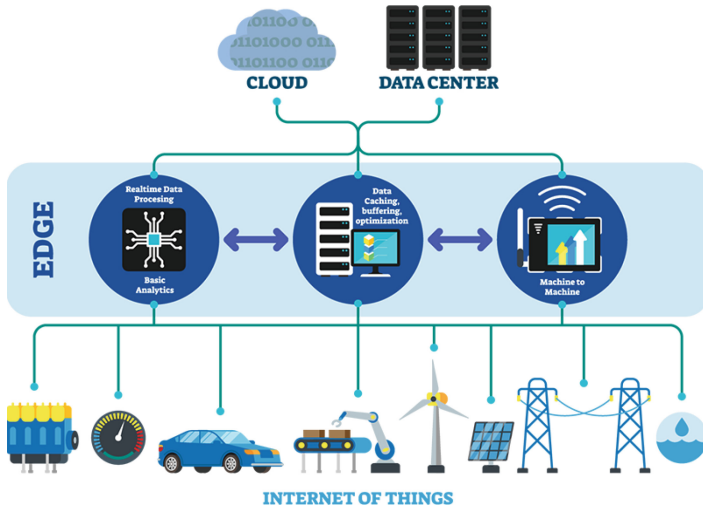


Fig. 1. IoT-Edge architecture from [9]

However, in-depth studies of this new computing model have shown that traditional non-IA techniques have limitations in improving the performance of Edge computing. With the successful application of AI in various fields, Edge computing researchers are beginning to turn their attention to AI, in particular machine learning, a branch of AI that has gained popularity over the last few decades.

3.2 Edge-IoT Protocols for Intelligent Systems

The Edge-IoT system is a heterogeneous network that includes different devices that may have different communication, data processing and storage platforms, networks

Processing the data from IoT monitoring is much more difficult. Redesigning the monitoring system's architecture will enable real-time data processing and responsiveness.

To enable real-time performance, the IoT monitoring system must continuously evaluate data as it comes in and improve its insight and decision-making capabilities to support sophisticated business logic [6].

Many research works have shown that edge computing can improve the performance of IoT networks and provide real-time services. They highlight the benefits of edge computing for various intelligent applications.

Authors in [5] discuss edge computing for Internet of Things (IoT) applications. It examines how edge computing can improve the performance of IoT networks, particularly latency, by improving computing and storage capabilities, and providing real-time services to users. The paper presents a comprehensive review of edge computing architectures for IoT, classifying them by architecture and comparing their performance in terms of network latency, compute capacity, and storage space. The authors also examine the security issues associated with these approaches, as well as the advantages and disadvantages of edge computing for IoT. In conclusion, the paper highlights the potential benefits of edge computing for various smart applications, including smart grids, smart cities, and smart transportation.

Paper [7] discusses the challenges of storing and processing the enormous amount of data generated by IoT devices, as their resources are limited. It explains that while cloud computing can be used to handle some of these resource issues, it can also introduce latency in time-critical IoT applications. This paper also highlights the limitations of Edge-computing architectures (ECAs)-IoT and proposes solutions to overcome these limitations. It outlines various IoT implementations in the edge computing domain and suggests four distinct scenarios for using IoT implementations with ECAs-IoT. Additionally, the document discusses the concepts of edge computing, implementation strategies, and edge intelligence. It provides a taxonomy for IoT applications based on different criteria and compares fog computing to cloud computing.

The universal complex event processing (CEP) mechanism for IoT real-time monitoring is proposed in [8]. It proposes a formalized hierarchical complex event model including raw event, simple event and complex event, which reduces the complexity of event modeling. The approach allows for flexible, complicated events to be defined by programming using sophisticated time and space semantics. The proposed system is installed at the network edge between cloud-based apps. Two application case studies of IoT monitoring for fruit transportation and city road manhole cover status are implemented based on the CEP system to highlight our proposed complex event model.

3 Edge-IoT Architecture and Protocols for Intelligent Systems

3.1 Edge-IoT Architecture for Intelligent Systems

Edge computing addresses the requirements of IoT applications on three categories, such as communication, computation and storage [5].

Communication: with the assistance of Edge computing, the performance in terms of bandwidth utilization, latency, device power consumption and packet data overhead that

- Description of the concept and architecture of Edge-IA technology by focusing on its applications in connected health and smart agriculture.
- Discussion of the security challenges and future directions in Edge-AI on IoT Environment.

The remainder of this paper is organized as follows: In Sect. 2, we introduce the motivation for using Edge computing and summarise related work on edge computing for real-time Internet of things applications. In Sect. 3, we present an example architecture for Edge computing-assisted intelligent IoT systems and integration protocols. Then, in Sect. 4, we discuss the concept and opportunities of Edge-IA technology and also present a architecture and application areas in connected health and smart agriculture. In Sect. 5, we discuss security challenges and future directions. Finally, in Sect. 6 we conclude by showing the benefits that Edge-IA brings to intelligent systems.

2 Background

2.1 Motivation for Using Edge Computing

The growth of internet of Things has led to increased demand for data processing and analysis, which introduced the development of new computing models, such as edge computing, which focuses on bringing processing and data closer together, storing end-user data to reduce latency, improve performance, and relieve computing demand on centralized data centers. Data processing and analysis at the network edge helps improve the performance of IoT networks and provide real-time services to users.

Traditional cloud computing architectures, while powerful and versatile, are not always equipped to meet the demands of real-time IoT applications. The latency associated with data transit to centralized cloud servers (as well as bandwidth limitations) can hinder the very essence of IoT: immediate, contextual responsiveness. To overcome these limitations, a paradigm shift is underway, which brings computation closer to the data source and therefore to the edge of the network [5]. Edge Computing is the transformative solution to the real-time challenges posed by IoT. Indeed, Edge-based IoT deployments must be managed, efficiently, effectively, and automatically. IoT applications that involve endpoints, edge computing and fog are complex to manage.

2.2 Edge Computing for Real-Time Internet of Things Applications

The IoT real-time monitoring market is expanding quickly in many IoT innovative applications, including smart logistics, smart farms, environmental monitoring, intelligent transportation, and smart power grids; that to the rapid development of sensors, GPS position sensors, RFID tags and readers, smart objects, and other IoT sensing technologies.

By the year 2025, 50 billion devices will be online, according to new research. These Internet of Things (IoT) sensing devices are data generators that continuously provide fresh perception data, offering enormous potential for real-time monitoring and intelligent application across numerous industries.

improves efficiency, and enables data-driven decision-making across various sectors, including home automation, healthcare, transportation, agriculture, and industry [1]. IoT represents a significant shift in how we interact with our surroundings, offering new possibilities for convenience, productivity, and innovation while also raising challenges related to data security, privacy, and scalability.

Edge computing is a distributed computing paradigm that brings computational processing and data storage closer to the source of data generation, typically at the "edge" of a network or IoT devices, rather than relying on centralized cloud data centers. This approach reduces latency, enhances real-time processing, and conserves bandwidth by processing data locally or in nearby edge servers [2]. Edge computing is essential for applications requiring low latency, such as autonomous vehicles, industrial automation, and IoT devices, enabling faster responses and improved efficiency. It complements cloud computing by decentralizing computing resources, allowing for more efficient and responsive data processing in various domains. We can result that this reduces latency, enhances data privacy, and conserves bandwidth by processing and analyzing data locally or in nearby edge nodes.

Artificial Intelligence (AI) is a multidisciplinary field of computer science focused on creating machines and computer systems that can perform tasks that typically require human intelligence. For this, AI encompasses machine learning and deep learning techniques that enable computers to perform tasks that typically require human intelligence, such as learning from data, making decisions, recognizing patterns, and natural language understanding [3].

Edge AI (or intelligent Edge) appeared in a desire to integrate artificial intelligence as close as possible to sensors or connected objects (IoT). The advantages are multiple: absence of prerequisites for permanent internet connectivity, data confidentiality, reduction of latency. To achieve the expected results, Edge AI makes it possible to move part of the computing flow directly into the connected objects, and therefore to minimize the use of the cloud for the tasks related to the processing.

In essence, Edge AI empowers devices and sensors to become "smart" by running AI algorithms locally, enabling quicker decision-making and reducing the need for constant internet connectivity. It finds applications in various fields, including autonomous vehicles, smart cities, healthcare, and industrial automation, where low latency and real-time processing are critical [4]. Edge AI is at the forefront of innovation, addressing the demands of our increasingly connected and data-driven world while raising challenges related to resource constraints, security, and scalability.

In this paper, we provide a holistic view of the convergence of Edge AI and IoT, offering insights into its architectural foundations, diverse applications, ongoing challenges, and promising future directions. The keys contributions of this paper are listed as below:

- Overview of IoT, Edge Computing, and Artificial Intelligence (AI) paradigms. Particular.
- Summary of related work on the motivations for using Edge computing and Edge-IoT systems for real-time monitoring.
- Overview on networking and communication protocols for Edge-IoT systems.



Edge- AI and Internet of Things for Intelligent Systems: Architectures, Applications and Future Perspectives

Fatou Diop^(✉), Babacar Mbaye Faye, and Ibrahima Niang

Cheikh Anta Diop University, Dakar, Senegal

{fatou110.diop,babacarmbaye.faye,ibrahima1.niang}@ucad.edu.sn

Abstract. In an era where connectivity is ubiquitous and data flows ceaselessly, Edge IoT (Internet of Things) systems have emerged as a pivotal technological paradigm poised to revolutionize the way we interact with the digital world. As the proliferation of IoT devices continues unabated, the traditional cloud-centric model for data processing and decision-making faces significant challenges. Edge IoT systems, by shifting computational intelligence closer to the data source, offer a compelling solution to these challenges, promising greater efficiency, lower latency, and enhanced real-time decision-making capabilities. On the other hand, the convergence of edge computing, artificial intelligence (AI), and the Internet of Things (IoT) has ushered in a new era of intelligent systems with transformative potential across various domains. This includes the development of edge computing infrastructure, AI models optimized for edge deployment, and IoT device architectures that support real-time data processing and analysis. This paper proposes a comprehensive review on Edge computing, AI, and IoT integration for intelligent systems. We focus on the concept of Edge-AI and its potential applications in real-time processing for Internet of Things. The paper introduces Edge-IA technology, explores commonly used networking and messaging protocols in IoT, and discusses the opportunities and challenges of Edge-IA and IoT in various industries, including agriculture and healthcare.

Keywords: Internet of things · Edge computing · Artificial intelligence · Edge-AI

1 Introduction

In recent years, the Internet of Things (IoT) has become one of the most important technologies of the 21st century. IoT is a revolutionary concept that involves connecting everyday objects and devices to the internet, allowing them to collect and exchange data, communicate with each other, and be remotely controlled. IoT transforms ordinary items, such as appliances, vehicles, wearables, and industrial machines, into “smart” devices capable of sensing, transmitting, and acting upon data. This interconnected network of devices and sensors provides real-time information, enhances automation,

9. Liu, N., Dong, X.: Research on the reform and security of college examination. Journal of Shenyang Architectural University (2008). [Citation Time(s):1]
10. Wang, L., Liu, H.-Y.: Analysis on the Restrict of College Examination Reform. Higher Engineering Education Research (2006)
11. Han, S.-J.: The exploration of evaluation mode in the high professional university. Educ. Prof. **29**, 166–167 (2011)
12. Li, X.-Q., Shu, H.: The Exploration and Practice of Examination Model Reformed by Stratified Teaching in Higher Vocational (2011)
13. Mathematics Curriculum. Journal of Chifeng University, vol. 9, pp. 252–253
14. Niu, W.-X.: Quality education calls for the examination reform. Educ. Naval Acad. **3**, 73–74 (2000)
15. Zhao, J.-Z.: The research on the present situation of the examination reform in higher vocational colleges. Chin. Exam. **7**, 40–44 (2010). [Citation Time(s):1]
16. Al-hawari, F., Althawbih, H., Alshawabkeh, M., Abu Nawas, O.: Integrated and secure web based examination management system. Computer Applications in Engineering Education (2019)
17. Khan, T.: Examination Scheduling System with Proposed Algorithm, s. d., 1
18. Omole, C.: Modifying Assessment Modes in the Science Classroom as a Solution to Examination Malpractice, s. d., 1
19. Jamaluddin, N.F., Aizam, N.A.H.: Timetabling Communities' Demands for an Effective Examination Timetabling Using Integer Linear Programming, s. d., 1
20. Ajith, R., Bijlani, K.: Serious Game as a Performance Assessment Tool That Reduces Examination Anxiety, s. d., 1

4 Results

We have a bank of questions that we can enrich over time with our two methods. These results are presented in a CSV file, and we plan to set up a platform to automate the populating of this dataset.

5 Conclusion

In this paper, we have recalled the context and the goal of automating evaluations. Subsequently, we used NLP to give a proposal for automatic question generation by proposing two features(negation, and duplication).

On related works such as [1, 2] and [3], we have multiple choice generations question that are based on wikipedia topics and courses respectively. In all cases, it is a generation of questions on knowledge bases. To improve these rich scientific works, we propose a reformulation of the questions already existing in a question bank by presenting two functions. One modifies the question by taking its negation, the other duplicates the question by modifying a main keyword.

In perspective, we are looking:

- The quality of the questions produced by our two functions.
- The generation of distractors from the questions
- The implementation of the application in a general way.

discuss other ways of generating questions such as information retrieval.

References

1. Graesser, A.C., Wisher, R.A.: Question generation as a learning multiplier in distributed learning environments. US Army Research Institute for the Behavioral et Social Sciences, Alexandria VA (cf. p. 1, 21, 23, 85, 122) (2001)
2. Raynaud, T.: Génération de questions à choix multiples thématiques à partir de bases de connaissances. <https://tel.archives-ouvertes.fr/tel-02901501/document>
3. Nwafor, C.A., Onyenwe, I.E.: An automated multiplechoice question generation using natural language processing techniques. <https://aircconline.com/ijnlc/V10N2/10221ijnlc01.pdf>
4. uvs(DISI) Université Virtuelle du Sénégal[en ligne] (modifié en mars 2023). <https://www.uvs.sn/luniversite-2/luniversite-en-chiffres/>. Accessed 21 June 2022
5. DATASCIENTEST [en ligne] (Created on 22/07/2020). <https://datascientest.com/introduction-au-nlp-natural-language-processing>. Accessed 11 Feb 2022
6. spacy[en ligne] (created since 2016). <https://spacy.io/usage/spacy101#whats-spacy>. Accessed 21 Oct 2022
7. Kuchling, A.M.: (Documentation Python 3.11.2) (modified on March 30, 2023). <https://docs.python.org/fr/3/howto/regex.html>. Accessed 30 Oct 2022
8. waset. Conférence internationale sur le système de gestion des examens Riga, Lettonie du 18 au 19 juin 2020. <https://jemsenegal.wordpress.com/2021/04/06/jems-un-modele-de-systeme-demanagement-des-examens-pour-les-universites-du-senegal/>

```
[ ] 1 import re
    2 def affirmative_en_negatives(x):
    3     verbe=trouve_le_verbe(x)
    4     print(verbe)
    5     phraseNeg=re.sub(verbe, ' ne '+verbe+' pas ',x)
    6     return phraseNeg
    7
```

```
[ ] 1 phrase=input('Donner une phrase: ')
```

Donner une phrase: La balise main spécifie le contenu principal d'un document.

```
[ ] 1 trouve_le_verbe(phrase)
```

'spécifie'

```
[ ] 1 affirmative_en_negatives(phrase)
```

spécifie

'La balise main ne spécifie pas le contenu principal d'un document.'

Fig. 3. Affirmative to negative

```
1 import re
2 qu=input('Une question')
3 listMots=[]
4 while nbq in range(3)):
5     nbq=int(input('Combien de question voulez vous reproduire?'))
6
7 for j in range(nbq):
8     m=input('saisir un mot')
9     listMots.append(m)
10 print(listMots)
11 mot_a_replacer=input('Donnez le mot cles a remplacer')
12 listqu=[qu]
13 i=0
14 for mot in listMots:
15     quNouveau=re.sub(mot_a_replacer,mot,qu)
16     listqu.append(quNouveau)
17     i=i+1
18 listqu
```

Fig. 4. Duplicate question

Une questionQuel attribut HTML est utilisé pour ajouter un style dans un élément?
 saisir un motlien
 saisir un motjeu de caractere
 saisir un motcouleur de font
 ['lien', 'jeu de caractere', 'couleur de font']
 Donnez le mot cles a remplacerstyle
 ['Quel attribut HTML est utilisé pour ajouter un style dans un élément?',
 'Quel attribut HTML est utilisé pour ajouter un lien dans un élément?',
 'Quel attribut HTML est utilisé pour ajouter un jeu de caractere dans un élément?',
 'Quel attribut HTML est utilisé pour ajouter un couleur de font dans un élément?']

Fig. 5. results of the duplication



Fig. 2. Verb detection

“ne ... pas” (in french) with the sub function. The sub() function searches for all substrings where the RE matches and substitutes them with a different string (Fig. 3).

3.3 Question Generation by Duplicating the Question

The execution of this code gives us the following results:

For this part we used regular expressions with the sub function which is explained above.

1. The user has to give a keyword to replace.
2. Then we build the dictionary of words that we will use
3. From this dictionary, we build a list to have a set of questions.

For more details on regular expressions in Python, please consult the documentation [7]. Try to add more comments about the methods and the performance of the models (Fig. 4).

This algorithm produces the following results (Fig. 5).

3.2 NLP Approaches Based on Word-Embedding

As in [4], NLP, Natural Language Processing, is a discipline that focuses on the understanding, manipulation and generation of natural language by machines. Thus, NLP is really at the interface between computer science and linguistics. It is about the ability of the machine to interact directly with humans.

3.2.1 Description of Our Two Algorithms Stated Above

Preprocessing In NLP, the first step is often preprocessing, which includes the tokenization and text cleaning steps. We are content here with a minimalist preprocessing: removal of punctuation and stop words (for visualization and vectorization methods based on counts). We propose to use the spaCy library which allows better automation in the form of preprocessing pipelines. spaCy is a free open source library for advanced natural language processing (NLP) in Python. If we work with a lot of text, we might want to know more about this. For example, what is it? What do the words mean in their context? Who does what to whom? What companies and products are mentioned? Which texts are similar? spaCy is designed specifically for production use and helps we create applications that process and “understand” large volumes of text. It can be used to create systems for information extraction or natural language understanding, or to preprocess text for deep learning. For more information on spaCy see [5].

3.2.2 Question Generation Using Negation

Can we reproduce other questions from this topic? It is possible to use the negation or remove the negation.

Example: the question (The imain; tag specifies the main content of a document. True or False and the correct answer is True) One could be modified as follows: The main tag does not specify the main content of a document. True or False and the correct answer is False Likewise for the question (Which of the tags does not belong to HTML 5? Here all the answers are good) In this case the modification is as follows: Which of the tags belong to HTML 5 ? and the answer is no right answer.

To make a question negative, we propose two steps: **Verb detection:** To write this algorithm, we first installed the spaCy library in order to be able to use the various useful functions to find the verb in the sentence and import the ‘fr_core_news_sm’ language (Fig. 2).

Once the verb is found, we will define an “affirmative_en_negative” function or we use regular expressions (re) with the sub function.

Regular expressions (denoted RE or regex patterns in this document) are essentially a small, highly specialized programming language embedded in Python and whose manipulation is made possible by the use of the re module. Using this small language, you define rules to specify a match against a desired set of strings; these strings can be sentences, email addresses, TeX commands or whatever. You can also use RE to modify a string or split it in different ways. Like our case in the “affirmative_en_negative” function, we enclose the verb in

- testinvite It is an e-examination platform that offers the following solutions:
 - the creation of exams,
 - the design of examination processes,
 - the presentation of online exams,
 - report generation
 - analysis of examination results,
 - the pricing structure.

In their [documentation](#), it specifies that there is no payment but certain sections such as sending exam invitations to candidates by the Test Invite electronic messaging system, starting a candidate’s exam, Video or photo surveillance for exam security are charged. Head to [testinvite](#) for more information.

We have many systems that offer interesting solutions for exams, But we propose automatic methods for populating a question bank using NLP.

A	B	C	
No.	QUESTIONS	REPONSE	DISTRACTEURS(OPTIONNEL)
1	Quel attribut HTML est utilisé pour ajouter un style dans un élément?	Style	
2	Quel attribut HTML est utilisé pour ajouter un lien dans un élément?	href	
3	Quel attribut HTML est utilisé pour ajouter une couleur de fond dans un élément?	background	
4	C'est quoi un élément sémantique ?	Un élément qui a une signification.	
5	Lesquelles des balises n'appartiennent pas à HTML 5?	<code><center></code> <code></code> <code><strike></code> <code><sp></code> <code><u></code>	

Fig. 1. bank of question

3 Matériel et Méthodes

In this study, we have taken the case of the UN-CHK, more precisely the multimedia stream with the Web Technology1-HTML/CSS module. We use two methods to populate the question bank:

- Use the negation of a question.
- Duplicate a question by modifying a keyword.

3.1 Presentation of the Dataset Web Technology-Initiation in HTML/CSS

This dataset has three parts: (Fig. 1)

- The question column proposed by teachers or tutors
- The answer column of each question
- The distractors column which is optional. The user can give it but we can propose an algorithm that calculates it from the proposed question.

We propose two techniques for question generation:

1. Use the negation of a question.
2. Duplicate a question by changing a keyword

Before this design idea using NLPs, we had set up a prototype whose goal was to bank exam subjects. This project is based on a database model and is UN-CHK, Senegal developed with the php language using the MVC model. This MVC model does not allow us to enrich the bank of questions in an automatic way, so it will be necessary to use other models of information retrieval and/or machine learning, more precisely the NLP models. The paper is organized as follow: We present our subject bank dataset in the next section. Then we explain the NLP approach we used to automatic question generation in section three. We describe the two algorithms we used to generate questions automatically in the fourth section and we finish by a general conclusion in the last section.

2 Bibliographic Review

The comprehensive design of computerized testing systems has been explored by many researchers in different institutes around the world such as those in [9–20]. For example in [9–16], the effective reform of college examination management is discussed Based on the concept of scientific development of examination management. In [16], the analysis, design, development, integration, deployment and security of a web-based exam management system which was developed in-house at the German Jordanian University (UGJ) was addressed. An algorithm for the exam scheduling system is recalled in [17]. An algorithm for the exam scheduling system is recalled in [17]. In relation to malpractices, [18] then identified modifications that could help relieve the student from exam-related stress and thus increase the student’s effort towards effective learning and discourage malpractices in the long run. term. The problem of university examination timetable was explained in [19] using topics from Universiti Malaysia Terengganu (UMT). Still for the simplification of exams, [20] proposes the “Serious games” as tools to carry out evaluations by reducing the exam anxiety encountered by learners.

However, we have plenty of EMS that exist today and which offer important solutions for e-exams with competitive prices. We will detail them on the following lines:

- testwe
testwe is a complete e-Exam solution: online platform for teaching staff and administration, offline software for learners In their [white paper](#) entitled “THE 5 IMPACTS OF THE E-EXAM” they describe how e-exams can help institutions compared to paper-based exams.
- Managexam By bringing together teaching and student administration around the same digital environment, Managexam facilitates the organization and management of evaluations while guaranteeing a high standard of educational integrity at each stage. Thanks to its simple and flexible tool Managexam allows the design of:
 - Classic subjects
 - Evaluation activity enriched with video, audio recordings or large documents with customization options such as random drawing of questions or even variability of data between candidates

For more details, [go to](#)



Question Design Using NLP

Maty Sene Kane^{1(✉)}, Alassane Diop¹, Zinflou Arnaud²,
and El Hadji Mamadou Nguer³

¹ Université Numérique Cheikh Hamidou Kane (formerly UVS), Thiès, Senegal
`maty1.sene@unchk.edu.sn, alassane.diop@unchk.edu.sn`

² Hydroquebec-CANADA, Montreal, Canada
`zinflou.arnaud@hydroquebec.com`

³ Université Numérique Cheikh Hamidou Kane (formerly UVS),
Dakar, Diamniadio, Senegal
`elhadjimamadou.nguer@unchk.edu.sn`

Abstract. Cheikh Hamidou KANE Digital University (ex UVS) has more than 60,000 students enrolled in 46 tracks containing more than 1,000 courses, the assessment of which requires the preparation of nearly 2,000 exam topics per year. Since its opening in 2013, it is estimated that nearly 11,000 topics have been prepared, and this number is growing (see [4]). In addition to these numbers, on the one hand we have the lack of an examination management system (EMS). On the other hand, we see enormous difficulties in receiving assessment topics at the teacher level, given the large number of topics a teacher has to design. We propose a framework called JEMS (Jolof Examination Management System) that allows for declarative expression and evaluation of topics close to their design. In order to be able to generate questions automatically, we have made use of other models such as natural language processing based models. This framework aims to automate evaluation questions from an existing question bank for a CE (Constituent Element). Automating evaluations in this context poses a number of scientific challenges that constitute the contributions of the presented work:

- The implementation of a file template in csv format.
- Question models that identify entities in CEs and generate statements against the natural language processing model

Keywords: NLP · generate questions automatically · examination management system

1 Introduction

Computer-based exams have grown in popularity in recent years as they support e-learning and make it easier to manage (i.e. design, set up, schedule and score, publish results) of exams. However, adopting an appropriate system that meets the needs of a certain institution can be difficult due to several concerns related to integration with existing systems, security, customization, ease of use and cost.

References

1. Kasemthaweesab, P., Kurutach, W.: Association analysis of Diabetes Mellitus (DM) with complication states based on association rules, In: Proc. 7th IEEE Conference on Industrial Electronics and Applications, pp. 1453–1457. (2012)
2. <http://www.who.int/mediacentre/factsheets/fs312/en/>
3. Zarkogianni, K., et al.: A review of emerging technologies for the management of diabetes mellitus. *IEEE Trans. Biomed. Eng.* **62**(12), 2735–2749 (2015)
4. Collins, G.S., Mallett, S., Omar, O., Yu, L.M.: Developing risk prediction models for type 2 diabetes: A systematic review of methodology and reporting. *BMC Med.* **9**(1), 103 (2011)
5. Shankar Acharya, D.O., Samanta, S., Vidyarthi, A.S.: Computational intelligence in early diabetes diagnosis: A review. *The Review of Diabetic Studies: RDS* **7**(4), 252 (2010)
6. Fikirte Girma Wolde, M., Sumitra, M.: Prediction of Diabetes using Data Mining Techniques, Dept of Computer Science and Engineering, In: Proceedings of the 2nd International conference on Trends in Electronics and Informatics (ICOEI) (2018)
7. Terry Jacob, M., Elizabeth, S.: Analysis Supervised Learning Techniques for Cost Effective Disease Prediction using on-Clinical Parameters, IIITM-KTechno park, Trivndrum, (2018)
8. Butwall, M., Kumar, S.: A Data Mining Approach for the Diagnosis of Diabetes Mellitus using Random Forest Classifier. *International Journal of Computer Applications* **120**, 0975–8887 (2015)
9. Dewangan, A.K., Agrawal, P.: Classification of Diabetes Mellitus Using Machine Learning Techniques. *International Journal of Engineering and Applied Sciences*, **2** (2015)
10. Devi, M.R., Shyla, J.M.: Analysis of Various Data Mining Techniques to Predict Diabetes Mellitus. *International Journal of Applied Engineering Research* **11**, 727–730 (2016)
11. Hina, S., Shaikh, A., Abul Sattar, S.: Analyzing Diabetes Datasets using Data Mining. *Journal of Basic & Applied Sciences* **13**, 466–471 (2017)
12. Mujumdar, A., Vaidehi, V.: Prédiction du diabète à l'aide d'algorithmes d'apprentissage automatique Conférence internationale sur les tendances récentes en informatique avancée, 2019, ICRTAC (2019)
13. Yuvaraj, N., Sri Preetha, K.R.: Prédiction du diabète dans les systèmes de santé à l'aide d'algorithmes d'apprentissage automatique sur le cluster. *Hadoop Calcul de cluster* **22**, 1–9 (2017)
14. Sisodia, D.: DS Sisodia Prédiction du diabète à l'aide d'algorithmes de classification Process. *Comput. Sci.* **132**, 1578–1585 (2018)
15. Erveen, S., Shahbaz, M., Guergachi, A., Keshavjee, K.: Performance Analysis of Data Mining Classification Techniques to Predict Diabetes. *Procedia Computer Science* **82**, 115–121 (2016). <https://doi.org/10.1016/j.procs.2016.04.016>
16. Nai-Arun, N., Sittidech, P.: Ensemble Learning Model for Diabetes Classification. *Advanced Materials Research* 931–932, 1427–1431 (2014). <https://doi.org/10.4028/www.scientific.net/AMR.931-932.1427>

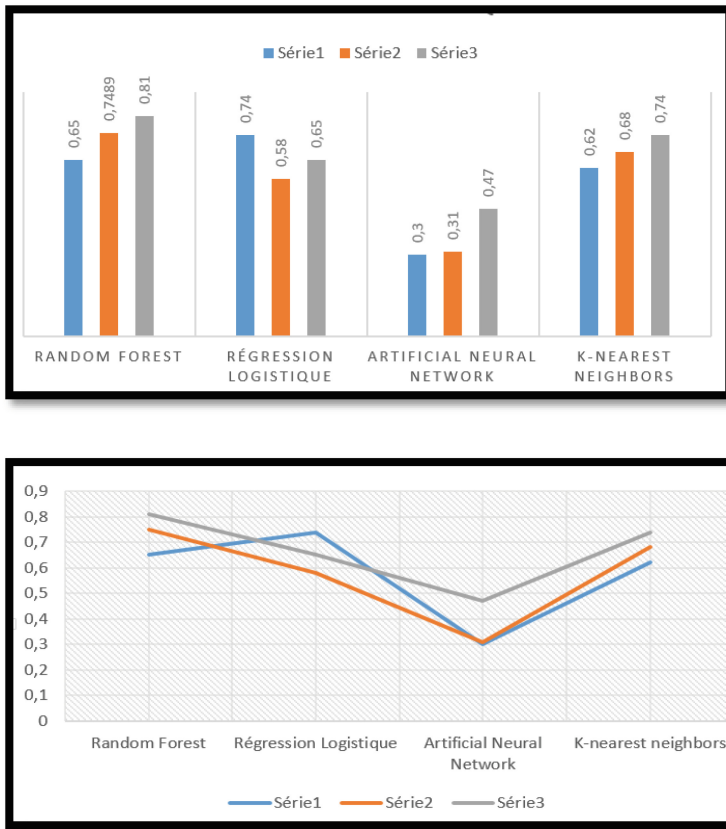


Fig. 10. Graphical representation of the results

5 Conclusion

Diabetes, acknowledged as one of the most perilous and persistent ailments leading to elevated blood sugar levels, has increasingly permeated every aspect of daily life, impacting families on a profound level. Recognizing the pervasive influence of this chronic disease, there is a heightened awareness of the critical medical challenge posed by the early detection of diabetes. In our study, we sought to compare three machine learning algorithms: Random Forest, Logistic Regression, and Neural Networks. The experimental findings, derived from the Kaggle dataset, reveal Random Forest's superiority in terms of heightened accuracy compared to the other algorithms. Subsequent endeavors will involve replicating the same experiments on additional diabetes databases or diverse datasets to validate and enhance the obtained results, with the ultimate aim of refining the algorithms for improved accuracy.

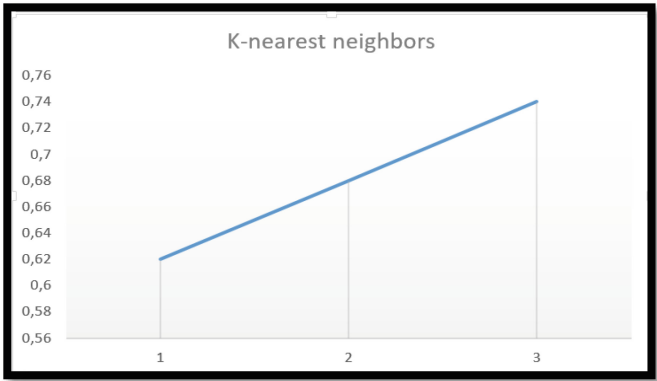


Fig. 9. Evolution of the tests with K-nearest neighbors

Table 5. Results obtained.

Model	Test 1	Test 2	Test 3
Random Forest	0,65	0,7489	0,81
Logistic Regression	0,74	0,58	0,65
Artificial Neural Network	0,30	0,31	0,47
K nearest neighbors	0,62	0,68	0,74

The graphs depict the performance of four models: Random Forest, Logistic Regression, K Nearest Neighbors, and Artificial Neural Network based on accuracy rates across varying iterations. Notably, the Random Forest model outperforms the Logistic Regression model, with the exception of iteration 1. Its peak accuracy is observed in iteration 3, reaching 81%. Conversely, the Artificial Neural Network (ANN) exhibits consistently modest results across different iterations, while the K Nearest Neighbors (KNN) algorithm demonstrates improvement with each iteration (Fig. 10).

In direct comparison, the Random Forest, Logistic Regression, and K Nearest Neighbors models outshine the Artificial Neural Network model. This underscores the notion that ANN, being a deep learning algorithm, may find greater utility in other data types, such as image processing. Early identification of individuals at a high risk of diabetes is a pivotal challenge in the healthcare domain. Within this study, the Random Forest model is pitted against the Logistic Regression model, the Artificial Neural Network model, and the K Nearest Neighbors model. The findings suggest that machine learning approaches effectively leverage extensive data sourced from electronic medical records for predicting diabetes risk.

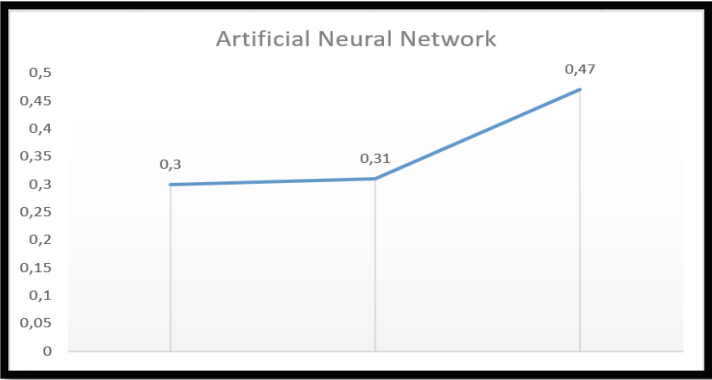


Fig. 8. Evolution of the tests with Artificial Neural Network

In the figure, we can clearly see that the best result is captured in the last test, this is justified by the fact that the more the tests increase, the more it gives satisfactory results. The graph represents the accuracy rate for the Artificial Neural Network model according to the number of iterations. It can be seen that the best iteration is iteration 3 with an accuracy rate equal to 47%.

4.4 K-nearest Neighbors

In the same proposed work, we also applied the K nearest neighbor’s algorithm on the same machine learning and test sample data while varying the K in steps of two. We obtained results for different tests. The results obtained from the experiments performed are reported in the Table 4 below:

Table 4. Results of the different tests with k-nearest neighbors

Model	Test1 K = 1	Test 2 K = 3	Test 3 K = 5
K-nearest neighbors KNN	0,62	0,68	0,74

From the results shown in the table we can see that the performance results vary between 0.62 and 0.74 for all precision measurements. We also notice that as the k parameter increases, the rate of precision measurements also increases. This is more explicit in the Fig. 9 below.

In the figure, we can clearly see that the best result is captured in the last test. This again shows that the performance is proportional to the variation of K. The graph represents the accuracy rate for the K-nearest neighbor’s model as a function of the values of K. It is illustrated that the best iteration is the iteration with k = 5 with an accuracy rate equal to 74%. Thus, a summary Table 5 of the different algorithms and their results are designed with the graph of illustration regrouping the four models.



Fig. 7. Evolution of the tests with Logistic Regression

In the figure, we can clearly see that the best result is captured in the first test. This divergence in performance between the training and the test phase is known as overlearning. The graph represents the accuracy rate for the logistic regression model as a function of the number of iterations. It is shown that the best iteration is iteration 1 with an accuracy rate equal to 74%.

4.3 Artificial Neural Network

In the context of our proposed research, we implemented the artificial neural network algorithm using the same dataset employed for machine learning and testing. Diverse tests were conducted, and it’s worth mentioning that, for this learning approach, a sigmoid activation function was utilized for the output layer, while both the input layer and the hidden layer employed a relu activation function. The outcomes of the experiments are detailed in the Table 3 provided below:

Table 3. Results of the different tests with artificial neural network

Model	Test 1	Test 2	Test 3
Artificial Neural Network (ANN)	0,30	0,31	0,47

From the results shown in the table, we can see that the performance results vary between 0.30 and 0.47 for the precise measurements. We also notice that as the number of tests increases, there is sometimes a slight increase. We can deduce that with the low accuracy rates of ANN that this algorithm which is a deep learning algorithm is more interesting on other types of data. This is more explicit in the Fig. 8 below.

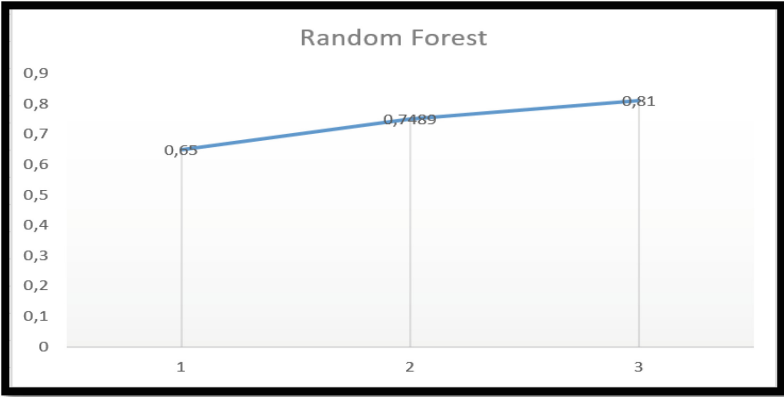


Fig. 6. Evolution of the tests with Random Forest

In the figure, we can clearly see that the best result is captured in the last test, this is justified by the fact that the more the tests increase, the more the model is trained, the more efficient it is and the better the results. The graph represents the accuracy rate for the Random Forest model as a function of the number of iterations. It is illustrated that the best iteration is the iteration 3 with an accuracy rate equal to 81%.

4.2 Logistic Regression

Subsequently, in the proposed research, we employed the logistic regression algorithm on the identical dataset used for machine learning and testing. Outcomes for various tests were acquired, and the results from the conducted experiments are presented in the Table 2 below:

Table 2. Results of the different tests with Logistic regression

Model	Test 1	Test 2	Test 3
Logistic Regression (RL)	0,74	0,58	0,65

From the results shown in the table we can see that the performance results vary between 0.58 and 0.74 for the precise measurements. We also notice that as the tests multiply, we sometimes see an increase and also a decrease as the tests multiply. This is more explicit in the Fig. 7 below.

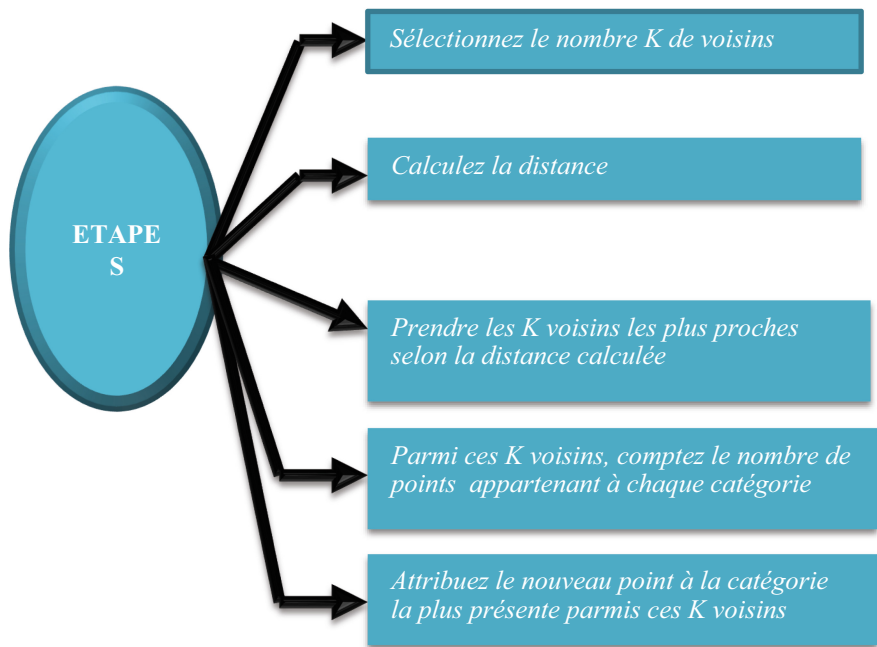


Fig. 5. K nearest neighbor’s steps

4.1 Random Forest (RF)

In our proposed study, we implemented the Random Forest algorithm in the realm of machine learning and tested it with sample data. Various tests were conducted, and the outcomes of these experiments are documented in the Table 1 provided below:

Table 1. Results of the different tests with random Forest

Model	Test 1	Test 2	Test 3
Random Forest (RF)	0,65	0,7489	0,81

From the results shown in the table, we can see that the performance results vary between 0.65 and 0.81 for the precise measurements. We also notice that the more the tests are multiplied, the more we note that the precision tends towards 100 which shows a good control at the training level. This is more explicit in the Fig. 6 below.

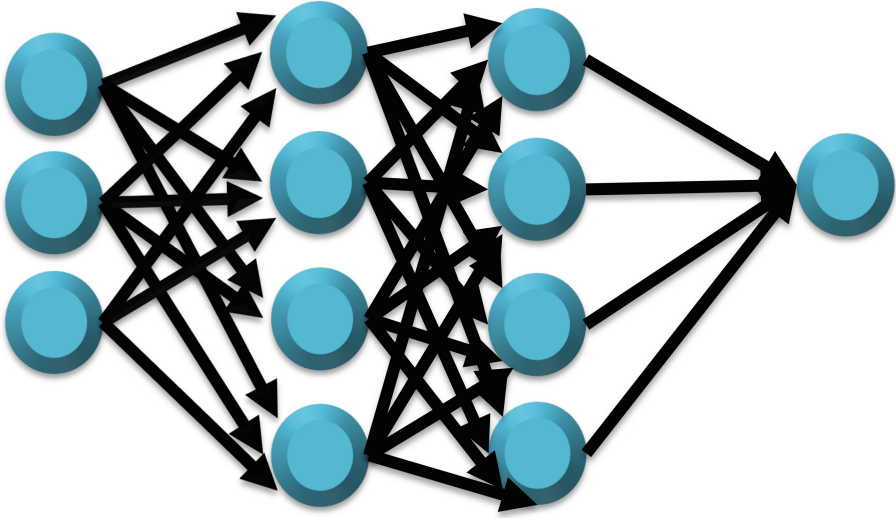


Fig. 4. Architecture of neural networks

*K Nearest Neighbors (KNN) stands out as one of the uncomplicated supervised learning methods employed to address classification and regression challenges. Its functionality revolves around the classification of new data points, determined by their similarity to neighboring data points. KNN is characterized as an algorithm devoid of assumptions about the data structure and distribution, rendering it a non-parametric algorithm. It is also referred to as a lazy learner algorithm, as it refrains from immediate learning from the training set. Instead, it stores the dataset and, during classification, takes action based on the dataset. KNN operates by classifying or predicting outcomes based on a fixed number (K) of data points in close proximity to input points. Essentially, for a selected value of K , an input point is classified or anticipated to belong to the same class as the nearest K neighboring points (Fig. 5).

4 Experimentation and Results

This section elucidates the methodology employed in this research article, outlining the acquisition of datasets and features. Additionally, it delves into the algorithms utilized and their corresponding evaluation criteria. Following the preprocessing steps that involved addressing null values and eliminating missing data, the predictive model was initially crafted using three algorithms: logistic regression, Random Forest, and neural networks. The Kaggle dataset under consideration encompassed individuals aged between 45 and 84, totaling over 700 participants. The impact of machine learning on diabetic prediction was tested comprehensively in this demographic. Google Colab served as the chosen working environment due to its ease of management and lack of requisite tool installations. The ensuing section presents the accuracy results derived from the examination of four distinct algorithms.

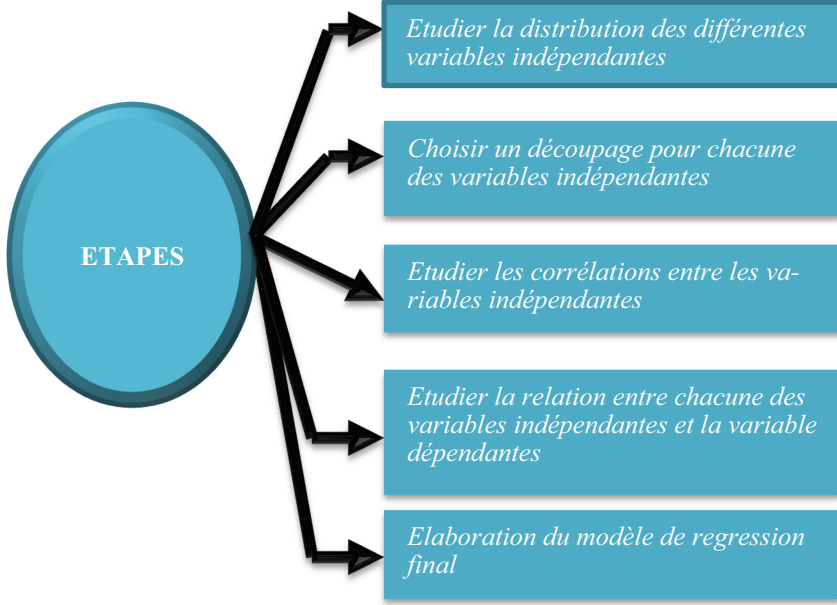


Fig. 3. Regression Logistic steps

Features are multiplied by random weights, $W_1, W_2, W_3, \dots, W_n$ and added with bias values, $b = b_1, b_2, \dots, b_n$. The resulting values are then input into a non-linear activation function, with various types of activation functions possible.

Activation functions can include (2), (3), (4), (5), which are some examples of activation functions (Fig. 3).

$$\text{Sigmoidfonction} : \sigma(z) \text{ or } a(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

$$\text{Tanhfonction} : a(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (3)$$

$$\text{RectifiedLinearUnit(ReLU)} : a(z) = \max(0, z) \quad (4)$$

$$\text{LeakyReLU} : a(z) = \max(0.001 * z, z) \quad (5)$$

Here is a synopsis of the proposed neural network model relevant to our study. The architecture comprises three layers, consisting of an input layer, two hidden layers, and an output layer. It's noteworthy that we have chosen to incorporate two hidden layers, but there is flexibility to create more or fewer layers as needed. The algorithm operates in the following manner: the outputs from the input layers serve as inputs to the first hidden layer, the outputs of which become inputs for the second hidden layer, and ultimately, the outputs of the second hidden layer become inputs for the output layer. The output layer produces the final predictive result (Fig. 4).

its predictive accuracy typically surpasses that of a single decision tree. As a rule, the greater the number of trees in the forest, the heightened robustness the forest exhibits (Fig. 2).

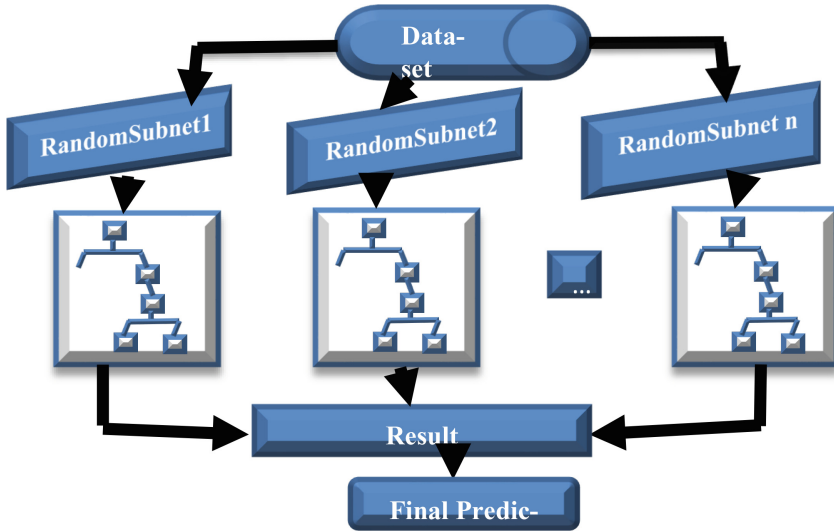


Fig. 2. Architecture of Random Forest

Presented below is the suggested framework for this algorithm aimed at predicting whether a patient is diabetic or not. The dataset used for consideration is comprised of diabetic data. Initially, 100 samples (Random Subnet) are extracted from the dataset, and an individual decision tree (Tree) is constructed for each sample. Each decision tree produces an output as a result. Ultimately, the final outcome is determined based on the collective voting of all the results generated by the various trees.

*Logistic Regression (LR): Logistic regression (LR) serves as a discriminant model contingent on the dataset's quality, characterized by the features:

$$\begin{aligned}
 X &= X_1, X_2, X_3, \dots, X_n (\text{where, } X_2 - X_n W_1, W_2, W_3, \dots, W_n, \text{ bias } b) \\
 &= b_1, b_2, \dots, b_n = \text{Distinct features poidset Cours } C \\
 &= C_1, C_2, \dots, C_n
 \end{aligned}$$

The equation for posterior estimation is expressed as follows:

*Artificial Neural Network (ANN) stands as a fundamental machine learning technique, forming the backbone of various deep learning algorithms. The training of the ANN model is accomplished using raw data, and in comparison to alternative classifiers, it possesses an extensive array of tuning parameters, contributing to its intricate structure. Optimizing the error in neural network instances requires a substantial amount of time compared to other techniques. To address this, instances of the neural network algorithm are trained on the graphics processing unit using CUDA programming. Each individual neural node within the ANN is trained with a set of features. $X = X_1, X_2, X_3, \dots, X_n (\text{where, } X_2 - X_n = \text{Distinct features})$

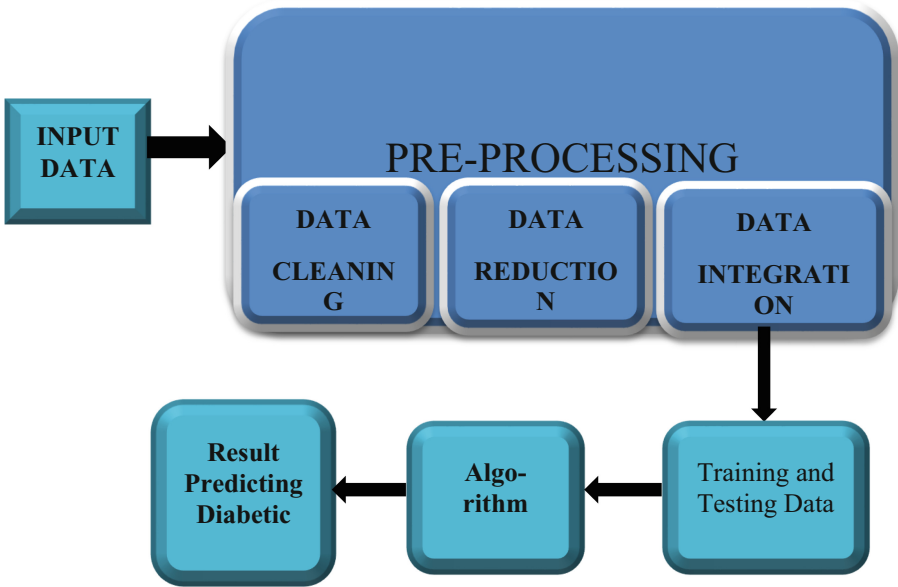


Fig. 1. System architecture

3.2 Data Pre-processing

Data pre-processing stands out as a pivotal stage in the data discovery methodology. Health information commonly exhibits missing or inconsistent data.

DATA CLEANING involves the strategic identification and correction, or removal, of inaccurate records. It encompasses the identification of incomplete, incorrect, inaccurate, or irrelevant information, with subsequent actions involving substitution, modification, or deletion of coarse data. Information cleaning is executed interactively using data merchandising tools or through scripted processes.

DATA INTEGRATION is the process of retrieving and consolidating heterogeneous data into a unified format and structure. This integration facilitates the utilization of diverse types of data, such as information sets, documents, and tables, for personal or business processes and functions.

DATA REDUCTION entails transforming numerical or alphabetical data, derived from empirical observation or experimentation, into a refined, organized, and simplified form. The core objective is to condense vast amounts of data into meaningful components.

3.3 Algorithms Used

*As implied by its name, the Random Forest algorithm constructs a forest comprised of numerous decision trees. This supervised classification algorithm is esteemed for its rapid execution. A multitude of decision trees converges to create a Random Forest, and predictions are made by averaging the outcomes of each individual tree. Notably,

Dewangan & Agrawal [9] adopt a hybrid classification model by combining various classification methods, such as C4.5, Random Forest, and Multilayer Perceptron. Trained on a diabetes dataset from the UCI repository, their work focuses on the detection and classification of diabetes mellitus.

Devi & Shyla [10] delve into early prediction of diabetes using diverse machine learning techniques, including Naïve Bayes, Multilayer Perceptron, Random Forest, Random Tree, and Modified J48. Their analysis, based on the Indian PIMA dataset, reveals that the modified J48 classifier achieves the highest accuracy among the considered techniques.

In another research endeavor [11], the comparison of different classification algorithms, including Naïve Bayes, Multi-Layer Perceptron, J48, Random Forest, and regression, is undertaken to extract intelligent results from diabetic patient data.

Aishwarya and Vaidehi [12] employ a multitude of machine learning algorithms, such as Support Vector Machines, Random Forest Classifier, Decision Tree Classifier, Extra Tree Classifier, Ada Boost algorithm, Perceptron, Linear Discriminant Analysis algorithm, Logistic Regression, K-NN, Gaussian Naïve Bayes, Bagging Algorithm, and Gradient Boost Classifier. Their study, using the Indian PIMA and another diabetes dataset, reveals a logistic regression accuracy of 96%.

Tejas and Pramila, in a different approach, focus on two algorithms, logistic regression, and SVM, for diabetes prediction. Their preprocessing of data yields optimal results, with SVM exhibiting superior accuracy at 79%.

Yuvaraj and Sripreethaa [13] construct a diabetes prediction model using Random Forest, Decision Tree, and Naïve Bayes algorithms within Hadoop-based clusters. The application of preprocessing techniques results in a notable 94% accuracy with the Random Forest algorithm.

Deepti and Dilip [14] explore the Decision Tree, SVM, and Naive Bayes algorithms, employing a ten-way cross-validation for enhanced performance. Naïve Bayes emerges with the highest accuracy at 76.30%, using the Pima Indian Diabetes dataset.

In a study by Sajida et al. [15], the role of Adaboost and bagging ensemble machine learning methods using J48 as a base for classifying diabetes mellitus is discussed. The experiment demonstrates that the Adaboost ensemble machine learning technique outperforms the J48 decision tree in classifying patients as diabetic or non-diabetic based on diabetes risk factors.

3 Proposed Architecture

Datasets are sourced from Kaggle. During the second phase, the data undergoes pre-processing, encompassing cleaning, integration, and processing. Employing machine learning algorithms enhances accuracy in our findings (Fig. 1).

3.1 Patient Database

Data was gathered and systematically analyzed to construct a robust model. The dataset comprises pertinent and valuable information acquired through a thorough questioning process. This information is categorized meticulously, with a primary focus and further subdivision into narrowed categories.

include heart disease, stroke, kidney disease, and mortality [1]. According to the World Health Organization (WHO), the global prevalence of diabetes in adults over 18 years of age was 8.5% in 2014 [2]. Moreover, WHO predicts that by 2030, diabetes will become the seventh leading cause of death [2]. Predicting diabetes has become a crucial focus in health research, and contemporary computer models play a significant role in aiding decision-making and supporting self-management of the disease [3].

The increasing importance of machine learning in healthcare is evident, as these techniques consistently deliver high-performance accuracy results, simultaneously reducing human error in decision-making processes. Consequently, they alleviate the strain on healthcare resources [4]. Ideally, the further development of models incorporating prior knowledge would enhance the accuracy of diabetes prediction [5]. Access to health data from a patient's health records has the potential to extract meaningful information and unveil hidden knowledge.

This study aims to conduct a comparative evaluation of the performance of machine learning-based models for predicting diabetes. The prediction approaches were applied to datasets sourced from Kaggle. To the best of our knowledge, this study represents one of the most significant efforts to date in the realm of diabetes prediction. The paper is structured as follows: Sect. 1 provides a comprehensive overview of works utilizing learning techniques in the health domain, with a specific focus on applications related to diabetes. In Sect. 2, the architecture of the system, based on different algorithms, is explained. Section 3 presents the obtained results, offering a comparative analysis of the outcomes derived from the various algorithms employed.

2 State of the Art

Machine learning techniques have found application across various domains, with the medical field being no exception. Notably, several studies leverage machine learning classifiers to address medical challenges, with a particular emphasis on the chronic and complex nature of diabetes, a condition that has captured global research attention.

One noteworthy study by Fikirte Girma et al., titled "Prediction of diabetes using data mining techniques" [6], focuses on predicting diabetes operations through the application of the Back propagation rule. The findings indicate that Back propagation demonstrates superior accuracy in polygenic prediction compared to SVM, J48, and Naïve Bayes formulas.

In another study, Terry Jacob et al. present "Analysis of Supervised Learning Techniques for Cost Effective Disease Prediction" [7], which explores cost-effective disease prediction using non-clinical parameters. The study employs various algorithms, with Naive Bayes yielding an 80.37% accuracy, and REP trees achieving a maximum accuracy of 77%, especially when considering Logistic regression.

A methodology proposed by Butwall & Kumar [8] relies on the Random Forest (RF) classifier to understand the behavior of diabetes in alignment with specific lifestyle parameters, including physical activity and emotional states, particularly in elderly diabetics. The research, conducted on the Indian Pima diabetic database from the UCI Machine Learning Lab, reveals the effectiveness of RF in diagnosing diabetes mellitus based on provided attribute values.



Integration of Artificial Intelligence with Diabetic Data for Increasingly Personalized Medicine

Madiop Diouf^{1,3(✉)}, Thierno Amadou Diallo², Elhadji Ndiaye Diallo³,
Birahime Diouf³, and Ibra Dioum³

¹ USSEIN University, LITA ESP/UCAD, Kaolack BP 55, Dakar, Senegal
madiop.diouf@ussei.edu.sn

² UASZ University, LI3 Ziguinchor, Senegal
t.diallo@univ-zig.sn

³ LITA ESP/UCAD, Dakar, Senegal

Abstract. Diabetes is considered the most deadly and chronic disease that causes an increase in glucose. Polygenic disease is one in which the exocrine gland does not produce the hypoglycemic agent and according to the International Federation of Polygenic Diseases 382 million people live with polygenic disease in the world. By 2035, this number will double to 592 million. Diabetes mellitus or simply the disease can be a disease due to increased blood glucose levels. Many difficulties can arise if diabetes is not treated and not identified by the doctor. Thus, artificial intelligence (AI), which has become the new term we hear every day in recent years, generally defines the ability of a machine to act on its own and which is not explicitly programmed to reproduce actions or functions that are generally those of human beings. Today, we find it in our computing machines, social networks, transportation and in the medical sector etc. Therefore, machine learning is one of the disciplines of artificial intelligence that seeks to find a way to create computer programs that automatically improve with experience. In this work, we will focus on the use of machine learning algorithms for the prediction of diabetes, which is a dysfunction of the blood sugar regulation system, in order to reduce the risks of complications of this chronic disease on the health of the patient. To achieve this goal, we used machine learning algorithms such as Random Forest RF, Logistic Regression RL, K-nearest neighbors KNN and Neural Networks ANN and the data were extracted from Kaggle which is a web platform owned by Google that operates as a community for data scientists and developers. The performance of the classifiers was compared based on the accuracy rate.

Keywords: ANN · RF · RL · KNN · IA

1 Introduction

Diabetes is a prevalent and serious medical condition with widespread global impact. The severity of the disease is attributed to complications that arise when individuals either neglect to screen for diabetes or fail to receive appropriate care. Common complications

15. Schäfer, M., Nadi, S., Eghbali, A., Tip, F.: Adaptive test generation using a large language model, arXiv preprint [arXiv:2302.06527](https://arxiv.org/abs/2302.06527) (2023)
16. Xia, C.S., Wei, Y., Zhang, L.: Automated program repair in the era of large pre-trained language models, In: Proceedings of the 45th International Conference on Software Engineering (ICSE 2023). Association for Computing Machinery (2023)
17. Xia, C.S., Zhang, L.: Conversational automated program repair, arXiv preprint [arXiv:2301.13246](https://arxiv.org/abs/2301.13246) (2023)
18. Sobania, D., Briesch, M., Hanna, C., Petke, J.: An analysis of the automatic bug fixing performance of chatgpt, arXiv preprint [arXiv:2301.08653](https://arxiv.org/abs/2301.08653) (2023)
19. Feng, S., Chen, C.: Prompting is all your need: Automated android bug replay with largelanguage models, arXiv preprint [arXiv:2306.01987](https://arxiv.org/abs/2306.01987) (2023)
20. Sobreira, V., Durieux, T., Madeiral, F., Monperrus, M., Maia, M.A.: Dissection of a bug dataset: Anatomy of 395 patches from defects4j, In: Proceedings of SANER (2018)

reports into formal test case specifications. The promising results, in terms of *executability* (i.e., the test case is syntactically correct), and *{validity}* (i.e., the test case actually makes the program fail), suggest that ChatGPT can “*understand*” the semantics of bug reports. This finding is essential as it opens new research directions with large language models, towards automating test case generation with a human in the loop (for writing bug reports).

Acknowledgments. This work was conducted as part of the Artificial Intelligence for Development in Africa (AI4D Africa) program, with the financial support of Canada’s International Development Research Centre (IDRC) and the Swedish International Development Cooperation Agency (Sida).

References

1. Bissyandé, T.F., et al.: IEEE 24th international symposium on software reliability engineering (ISSRE). IEEE **2013**, 188–197 (2013)
2. Lopez, A.: Statistical machine translation. ACM Computing Surveys (CSUR) **40**, 1–49 (2008)
3. Stahlberg, F.: Neural machine translation: A review. Journal of Artificial Intelligence Research **69**, 343–418 (2020)
4. Allamanis, M., Barr, E.T., Devanbu, P., Sutton, C.: A survey of machine learning for big code and naturalness. ACM Computing Surveys (CSUR) **51**, 1–37 (2018)
5. Hu, X., Li, G., Xia, X., Lo, D. Jin, Z.: Deep code comment generation, In: Proceedings of the 26th conference on program comprehension, pp. 200–210. (2018)
6. Goues, C.L., Pradel, M., Roychoudhury, A.: Automated program repair. Commun. ACM **62**, 56–65 (2019)
7. Monperrus, M.: Automatic software repair: A bibliography. ACM Computing Surveys (CSUR) **51**, 1–24 (2018)
8. Gulwani, S., Polozov, O., Singh, R., et al.: Program synthesis, Foundations and Trends®. Programming Languages **4**, 1–119 (2017)
9. Le, X.-B.D., Bao, L., Lo, D., Xia, X., Li, S., Pasareanu, C.: On reliability of patch correctness assessment, In: 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), IEEE, pp. 524–535. (2019)
10. Xiong, Y., Liu, X., Zeng, M., Zhang, L., Huang, G.: Identifying patch correctness in test-based program repair, In: Proceedings of the 40th international conference on software engineering, pp. 789–799. (2018)
11. Anand, S., et al.: An orchestrated survey of methodologies for automated software test case generation. J. Syst. Softw. **86**, 1978–2001 (2013)
12. Taneja, K., Xie, T.: Diffgen: Automated regression unit-test generation, In: 2008 23rd IEEE/ACM International Conference on Automated Software Engineering, IEEE, pp. 407–410. (2008)
13. Thummalapenta, S., Xie, T., Tillmann, N., De Halleux, J., Schulte, W.: Mseqgen: Object-oriented unit-test generation via mining source code, In: Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, pp. 193–202. (2009)
14. Fraser, G., Arcuri, A.: Evosuite: On the challenges of test case generation in the real world. IEEE sixth international conference on software testing, verification and validation IEEE **2013**, 362–369 (2013)

section will cover some limitations and suggest some research directions for future work to increase the amount of executable and valid test cases.

5.1 Threats to Validity

In this study, ChatGPT, an openly available LLMs known to have been trained on public data is exploited to generate test cases for bugs of the Defects4J dataset. As Defects4J is a standard benchmark that is publicly available, it is likely that parts of the dataset have been part of the model's training data, this is a threat to the validity of the results as it reflects a data leakage problem. To address this concern, we performed manual investigations of the generated test cases to ensure their difference with the original tests, confirming ChatGPT's capability of understanding the semantics of the bug report. Still, in further iterations of this work we will assess ChatGPT's performance on newly reported faults. Additionally, due to the randomness of ChatGPT, it is essential to verify that ChatGPT is correctly replying to the given prompt before starting the experiments.

5.2 Limitations & Future Work

In this study the *Executability* only reflects if a generated test case is directly executable or not. This doesn't reflect the amount of effort for a human to make it executable. After manually reviewing the generated test cases, it has been shown that most can be made executable through the modification of one or two lines of code. The most common issues are usually: missing imports, duplicate function names, or the use of a deprecated function. Those limitations could systematically be fixed in future work (e.g. with prompt engineering) significantly increasing the amount of executable test cases.

For future iterations of this study we suggest to investigate the potential of *fine-tuning LLMs* to translate the informal bug reports into formal test cases with a higher rate of executable and valid test cases.

In our experiments *bug reports* were collected and directly used as prompt to demonstrate the feasibility and applicability of our idea to address real software faults reported by the users. Nevertheless, pre-processing the textual data might be beneficial to keep ChatGPT focused on the main context of the bug report, therefore increasing the executability and validity of the generated test case.

To fix a bug, the first step is to reproduce it with a bug-triggering test case which has been proven feasible in our study. To reach to directly fix a newly reported bug, further research should be made on how we can improve and automatically deploy APR tools to not just automatically address the bug but fix it.

6 Conclusion

Large language models have recently gained substantial popularity, thanks to the public release of ChatGPT, whose potential as a disruptive technology has been largely advertised. The literature has empirically studied various capabilities of the inner language model for various natural language processing tasks. In this work, we investigate the feasibility of leveraging the inner language of the GPT model to translate informal bug

of some Math project versions is that the test suite is compiling, but their execution is never-ending. Therefore, there were 10 Math bug reports for which we could not determine the *validity* of the generated test cases, which explains the lower validity results for this project. Overall, we got valid test cases for 30% of the bug reports, which is really promising. Among the executable test cases, we observed that 59% were valid. This shows that once the initial *executability* challenge is passed, the tests generated by ChatGPT are in fact valid, highlighting its understanding. Meaning that ChatGPT was able to understand the semantics of the user written bug report and translate them into a bug-triggering test case. These results highly motivate further research in this domain.

Table 2. Generation performance for ChatGPT: percentage of bug reports where we successfully generated at least one test case.

Project	# of bug reports	Percentage of generation success		
		Overall executability	Overall validity	Validity among executable
Chart	6	33%	17%	50%
Cli	30	53%	37%	69%
Closure	127	46%	28%	59%
Lang	60	60%	43%	72%
Math	100	43%	15%	35%
Time	19	84%	68%	81%
Total	342	50%	30%	59%

Further manual investigations also highlighted that *executability* and *validity* can in most cases be fixed with minor modifications (e.g., adding relevant imports or changing duplicated function names).

Findings: The experimental results based on ChatGPT APIs show that a large language model (LLM) can take bug reports as inputs and produce test cases that are executable in 50% of cases. Beyond *executability*, about 30% of the bugs could be reproduced with valid test cases. Specifically, over half (59%) of the executable test cases were valid test cases.

These results, which are based on an off-the-shelf LLM as-a-service, show promises for automated test case generation, beyond unit testing, leveraging complex information from user-reported bugs.

5 Discussion

Overall, our empirical study validates the hypothesis that ChatGPT can “understand” bug reports: given a bug report, it can extract its semantics and translate them into formal test cases. However, some challenges that should be addressed in the future remain. This

The instruction is unique for all queries to ChatGPT and is as simple as follows: “{write a Java test case for the following bug report:}”. For the bug report, our feasibility study considers that no pre-processing should be applied on the bug report, and the information should not include follow-up comments or attachments. For every prompt, we request the ChatGPT five (5) times and assess the different generated test cases. In practice, before running the generated test cases, the ChatGPT outputs are parsed to clean them from natural language texts (e.g., explanations) that would lead to compilation failures. Afterwards, the test cases are systematically included in the test suite, which is fully executed by the Defects4J test pipeline. Execution results are then logged, allowing us to compute the metrics of executability and validity.

4 Results

Figure 1 provides an illustrative example of a bug report (from the CLI project) and the associated formal test cases (ground truth in Defects4J and generated from ChatGPT). As we can see in this example, ChatGPT is able to reproduce a formal test case from a bug report, which can enable various software automation tasks, such as spectrum-based fault localization, patch validation in program repair, and more generally automated software testing.



Fig. 1. Example of bug report and associated test cases - Cli 17 from Defects4J

On the Defects4J dataset, we compute the proportion of bug reports for which ChatGPT is able to successfully generate test cases.

Table 2 summarizes the metrics. On average, executable test cases were obtained for 50% of the bug reports across all projects. The validity of the generated test cases varies greatly from one project to another, which can be explained by the different source and format of user-written bug reports. Unfortunately, a commonly known issue

3 Experimental Setup

In this section we overview the settings under which we assess the capability of a Large Language Model to translate informal bug reports from real software projects into formal test case specifications that reproduce the buggy behavior. In particular, we present the benchmark, the metrics as well as the experimental design.

3.1 Benchmark

We consider the **Defects4** repository [20], which includes real-world faults from various Java software development projects as enumerated in Table 1. We collect the bug reports associated to these faults. One must mention that not all bug reports were available, and some bugs referred to the same bug report, in that case it was only considered once to avoid bias in the results because of duplicates. We considered Defects4J due to its wide adoption in the software testing and software research communities.

Table 1. Java Projects from Defects4J used in the study.

	Projects					
	Chart	Cli	Closure	Lang	Math	Time
#Faults	26	39	174	64	106	29
#Faults associated to Bug Reports	6	30	127	60	100	19

3.2 Metrics

As introduced, the goal of the study is to assess whether ChatGPT can generate test cases from bug reports. Therefore, our evaluation is focused on measuring the quality of the generated test cases. We thus consider two main metrics:

- **Executability:** a ChatGPT-generated test case may not even be syntactically correct to be compiled and executed. While often, the generated test case can be made executable after manually implementing small edits (e.g., adding relevant imports), we conservatively consider that executability is a binary metric, and is automatically computed once ChatGPT outputs are yielded (with no manual changes added).
- **Validity:** an executable test case may or may not fail on the target buggy program. We follow the convention of patch validation in program repair and consider the generated test case to be valid only when it, indeed, fails on the buggy program. Otherwise it is considered as invalid.

3.3 Experimental Design

We rely on the ChatGPT API (version 3.5) for our experiments. We construct the prompt by concatenating two pieces of information: the instruction and the bug report.

With recent advances in natural language processing techniques, such as the advent of large language models (LLMs), a wide range of tasks have seen machine learning achieve, or even exceed, human performance. Machine translation [2, 3], in particular, has been a very active field where several case studies have been explored beyond language translation. For example, in software engineering, several research directions have investigated the feasibility of leveraging natural language inputs for producing programming artefacts and vice-versa. Some milestones have been recorded in the literature in code summarization [4, 5], program repair [6, 7], and even program synthesis [8]. Nevertheless, bug reports have scarcely been explored. Yet, automating bug reproduction via analysis of bug reports holds tremendous value. In this work, we propose to *study the feasibility of exploiting an LLM for reproducing bugs*. We focus on ChatGPT, which has recently received much attention and presents the advantage that its model has been trained on a large corpus of natural language text as well as source code of software programs.

But *can ChatGPT understand bug reports*? We consider the management of bug reports as an example case where machine learning can be helpful while keeping the human in the loop. “Understanding bug reports” suggests the eventual possibility of reproducing the reported bug. Our prompt is therefore focused on requesting ChatGPT to exploit a bug report’s textual content (in natural language) and generate a formal test case (in a programming language). We assume that if ChatGPT can generate a test case that not only is executable but also fails on the associated buggy program version, then ChatGPT may have “understood” (in the sense of “*captured the semantics of unwanted execution behavior reported by the user*”). This assumption is, obviously, an over-approximation of the relevance of the generated test case since the generated failing test case may be based on random inputs that are irrelevant to the reported bug. Nevertheless, it would constitute a first milestone towards automatic test case generation based on user inputs, which reflects realistic and complex user experience.

2 Related Work

To address a bug, the first step is to understand it and reproduce it. To demonstrate that there is a bug, the developer has to write a bug-triggering test case since the original test suites are usually scarce and incomplete [9, 10]. Several techniques [11–14] have been developed to help developers with this very time-consuming process. However, these techniques mostly rely on formal specifications.

Recently, some work on test case generation using large language models (LLM) was done by [15] but their TestPilot still requires the functions signature and implementation as prompt. The use of ChatGPT to directly enhance Automated Program Repair (APR) techniques [16–18] highlights even more the potential of this new LLM. Feng et al. [19] have investigated ChatGPT’s ability to help developers reproduce the bug while extracting important steps to reproduce the bug from the bug report. However they did not perform any test case generation. Additionally, they still require a human to actually reproduce the bug. In contrast, in our study, we are aiming at using the unprocessed human written bug report as direct input for test case generation, enabling automatic bug reproduction with only a human-in-the-loop to report the bug but not to address it.



Leveraging Conversational AI for Accelerating User-Driven Software Testing

Aminata Sabané^{1,2}, Laura Plein³, and Tegawendé F. Bissyandé^{1,2,3}(✉)

¹ CITADEL Interdisciplinary Excellence Centre in Artificial Intelligence for Development,
Ouagadougou, Burkina Faso

aminata.sabane@ujkz.bf, tegawende.bissyande@citadel.bf

² Université Joseph Ki-Zerbo, Ouagadougou, Burkina Faso

³ SnT, Université du Luxembourg, Luxembourg, Luxembourg

laura.plein@men.lu

Abstract. This work addresses a research challenge in automating the translation of natural language inputs into programming language specifications. We consider the case of bug reports, which are informally written by users, and that must be specifying into executable test cases for reproducing the bug on the target software. Software bugs are indeed largely reported in natural language by users. Yet, we lack reliable tools to automatically address reported bugs (i.e., enabling their analysis, reproduction, and bug fixing). We therefore build on the recent promises brought by ChatGPT for various tasks, including in software engineering, and establish the following research question: *What if Conversational Artificial Intelligence (AI) models could be used to explore the semantics of bug reports as well as to automate their reproduction?* We evaluate the capabilities of ChatGPT, a state-of-the-art conversational AI, i.e., chatbot, using the popular Defects4J benchmark with its associated bug reports. The results reveal that ChatGPT can generate executable test cases that could trigger 50% of the bugs reported in natural language. These results are promising not only for the research community, but also for practitioners.

Keywords: ChatGPT · Debugging · Translation · Test cases · Bug reports

1 Introduction

Software users are expected to provide feedback on their experience in running programs. Such feedback often leads to various improvements by developers responding to feature requests and bug reports. In this respect, development platforms, such as GitHub, offer tool support for collecting reports and continuously monitoring how developers address them. Unfortunately, various studies have shown that bug reports are under-exploited [1]. Recurrently, indeed, researchers and practitioners point to the general quality of such reports: developers put much effort to “understand” and reproduce the potential bugs that are reported; researchers struggle to build tools for automatically capturing the semantics of the natural language text and transforming them into actionable inputs for existing (testing) frameworks.

Artificial Intelligence

11. Hahn, C.: A Domain Specific Modeling Language for Multiagent Systems, In: présenté à 7th Int. Conf. on Autonomous Agents and Multiagent Systems, Estoril, Portugal, pp. 233–240. (2008)
12. Santos, F., Nunes, I., Bazzan, A.L.C.: Supporting the Development of Agent-Based Simulations: A DSL for Environment Modeling, In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), pp. 170–179. (2017). <https://doi.org/10.1109/COMPSAC.2017.224>
13. Phung, S., et al.: An Agent-Based Modeling and Simulation Environment for Dynamic Biological Systems, In: Proceedings of the International Conference on Bioinformatics & Computational Biology (BIOCOMP), The Steering Committee of The World Congress in Computer Science, Computer p. 349. (2012)
14. Pavón, J., Gómez-Sanz, J.: Agent oriented software engineering with INGENIAS, In: International Central and Eastern European Conference on Multi-Agent Systems, Springer, pp. 394–403 (2003)
15. Gago, I.S.B., Werneck, V.M., Costa, R.M.: Modeling an educational multi-agent system in maSE. In: International Conference on Active Media Technology, Springer, pp. 335–346 (2009)
16. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design. *Auton. Agents Multi-Agent Syst.* **3**, 285–312 (2000)

The overall proposition takes account of the model modelling and implementation, model simulations, data management and visualization in relation with the agroecosystem domain. The overall proposition allows for agroecosystem experts to easily design agroecosystem models.

In terms of the way forward, a programming language will be proposed for agroecosystem-oriented model development. In addition, the detail design and implementation of the independent platform will be done. Finally, a methodology will be proposed as a guideline for the design and simulation of agroecosystem models.

References

1. Jahel, C. : Analyse des dynamiques des agroécosystèmes par modélisation spatialisée et utilisation d'images satellitaires, cas d'étude de l'ouest du Burkina Faso. PhD Thesis, AgroParisTech (2016)
2. Kremmydas, D., Athanasiadis, I.N., Rozakis, S.: A review of Agent Based Modeling for agricultural policy evaluation. *Agric. Syst.* **164**, 95–106 (2018). <https://doi.org/10.1016/j.agry.2018.03.010>
3. Belem, M., Saqalli, M.: Development of an integrated generic model for multi-scale assessment of the impacts of agro-ecosystems on major ecosystem services in West Africa. *J. Environ. Manage.* **202**, 117–125 (2017). <https://doi.org/10.1016/j.jenvman.2017.07.018>
4. Belem, M., Bazile, D., Coulibaly, H.: Simulating the Impacts of Climate Variability and Change on Crop Varietal Diversity in Mali (West-Africa) Using Agent-Based Modeling Approach. *Journal of Artificial Societies and Social Simulation* **21**(2), 8 (2018). <https://doi.org/10.18564/jasss.3690>
5. Grignard, A., Taillandier, P., Gaudou, B., Vo, D.A., Huynh, N.Q., Drogoul, A.: GAMA 1.6: Advancing the Art of Complex Agent-Based Modeling and Simulation, In: PRIMA 2013: Principles and Practice of Multi-Agent Systems, Boella, G., Elkind, E., Savarimuthu, B.T.R., Dignum, F., Purvis, M.K. (eds.), *Lecture Notes in Computer Science*. pp. 117–131. Berlin, Heidelberg: Springer, (2013). https://doi.org/10.1007/978-3-642-44927-7_9
6. Bousquet, F., Bakam, I., Proton, H., Le Page, C.: Cormas: Common-pool resources and multi-agent systems. In: *Tasks and Methods in Applied Artificial Intelligence*, Pasqual Del Pobil, A., Mira, J., Ali, M., (eds.), in *Lecture Notes in Computer Science*, vol. 1416, pp. 826–837. Berlin, Heidelberg: Springer Berlin Heidelberg (1998). https://doi.org/10.1007/3-540-64574-8_469
7. Lytinen, S.L., Railsback, S.F.: The Evolution of Agent-based Simulation Platforms: A Review of NetLogo 5.0 and ReLogo
8. Czarnecki, K., O'Donnell, J.T., Striegnitz, J., Taha, W.: DSL Implementation in MetaOCaml, Template Haskell, and C++, In: *Domain-Specific Program Generation: International Seminar, Dagstuhl Castle, Germany, March 23–28, 2003. Revised Papers*, Lengauer, C., Batory, D., Consel, C., Odersky, M., (eds.), in *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, p. 51–72. (2004). https://doi.org/10.1007/978-3-540-25935-0_4
9. Gérard, S., Dumoulin, C., Tessier, P., Selic, B.: 19 Papyrus: A UML2 Tool for Domain-Specific Language Modeling, In: *Model-Based Engineering of Embedded Real-Time Systems*, Springer, Berlin, Heidelberg, pp. 361–368. (2010). https://doi.org/10.1007/978-3-642-16277-0_19
10. Nassar, M., et al. : Vers un profil UML pour la conception de composants multivues, *Rev. Sci. Technol. Inf.-Sér. Obj. Logiciel Bases Données Réseaux*, 11(4), (2005)

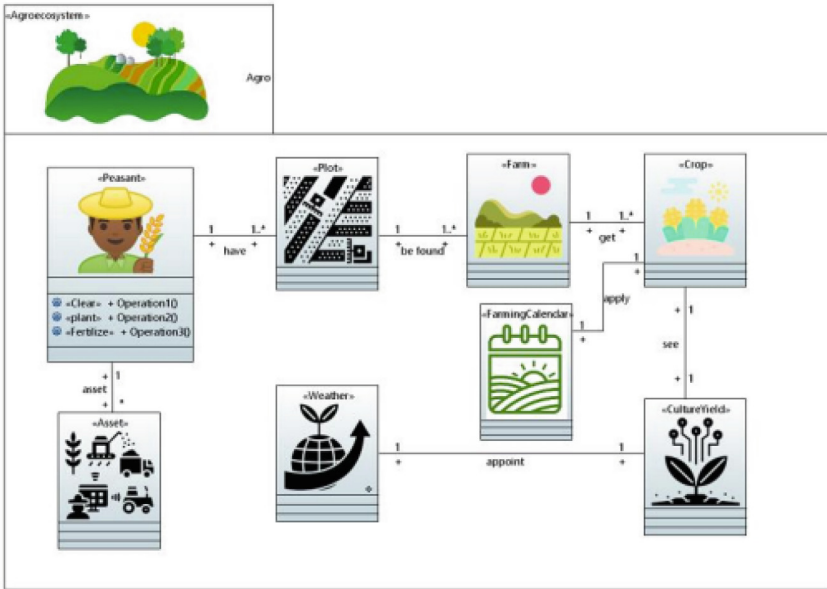


Fig. 4. Application of the graphical notation language in the modelling of an agroecosystem

None of these propositions take account of agroecosystem modelling and simulation. They do not propose read-to-use concepts in the agroecosystem domain. In addition, although these previous studies propose a high level abstract syntax, they do not propose any methodology for MAS and ABM modelling and development.

A range of methodologies [14–16] propose guidelines for MAS and ABM design. Furthermore, they propose an abstract syntax for multi-agents and ABM development. However, these methodologies do not propose read-to-use concepts for agroecosystem modelling and simulation, making difficult their use by an agroecosystem expert.

This work concerns the development of DSL for agroecosystem-oriented models creation and simulation. The main aim is to develop read-to-use concepts for modelling and simulating agroecosystems. The abstract syntax and graphical notation language developed allows for an agroecosystem expert to design easily agroecosystem models. At this current stage, however, account is not taken of the code generation and simulation. This aspect will be delved into in future work.

5 Conclusion

The main objective of this study was to define a basis for the development of an independent platform for modelling and simulating agroecosystem-oriented models. Originally, we aimed to propose a meta-model, a UML profile, a domain specific modelling language and a general architecture. To this end, we used an approach based on DSL and agent-based modelling.

Table 1. Graphical notation of the different concepts

Stereotypes (concepts)	Meta-class	Graphical notation
Household	Agent	
Agroecosystem	Package	
Farm	Class	
Plot	Cell	
Cropping System	Class	
Crop	Class	
Farming Calendar	Agent	
Climate Agent	Class	
Weather	Class	
Farmer	Role	
Breeder	Role	
...	...	...

4 Discussion

The development of DSL in the multi-agents system and agent-based modelling is not new. A range of works have already been carried out in this domain. In this regard, [11] proposed a domain specific modelling language for multi-agent systems that provides a clear syntax and semantics to define agent-based systems in a graphical visualized manner. However, this DSL has been proposed for general purpose in the multi-agent system domain. The use of this DSL is difficult for non-expert multi-agents, particularly in a specific domain such as agroecosystem. [12] proposed a DSL to target the agent-based modelling and simulation domain with a focus on modelling the simulated environment. In order to allow biologists to create and simulate models easily, [13] proposed GRAN-ITE (Genetic Regulatory Analysis of Networks Investigational Tools Environment), for large-scale model simulation in the domain of biology.

3.2 UML Profiles

In order to allow for graphical modelling and code generation, a UML profile has been built respecting the meta-model. Each sub-domain of the meta-model is then represented as a UML package. *Simulation*, *Agent*, *Agroecosystem*, *Persistence* and *Visualization* packages are considered as stereotypes extending the UML package. Each concept of the meta-model has later been considered as a stereotype. In this way, the concept of *Model*, *Simulation*, *Scenario*, *Parameter*, *Output* extends the UML Class. In the agent package, the concept of *Agent*, *Role*, *Capability*, *Resource* and *Location* extend the UML class. As for the agro-ecosystem package, *Household*, *Climate* and *Animal* extend the *Agent* stereotype. In the same package, *Farmer*, *Breeder* and *Forestry* extend the *Role* concept. Finally, *Clear*, *Plant*, *Cultivate*, *Transport*, *Stock*, *Sell*, *Buy* extend the *Capability* concept.

3.3 Graphical Notation Language for Agroecosystem Modelling

Based on the proposed profile, a graphical notation language for agroecosystem modelling has been proposed. The graphical language built in this study concerns only the agroecosystem package and the related UML profile. To each main concept of the package, a visual representation has been associated. Table 1 provides the description of some visual representations used in this study, while Fig. 4 provides an example of the use of graphical language.

3.4 Architecture of the Platform

A multi-layer architecture has been proposed for the platform development. This architecture describes the overall structure and functionalities of the platform. The main layers of the architecture are:

View layer: provides interfaces to manage interactions between users and the frameworks for models development and simulation and to visualize the simulation outputs.

Modelling layer: provides tools and utilities for graphical modelling and model implementation using a DSL. it contains as principal components the DSML and a DSL. it should also allow for code generation.

Simulation layer: provides tools and utilities for model simulation. it should implement the simulation, agent and agroecosystem sub-domains of the meta-model.

Utilities layer: provides tools and utilities for data management and data visualization. This layer should help to manage interaction with different formats of data supports (database, GIS data, CSV, txt, JSON, XML, etc.) and to visualize data (map, chart, file, video, etc.). It implements the persistence and visualization sub-domains of the meta-model.

The ***Climate*** agent computes the climate condition. It is characterized by different climate conditions and their probability distribution.

In this system, the *Household* agent plays the following roles: *Farmer*, *Breeder* and *Forestry role*. In terms of *Capabilities*, a *Household* can, among other things, *Clear*, *Plant*, *Cultivate*, *Transport*, *Stock*, *Sell*, *Buy*, etc. The *Climate* agent plays the *Weather* role.

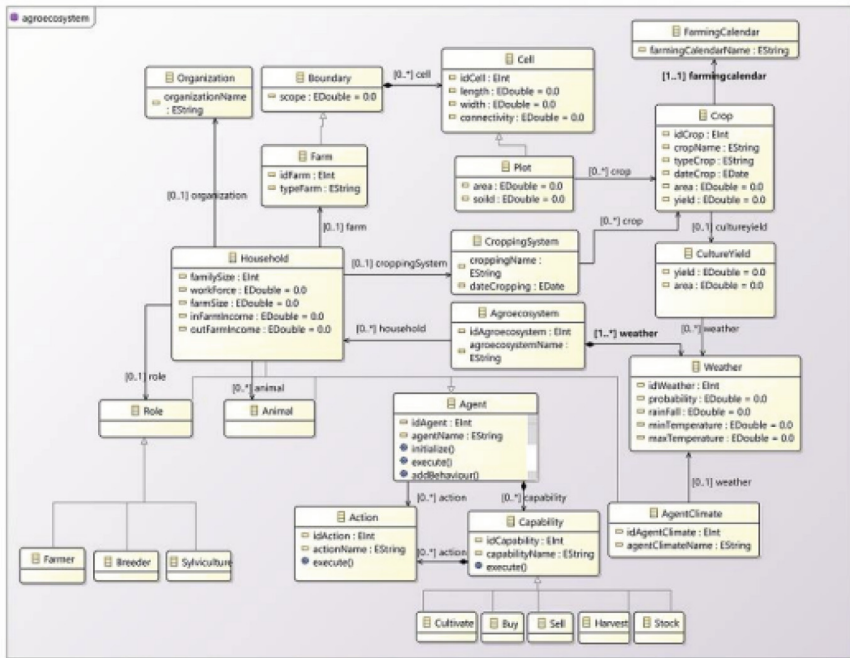


Fig. 3. The meta-model of agro-ecosystem sub-domain

The «Persistence» sub-domain covers the collection, storage and archiving of data, as for example, the interactions between models and data sources, such as ontologies, DSLs, GIS and Big Data.

The «Visualization» sub-domain is responsible for simulation results visualization.

“Simulation” Sub-domain

The « Simulation » sub-domain describes the core concepts for a general description of simulation models. Then, a **Model** is characterized by its name, a set of **Agents**, **Scenarios**, a set of **Output** variables. A **Model** is used to achieve Simulation. A Simulation is characterized by a range of **Scenarios**. A **Scenario** is characterized by a set of parameters, each parameter being used to initialize a **Variable**. An **Output** variable is related to a mode of **Visualization**.

“Agent” Sub-domain

The agent sub-domain provides the description of the core concepts for representing a multi-agents system (Fig. 2). In this sub-domain, An **Agent** is characterized by a set of **Roles**, **Capabilities**, **Acquaintances** and a **Location in the Environment**. A **Role** defines the behavior of an agent in the system and a **Capability** defines a task that an agent can achieve. An **Environment** is a cellular automaton characterized by a set of **Cells**.

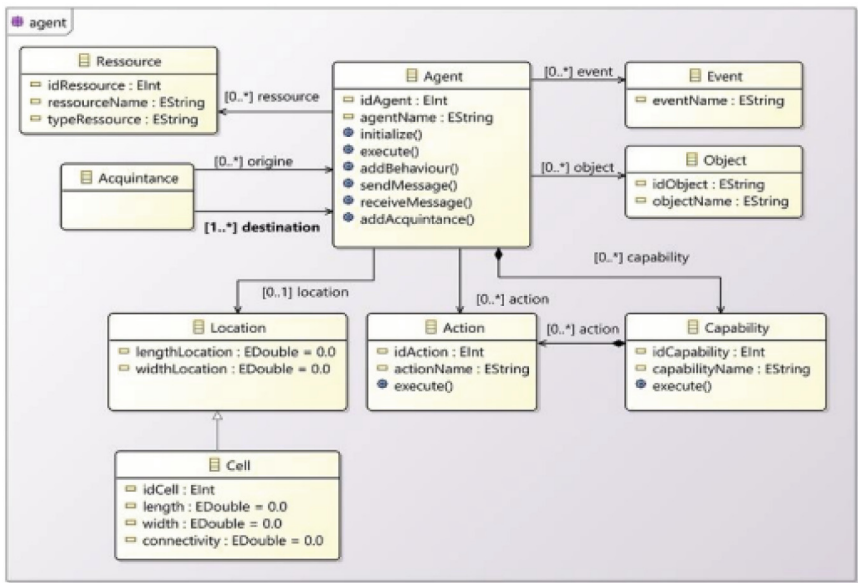


Fig. 2. The meta model of agent sub-domain

“Agroecosystem” Sub-Domain

The agroecosystem sub-domain describes the core concepts required to represent an agroecosystem based on agent-based point of view (Fig. 3). The agrosystem sub-domain is an extension of the agent sub-domain.

An **Agroecosystem** is a model characterized by three types of agent: **Household**, **Climate** and **Animal** agents. A **Household** has a **Farm** characterized by a set of **Plots**.
A

2.5 Fifth Step: DSML Development

Using the meta-model and UML profile previously developed, a DSML has been developed. The DSML proposes a graphical notation language for agro-ecosystem oriented model design.

2.6 Sixth Step: Architecture Analysis

Finally, a multi-layer architecture is proposed for the development of the independent platform. It proposes an overall description of the framework structure and behavior.

3 Results

As regards the main results, this study allowed to develop an abstract syntax language, a UML profile, a graphical notation language and to propose an architecture for the development of an independent platform. These results are described in the following section.

3.1 Abstract Syntax Language

For a better representation of the model, our system has been organized under five sub-domains: “Persistence”, “Simulation”, “Agent”, “Agroecosystem” and “Visualization” sub-domains (Fig. 1). For each sub-domain, the main concepts have been identified and a meta-model developed.

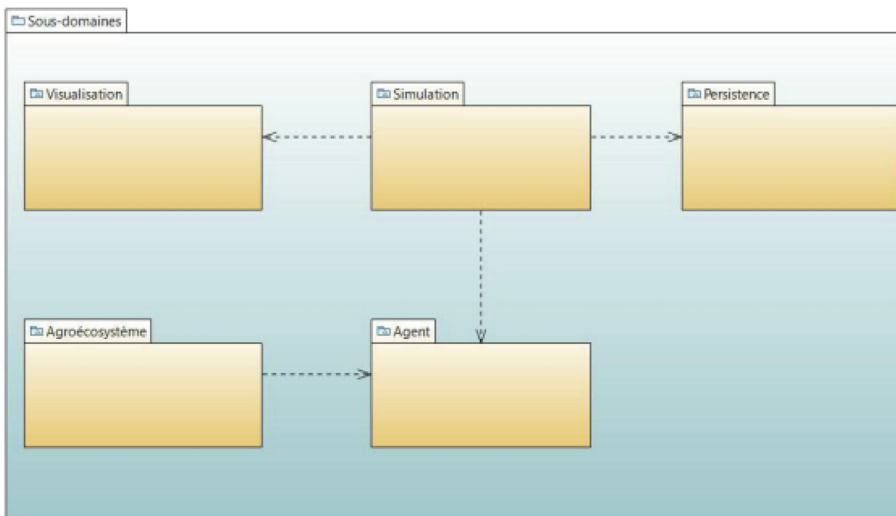


Fig. 1. The different parts of the meta-model

taken at how an independent platform could be put in place for agroecosystem-oriented models development and simulation. Through this platform, we aim to automatize the development and simulation of models. In addition, this platform should make it possible to interact with external data sources (GIS, database, CSV file, Excel file, etc.) and to visualize simulation results. Specifically, the platform should include:

- a graphical language for agroecosystem-oriented model design;
- a domain specific language as programming language;
- tools for visualizing simulation results;
- and tools for models simulations.

2.2 Second Step: Domain Analysis

The main objective of this step is to understand the domain under study through the identification of the main concepts of study. Drawing from the study specification, the literature review and expert judgment, the domain analysis step helped to identify the main concepts. These have been identified to take account of the model simulation and visualization, agroecosystem models development using an agent-based modelling approach.

2.3 Thirst Step: Development of the Meta-model

Based on the main concepts previously identified, a meta-model has been developed as follows: To start with, we defined the abstract syntax of our language through the delineation of meta-models, using domain concepts (i.e. domain vocabulary) and the relationships between these concepts. To this end, we used the papyrus tool [9].

2.4 Fourth Step: Creating the UML Profile

Based on the meta-model, a UML profile has been created. A papyrus tool [8] was used at this stage. Firstly, we created i) stereotypes using the domain concepts, ii) the list of tagged values and iii) constraints that can be expressed in OCL (Object Constraint Language) on stereotypes and tagged values.

The UML profile is an extension of UML meta-model for a specific domain. In fact, UML offers a number of writing rules or standardized graphical representations, as well as common mechanisms or concepts applicable to all diagrams. Certain elements, such as stereotypes, are specifically designed to ensure the adaptation and evolution of the notation, mainly to take into account the particularities of the different situations to be modelled. Compared to the standard UML formalism, developers often wish to add extra characteristics to consider certain specificities of their application domain [10]. But UML does not meet all needs, because there are few tools for analysing specifications and not enough concepts from certain domains. To meet this need, UML features an extensibility mechanism based on stereotypes, tagged values and OCL constraints. This extension mechanism makes it possible to particularize the UML meta-model to accommodate specific modelling needs [10].

interacting at different scales of description. In an ABM, the agent has its own decision model. The agent can perceive its environment and acts on it, which in turn may impact its decision-making process. These ABM characteristics are relevant to represent the agroecosystem. For example, [3] used agent-based model to represent the impact of cropping systems on soil carbon sequestration and household income. [4] used multi-agents system to represent the impacts of climate change and variation on crop diversity management by the farmers.

However, although agro-ecosystem modelling is becoming more and more important, a specific modelling framework is missing not only for modelling agroecosystems but also for their simulation. Currently, modellers use general modelling and simulation platforms, such as Gama [5], Cormas [6], NetLogo [7], etc. that non-modellers and non-agent experts find difficult to use.

The objective of this study is to propose an independent platform for agent-based modelling and simulation of agroecosystems. Specifically, this study, the first step in developing the general framework, aims to propose a meta-model of agroecosystem, a graphical notation language for agroecosystem-oriented modelling and to design the architecture of the platform.

In this study, we used an approach based on domain-specific language (DSL) and agent-based modelling. A DSL is a language offering expression power focused on a particular domain, such as a specific class of applications or an aspect of the application [8]. A DSL is a programming language for a specific domain. It is opposite to classical programming language, such as Java, C++ and C#, used for a range of domains. In this study, we are using DSL in order to propose a new programming language for agroecosystem-oriented model development. The objective is to overcome the difficulties related to the development of a model using general programming language and non-dedicated programming language.

As a result, a meta-model is proposed as the DSL abstract syntax. Thereafter, the language has been concreted by proposing a graphical notation language. Finally, a multi-layer architecture is proposed. The overall proposition takes account of the model design, simulations and their visualization.

This paper is organized as follows: The second section describes the general approach and tools used in the study. The last section describes the main results arrived at in the study. Specifically, this section describes the meta-model for agro-ecosystem modelling, the UML profile and the DSML. Finally, the results are discussed.

2 Materials and Method

Our methodology involves six steps, including the study specification, the domain analysis, the UML profile development, the DSML development and the architecture design.

2.1 First Step: Specification

The first step aims is to define the scope of our study. Specifically, the objective was to define the goal of the platform and the intended use of the platform. Then, a look is



A Domain Specific Language (DSL) for Agroecosystems Modelling and Simulation

Jean-Armand Yanogo^{1,2}(✉), Mahamadou Belem^{1,2}, Toundé Mesmin Dandjinou^{1,2},
Saïd Cham's Nour Ougda^{1,2}, and Theodore Marie Yves Tapsoba^{1,2}

¹ Université Nazi Boni, Bobo-Dioulasso, Burkina Faso
Jeanarmand_yanogo@yahoo.fr

² Laboratoire d'Algèbre, de Mathématiques Discrètes et d'Informatique (LAMDI),
Université Nazi Boni, Bobo-Dioulasso, Burkina Faso

Abstract. Modelling agroecosystems is a complex process that implies understanding the interactions between the different elements of the system. However, although agroecosystem modelling is becoming more and more important, a specific modelling framework is missing not only for the modelling of agroecosystems but also for their simulation. Currently, modellers use general modelling and simulation platforms that non-modellers find difficult to apply. Consequently, a specific framework for agroecosystem modelling and simulation is required. This study intends to propose a basis for the development of an independent platform for the creation and simulation of agroecosystem-oriented models. Specifically, the objectives of this paper are to achieve the requirement analysis and the domain analysis, to propose a domain specific modelling language and to design the platform architecture. Using a model-driven engineering and an agent-based modelling approach, a meta-model has been proposed as the abstract syntax of the language. Thereafter, the language has been concretized by proposing a graphical notation language. Finally, a multi-layer architecture has been proposed. The overall proposition takes account of the model development, simulations and their visualization. The general framework will be developed as part of the next steps.

Keywords: agroecosystem · simulation · agent-based model · meta-model · domain specific language · domain specific modelling language

1 Introduction

Agroecosystems are cultivated ecosystems composed of abiotic and biotic elements that interact with each other in an agricultural, pastoral and forestry-type space, modified by man for the purpose of food production [1]. Agroecosystems are characterized by a dynamic and structural complexity coming from the interaction between the ecological and socio-economic processes in which they are integrated. So modelling agroecosystems is a complex process that implies understanding the interactions between the different elements of the system. Agent-based models (ABMs) are highly relevant for representing and modelling agro-ecosystems [2]. ABMs comprise a set of agents

15. Jiang, H., Liang, Y.: Online path planning of autonomous Drones for bearing-only stand-off multi-target following in threat environment. *IEEE Access* **6**, 22531–22544 (2018)
16. Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J.: Multi agent cooperation and competition with deep reinforcement learning. *PLoS ONE* **12**, 6379–6390 (2017)
17. Wu, C., et al.: DRONE autonomous target search based on deep reinforcement learning in complex disaster scene. *IEEE* **7**, 117227–117245 (2019)
18. Hidayatno, A., Destyanto, A.R., Hulu, C.A.: Industry 4.0 technology implementation impact to industrial sustainable energy in Indonesia: A model conceptualization. *Energy Procedia* **156**, 227–233 (2019)
19. Singh, V., Misra, A.K.: Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture* **4**(1), 41–49 (2017)
20. Suganthi, J., Suganthi, V., Giridharan, S.: Detection and Prevention Mechanism of Snakes and Insects Biting from Farmers using IOT Monitoring System. *Open Access Quarterly International Journal* **2**(1), 298–30 (2018)
21. Vaca-Castano, G., Driggers, R., Furxhi, O., Arvidson, C., Mazzotti, F.: Multispectral camera design and algorithms for python snake detection in the Florida Everglades. In *Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV* **10986**, 272–279 (2019)
22. Implications for herpetology and global health: Durso A.M., Moorthy G.K., Mohanty S.P., Bolon I., Salathé M., and Ruiz de Castañeda R. Supervised learning computer vision benchmark for snake species identification from photographs. *Frontiers in Artificial Intelligence* **4**, 582–110 (2021)
23. Bandala, A.A., Dadios, E.P., Vicerra, R.R.P., Lim, L.A.G.: Swarming algorithm for unmanned aerial vehicle (drone) quadrotors—swarm behavior for aggregation, foraging, formation, and tracking. *Journal of Advanced Computational Intelligence and Intelligent Informatics* **18**(5), 745–751 (2014)
24. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications* **1**(1), 7–18 (2010)
25. Jenkins, B.: Watching the watchmen: Drone privacy and the need for oversight. *Ky. LJ* **102**, 161 (2013)
26. Rahman, A., Jin, J., Wong, Y.W., Lam, K.S.: November. Development of a cloud-enhanced investigative mobile robot. In *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pp. 104–109 (2016)
27. Simelane, P.T., Kogeda, O.P., Lall, M.: A cloud computing augmenting agricultural activities in marginalized rural areas: A preliminary study. In: *2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pp. 119–124 (2015)
28. Li, C., Sun, X., Cai, J.: Intelligent mobile drone system based on real-time object detection. *Journal of Artificial Intelligence* **1**(1), 1 (2019)
29. Fao, I., W UNICEF: The state of food security and nutrition in the world. Rome, Italy: Food and Agriculture Organization of the United Nations. (2017)
30. WORLD HEALTH ORGANISATION: Snakebite Envenoming. Available at <https://www.who.int/news-room/fact-sheets/detail/snakebite-envenoming>. (2019)

9 Conclusion

In this paper, we have managed to identify common snakes that are mostly found in MRAs and what type of agriculture is mostly practiced. We have also try to discover mechanisms mostly used to prevent snakebites, killing of snakes, and reasons for killing them in MRAs. We, therefore, outline that the proposed cloud computing model that augments the use of ICT to improve agriculture as an activity in MRAs will be of great help since people living in MRAs seem to be neglected and suffering from snakebites and envenoming which leads to them to dissent farms. Farmers' dissenting farms lead to low yields or food security issues in MRAs. The proposed cloud architecture would use Drones to track and detect snakes in farms. Farmers could observe everything on their farms via phones connected with drones. This will improve food security, safe farming, and balance biodiversity.

References

1. Nugroho, A.D.: Agricultural market information in developing countries: A literature re-view. *Agricultural Economics* **67**(11), 468–477 (2021)
2. Citroni, R., Di Paolo, F., Livreri, P.: A novel energy harvester for powering small UAVs: Performance analysis, model validation and flight results. *Sensors* **19**(8), 1771 (2019)
3. Shaikh, F.B., Haider, S.: Security threats in cloud computing. *Internet Technology and Secured Transactions (ICITST)*, In: 2011 International Conference for (2011)
4. Kamei, K.: Cloud networked robotics. *IEEE Network* **26**(3), 28–34 (2012)
5. World Health Organisation: Snakebite Envenoming. (2019) Available at <https://www.who.int/news-room/fact-sheets/detail/snakebite-envenoming>, last accessed 2021/17/05
6. Naik, S., Khuntidar, B.K., Mohanta, M.P., Mondal, S.: A clinico-epidemiological study of snakebite among children in a rural medical college from eastern India. *International Journal of Pediatrics and Neonatology* **1**(1), 11–14 (2020)
7. Joshi, N.P.: Ecological and ethnobotanical values of weeds found in the spring rice fields in Chitwan, Nepal. *Ethnobotany Research and Applications* **22**, 1–19 (2021)
8. Cruz, L.S., Vargas, R., Lopes, A.A.: Snakebite envenomation and death in the developing world. *Ethnicity & disease* **19**(1), 42 (2009)
9. Wood, D., Sartorius, B., Hift, R.: Classifying snakebite in South Africa: Validating a scoring system. *South African Medical Journal* **107**(1), 46–51 (2017)
10. Wood, D., Sartorius, B., Hift, R.: Estimating the burden of snakebite on public hospitals in KwaZulu Natal. *Wilderness & Environmental Medicine* **27**(1), 53–61 (2016)
11. Sharma, R., Kamble, S.S., Gunasekaran, A.: Big GIS analytics framework for agriculture supply chains: A literature review identifying the current trends and future perspectives. *Computers and Electronics in Agriculture* **155**, 103–120 (2018)
12. Ju, C., Son, H.I.: Multiple UAV systems for agricultural applications: Control, implementation, and evaluation. *Electronics* **7**(9), 162 (2018)
13. KEHOE B., KAHN G., MAHLER J. Autonomous multilateral debridement with the raven surgical robot. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1432–1439. IEEE (2014)
14. Castiblanco, C., Rodriguez, J., Mondragon, I., Parra, C., Colorado, J.: Air drones for explosive landmines detection. In: *ROBOT2013: First Iberian Robotics Conference*, Springer. Colombia:Bogota, pp. 107–114. (2014)

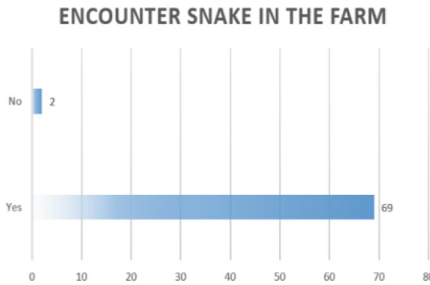


Fig. 7. Farmers who encountered snakes in the farm

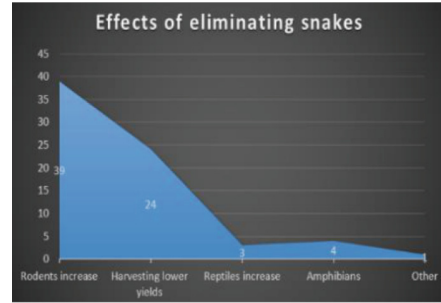


Fig. 8. Effects of eliminating snakes

In Fig. 7 97% of the farmers agree that they encounter snakes on their farms. Farmers eliminate or remove 73% of these snakes, and 67% of the snakes fight back, which may result in hospitalization or fatality. Farmers being killed by snakes lead to a reduction in productivity (since some productive and knowledgeable farmers die from snake bites), and it also leads to the loss of livestock. With more frequent snake bites in MRAs, farmers surrender farms to snakes leading to fewer farms to grace in. Some-times farmers may use fire to clear such bushy areas leading to the loss of animals, trees, ecosystems, etc. Snakes also get killed on farms in rural areas, which creates an imbalance in biodiversity. While these snakes are eliminated, 54% of the rodents impact their agricultural products, as presented in Fig. 8.

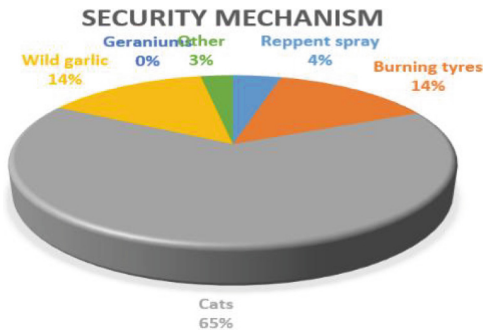


Fig. 9. Security mechanisms to curb snakes

As shown in Fig. 9 farmers use different strategies to curb snakes in their farms, however the above-presented security mechanism is not able to handle the death rate of both farm workers and snakes on farms. The most common strategy or method used by farmers in farms to get rid of snakes is cats, which is at 65%.

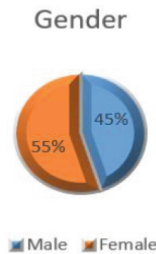


Fig. 2. Gender

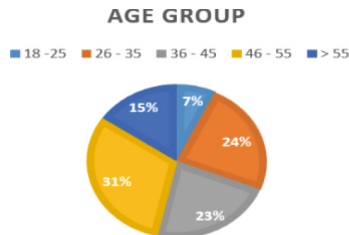


Fig. 3. Age group

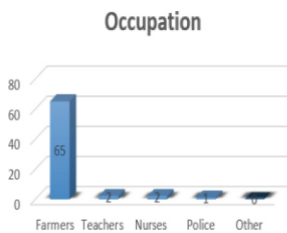


Fig. 4. Occupation

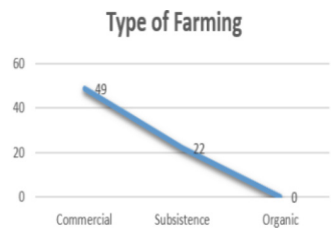


Fig. 5. Type of farming

Data on gender, age bracket, occupation (see Fig. 4), and type of farming (see Fig. 5) were collected. This was the way of finding out which gender dominates MRAs and their age bracket, whether they are employed or not, and also how many people they are supporting. Data on how farming is practiced in MRAs were collected, as the tools used and the type of farming. A total number of 71 farmers were given questionnaires and participated. In our data, as shown in Fig. 2 we discovered that out of 71 participants, 55% were male and 45% female. Figure 3 also shows that it was mostly farmers aged 46-55 at 31% and followed by youth at 24% aged 26-35, which proves the scarcity of jobs in MRAs.

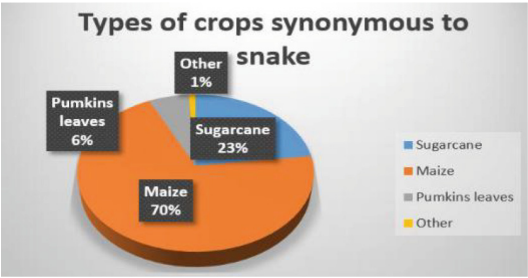


Fig. 6. Types of crops synonymous to snakes

In farms, we also discovered that 86% of crops are synonymous with a snake, as presented in Fig. 6. Most farmers have planted maize with 70% and 23% of its sugarcane, as reflected in Fig. 6.

In Fig. 1, the bottom of the architecture shows different farms with different kinds of snakes. We have farms from different locations in the Zululand district. A swarm of drones will fly around the phone, controlled by the farmer's phone. Farmers have two options either they download or upload information using the different types of phones they are using. Since the drones are going to be tracking and detecting snakes in farms. Using the mobile phone, tracked and detected information is uploaded from the phone to the cloud wirelessly. And we have a firewall to prevent all trespassers from tampering with the system. Tracked and detected information is then transmitted to the central access point connected to the server providing mobile network services. The server is a database that stores the user's information about that particular object that is detected. In the cloud, everything will be processed then cloud controllers process the request to provide mobile users (farmers) with the corresponding cloud services. When the farmer opens his/her Swarm of Cloud-based Unmanned Aerial Vehicle system on the phone, he/she will be able to see the real-time navigation of drones as they track and detect snakes in the farm.

7 Methodology

The most comprehensive investigation used questionnaires to learn about how farming is practiced in Zululand, which mobile devices are most commonly used in MRAs, current safety measures used against snakes, the most common snake affecting MRAs, methods they use to live, detect and handle snake bites. We created the paper-based questionnaire, and since we were conducting the study in MRAs, we went there physically to clarify more questions. The questions were both structured and unstructured questions; it was close-ended questionnaires. We interviewed both farmers and farm workers. The case study was conducted in the Zululand district. The sample was non-random. We were questioning the people on a willing basis.

8 Data Analysis and Results

The main objective of this paper was to identify what are common snakes that are mostly found in MRAs and what type of agriculture is mostly practiced. Mechanisms are mostly used to prevent snakebites, killing of snakes, and reasons for killing them in MRAs. We collected data through close-ended questionnaires given to farmers in the Pongola Zululand district in the KwaZulu-Natal province of South Africa. Farmers lack skills to improve their security and safety due to a lack of technological tools to help them improve their farming to conclude our study properly. It was necessary to analyse the data so that we could correctly test our suggestions as well as answer our research questions and present the results of the study to our readers in an understandable and convincing form. A total number of 71 farmers were given questionnaires and participated. Data on gender, age bracket, occupation, number of people in the household were collected, crops and types that are synonymous with snakes, frequent encounters with the snake and What they do with the snake or does it defend itself, and reasons for killing snakes, common things that are bitten by a snake(s) in farms and security mechanisms currently in place to curb snakes. The data that answers the above-men-tioned points are also presented graphically below.

to have far-reaching effects across today’s society, trans-forming our lives and how we do business. The agricultural industry seems to have embraced drone technology with open arms, using these advanced tools to transform modern farming. The advent of drones, better known as drones, has proved to be beneficial for the overall development of the human race – both technologically and strategically [28].

Drones-based agricultural robotics, in particular, has attracted immense interest as is evident in 10-year sales forecasts of the technology [25]. In Agriculture, from crop monitoring to planting, livestock management, crop spraying, irriga-tion mapping, and more. Agricultural drones help to achieve and improve what’s known as precision agri-culture. This approach to farming management is based on observing, measuring, and acting based on real-time crop and livestock data. It erases the need for guesswork in modern farming and allows farmers to maximize their yields and run more efficient organizations, all while enhancing crop production.

There are multiple uses for agricultural drones, including Scouting land and crops, checking for weeds and spot-treating plants, monitoring overall crop health and manag-ing livestock, and monitoring for health issues. Drones have propulsion systems, infrared cameras, global positioning and navigation sys-tems, programmable controllers, and automated flight planning. Plus, with cus-tom-made data processing software, any col-lected information can instantly be used towards better management decisions. This will make it easier for research-ers to track and detect snakes on farms.

In recent years, the cost of agriculture drones has rapidly declined, which has not only led to the explosion of drone use cases in agriculture but has made it a no-brainer investment for modern farmers [11].

Multi-rotor Drones are the cheapest, easiest to make and to operate of all the drone types mentioned above (based on aerial platforms). Single-pilot livestock management and observation are made possible by including cameras on multi-rotor drones.

6 Proposed Architecture

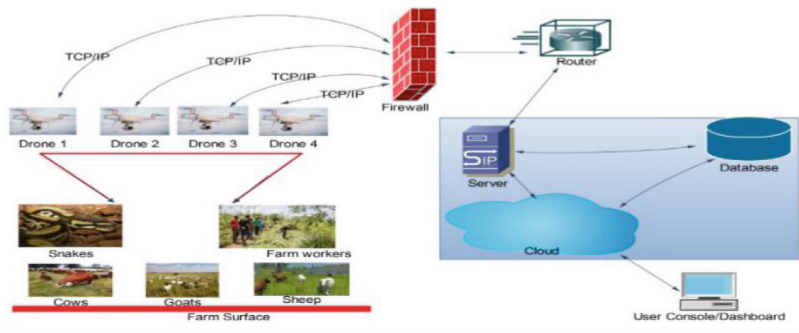


Fig. 1. Proposed system architecture

to accurately track snakes while Faster R-CNN would be on image detection since it's difficult to spot green snakes in long green grass and sugarcane.

3 Purpose, Objective, and Significance of the paper

The purpose of this study was to ascertain the penetration and use of drones as means of detecting and tracking snakes in the farming community of rural Zu-luland region, KwaZulu-Natal. The objective was to develop a model of cloud-based swarm Robotics for clustered drones to accurately detect and track stationary and motion-based snakes in an agricultural environment or farm. The model platform could assist the local farming community from snakebites and envenoming. In addition, the model will help curb snake killings and enhance biodiversity in rural areas. The expected contribution of this research is developing a cloud-based swarm DRONE model that facilitates the intelligent detection and tracking of stationary and motion-based snakes in farms using PSO and a Faster R-CNN algorithm. This will improve safe farming and biodiversity conservation and also improve food security in developing countries.

4 Cloud Robotics

Cloud computing is a computing style that provides power referenced with Information Technology (IT) as a service [24]. Cloud computing is the type of computing that allows access to information and computer resources anywhere as long as the network connection is available. Most organizations and individuals have migrated their tasks, data, and applications to the cloud. Such include but are not limited to Amazon, where we find Dropbox, Twitter, Instagram, Quora, etc. All these are applications used by millions of people that use the Cloud.

Cloud Robotics is a topic that has garnered much attention from the research community, especially with the proliferation of cloud infrastructure, improved communications technologies, and commoditized Robotics. Cloud Robotics is any automation system that relies on either data or code from a network to support its operation [13]. Cloud Robotics can also be described as any automation system that relies on either data or code from a network to support its operation. Drones-based agricultural Robotics, in particular, have attracted immense interest, as is evident in 10-year sales forecasts of the technology [25]. The benefits of Cloud Robotics include shared knowledge databases that disparate robots have access to, the ability to offload processing-intensive tasks to cloud infrastructure, and robot access to skill/behaviour databases [26].

In this paper, we incorporate cloud and drones, and agricultural workers will be able to gather and store data, automate redundant processes, and improve efficiency. Since Drones will be tracking and detecting snakes on farms, the data recorded or snake's data tracked will be stored in the cloud for future use.

5 Drones in Agriculture

ICT in agriculture has attracted a lot of attention and researchers for the past few years all over the globe, as it seems to have outstanding benefits compared to old ways of practicing farming [27]. Drones technology is a phenomenal innovation that continues

and processing system to detect Burmese pythons on the wild. They have leveraged IMEC sensor array technology to collect data on the Vis-NIR regions of the spectrum of several pythons to determine a set of bands that can help with the detection. They further allude that Hyperspectral measurements and band selection algorithms show that an optimal solution involves the combination of bands in both visual and NIR. Due to technological fabrication issues is more difficult and expensive to combine very broad-spectrum bands in a single mul-tispectral focal plane array. For that reason, they compare the Vis multispectral sensor vs. the NIR multispectral sensor showing that NIR multispectral sensor has a lead on the task. Consequently, their analysis was concentrated on the NIR multispectral collection to find an algorithm that performs best in the task of python discrimination. They first trained a random forest algorithm since it con-siders only spectral information. Later, they studied the use of deep learning to improve the detection of pythons in the wild. They discovered that deep learning algorithms need lots of labeled data to converge. Their collection included less than 100 images. Therefore, they decided to use a pre-trained model on RGB data and take advantage of transfer learning by fine-tuning the network using their data. Accordingly, they selected three bands that are used in substitution of the original RGB bands in the deep network. They conclude that future research-ers need to cover aspects of studying different deep network architectures, in-clude more spectral bands on the algorithm; perhaps a new data collection that supports these networks is required, and finally, actual hardware implementation of the camera.

Some researchers use a combination of algorithms to achieve the task of classifying snakes [22]. In their study, they investigate human perception and the se-lection of words in describing a snake based on their visual view. The descrip-tions are presented in unstructured text, and the NLP processing involves pre-processing, feature extraction, and classification. Four machine learning algo-rithms (naïve Bayes, k-Nearest Neighbour, Support Vector Machine, and Deci-sion Trees J48) were used during training and classification.

Generally, all existing works related to swarm intelligence were derived from the group or social behaviour of animals or insects [23]. In their study, they exhibit the compatibility of applying swarm algorithm on DRONE quadrotors for aerial surveillance, search, and reconnaissance operations through flight for-mations and reconfiguration by abiding swarming patterns and behaviour.

They conclude that the increased number of robots can yield higher accuracy. This directly implies that the centroid can be controlled accurately by increasing the number of robots because the resolution of the swarm increases proportion-ally with the number of swarm members. The foraging behaviour experiment revealed that the time it takes to hit or reach the desired position decreases with the increase in swarm members. They suggest that for future purposes, aggregation can be implemented, including other robot types i.e., underwater or land-based robots. Optimized searching algorithms in three-dimensional domains can be derived from the foraging behaviour. Lastly, they suggest that all of this be-haviour can be mixed with other algorithms, such as pheromone and CNN algorithms, to introduce multi-tasking for the swarm. Adopting their suggestion, our study seeks to combine Swarm and Faster R-CNN algorithm to track and detect snakes in agricultural fields in MRAs. Swarm algorithm will guide could-based Drones

faster automation, digitization, and big data collection that manufacturers and industries must align with. Efficiency improvement, reducing costs, and maintaining quality can be more comfortable than before with the help of these components. So this is what motivated us to work on such an experiment where the effective procedure of automated SCDRONE tracking and detection of snakes in farms can somehow help protect farmers and farm workers from dangerous snakes and also protect the very same snakes from being killed by farmers for biodiversity purposes. This will help farmers to produce better and support the integration of industry 4.0.

Many algorithms have been used to try to track and detect plants and animals on farms. The algorithms take images, make segmentation to extract features from them, and use the features, and the machine classifies which disease the plant has [19]. In their study, they develop the design of how machine learning can be used in automatically detecting plant diseases by seeing the plant leaves. Their objective was to construct a system that takes images as input, and after precise testing, it gives the disease name in the output. To implement their proposed method, they collected data manually and used a faster R-CNN algorithm and some necessary tools. Their implementation process consists of several segments and pre-processing, which is described below: Dataset Collection, TensorFlow in disease detection, Labelling of the images, and Algorithm used.

Their expected result is obtained with some computational efforts where the efficiency of the proposed algorithm can be shown, and the classification of leaf disease can be specified. They managed to achieve an accuracy value, and that is 67.34%. Their study is similar to our study because we track and detect motion-based objects (snakes) while they are detecting stationary plant diseases. We both opt to use the Faster R-CNN algorithm for image detection because it's much faster. Another reason we opt for faster R-CNN is its ability to detect objects in real-time compared to R-CNN and the Fast R-CNN algorithm. Our model allows a swarm of cloud-based Drones to track and detect snakes in real-time in MRAs, which is why we also adopt PSO for guiding the Drones' path. We will improve from the 67.34% accuracy they achieved even though we use different objects.

Many algorithms have been used to track and detect snakes, to be precise. The surveillance and tracking of the snakes are difficult due to their size and the nature of their movement [20]. In their study, they proposed a system that seeks to identify snakes and small dangerous insects to the farmers and improve productivity using a classification algorithm. Their classification algorithm and feature extraction describe the unique features of snakes and small dangerous insects. Then they produce the buzzer in the current location and display the location use of GPS. In their method, the detection of motion in the video frame and identification of the objects in the area of motion using features extraction, which describes the unique features of snakes and small dangerous insects. They use the platform Raspberry PI, which they say is the sequence of credit card-sized single-board computers established in the United Kingdom by the Raspberry PI Foundation. The position of the snakes, once detected, is tracked in order to calculate the distance of snakes with areas of farmers in the agricultural land.

Given the magnitude of the snake problems, better detection tools are needed to help to find the snakes [21]. In their study, they study the development of a camera

integrate and evaluate a set of low-cost technologies that allow the detection of explosive landmines autonomously and without compromising the mission. DRONE was equipped with cameras that enable visual feedback of the terrain in real-time. By capturing several sequences of images, visual algorithms for landmine detection are applied. The detection process was performed using the still images captured by the bottom camera of the drone. Outdoor tests were conducted using tuna cans as landmine-like objects and under different sunlight profiles and wind speeds. Some objects were randomly placed on the surface (fully visible), while others were partially buried.

The difference between their study and the current study is that we are not only tracking and detecting still objects, but we are also tracking moving objects in the form of snakes. Our Drones will be capable of tracking and detecting stationary and motion-based snakes. Our model is also incorporated with the cloud to store this information and classify the kind of snakes detected, which will help as awareness people of MRAs to be aware of such snakes and relocate those snakes to areas of safety before they are killed.

Artificial Intelligence (AI) is a vital technology for the future of drone systems to improve their independent performance (Yadav and Gaur, 2014). Drones should be able to perform cloud-based tasks autonomously and have abilities to perform self-determination of tasks.

Future drones should be able to autonomously plan flight paths based on their respective missions and corresponding constraints [15]. In our study, Drones fly around the farm, tracking and detecting snakes. When the constraints change, Drones should be able to autonomously adjust the flight path.

One of the characteristics of intelligent Drones in the future is the ability to efficiently perform complex tasks through independent cooperation [16]. AI has played an increasingly important role in the field of automated control of drones [17]. In their study, they prove that deep reinforcement learning can be successfully applied to an ancient puzzle game Nokia Snake after further processing. A game with four directions of movement. Through deep intensive learning and training, the Snake (or self-learning Snake) learns to find the target path autonomously, and the average score on the Snake Game exceeds the average score on the human level.

Therefore, their proposed Snake algorithm to be able to find the target path autonomously is an attempt and key technology designing of autonomous search and rescue personnel and material dispatching drones. They apply the reinforcement learning method to the process of simulating the autonomous exploration of the target by the drone in the game environment of the Snake Game. The snake body is used to represent the drone, and the hotspot is used to represent the target to be searched.

In their study, a single drone was used for testing, and results were achieved. In our study, through a combination of PSO and Faster R-CNN algorithm, we used a swarm of drones to track and detect snakes on a farm. The swarm of cloud-based Drones will be utilized to track and detect snakes.

In the recent era, the swift development in digital technology has started a new evaluation in the industrial revolution called Industry 4.0 generally [18]. This revolution is all about introducing modern technologies to connect the components with the industries and support sustainability as well. It brings new and augmented algorithms to promote

The study is similar to ours because we seek to utilize a swarm of Drones in agriculture to detect and track motion-based snakes on farms. The difference is that we don't only discuss Drones. Our study develops a cloud-based model to help farmers detect and track motion-based snakes on farms. The Swarm of Cloud-based Unmanned Aerial Vehicles (SCDRONE) tracks and detects motion-based snakes that can pose a danger to farm workers.

According to [12], 80% of the commercial market for Drones is expected to be occupied by agricultural Drones in the future. They further outlined that by introducing Drones to traditional agriculture, working hours and labour requirements have been significantly reduced, and the efficiency of agricultural works has improved significantly. However, they've seen that using a single drone it's a drawback because it uses the battery as its main source of power. In their study, they propose a multi-drone system that will make it possible to carry out cooperative works simultaneously to curb the inefficiency of a single drone in terms of time and energy. In their study, they develop a multi-drone system for agriculture using the distributed swarm control algorithm and evaluate the system's performance, which is also similar to our study because we target the swarm of Drones.

The difference is that we don't only check the functionality of the swarm of Drones in terms of performance and efficiency to execute tasks. Our study develops a cloud-based model to help farmers detect and track motion-based snakes in farms using the Swarm algorithm and the Faster R-CNN algorithm. Our study is not only beneficial to farmers. It even helps to protect and save snakes that farmers and farm workers kill. The Swarm of Cloud-based Unmanned Aerial Vehicle (SCDRONE) tracks and detects any motion-based snakes that can pose a danger to farm workers. Performance and efficiency in executing tasks will be observed and improved, cloud and drone interaction will be checked, and path planning will also be monitored.

Cloud computing can be defined as utilizing the internet to provide technology-enabled services to people and organizations [3]. Cloud Robotics, to be specific, is a topic that has gathered much attention from the research community globally, especially with the increase in cloud infrastructure, improved communications technologies, and commoditized Robotics. Robotic services are systems, devices, and robots with three functions: Sensation, actuation, and control [4].

There have been numerous studies in the area of Drones used in agriculture, wildlife tracking and conservation, Cloud Robotics, drone-based image processing, and drone-based path planning.

Cloud Robot and Automation systems a system that relies on either data or code from a network to support their operation, i.e., where not all sensing, computation, and memory are integrated into a single standalone system [13]. In their definition, the researchers are trying to include future systems and many existing systems that involve network teleportation or networked groups of mobile robots, such as Drones or warehouse robots, as well as advanced assembly lines, processing plants, and home automation systems.

Drones have been used to detect explosives landmines [14]. The researchers outlined that the military has been the first to deploy machines to overcome the risks involved when a human carries out the landmine detection process. The goal of their study was to

agriculture. In Sect. 6, we explore snakes. In Sect. 7, we present related work. In Sect. 8, we present the proposed architecture. In Sect. 9, we discuss the methodology followed by findings and data analysis. Lastly, in Sect. 10, we present the conclusion.

2 Related Work

Snakebite is a neglected tropical disease and one of the major causes of mortality in developing countries [6]. They further state that deaths due to snakebites are 2.8% of total deaths. Most cases are during monsoon 55% and in rural areas 93%. Snake bites are well-known veterinary emergencies in many parts of the world, especially in rural areas [7]. They further allude that, Snake-bite is an environmental and climatic hazard. It results in the death or chronic disability of many animals and people, especially those involved in farming.

Papua New Guinea has one of the world's highest incidence rates of snake bites [8]. Papua New Guinea records 561.9 cases per 100 000 population. They further allude that in Africa, the annual incidence rate of snake bites in the Benue Valley of northeastern Nigeria is 497 per 100,000 population, with a case-fatal-ity ratio of 12.2%. In northern Africa, the species that causes most bites and deaths belong to the family Viperidae, *Echis* sp (saw-scaled vipers).

In the case of our study, which is MRAs of Southern Africa, there are some 38 venomous snake species in South Africa (SA), of which approximately half are dangerous to humans [9]. They further allude that the highest incidence of snakebite in South Africa is in the rural northeastern coastal belt of KwaZulu-Natal. Small local studies have suggested an annual incidence of snakebite in northeastern parts of the province of 28–96.5 per 100 000. Prevalent species that cause problems in KwaZulu-Natal include Mozambique spitting cobra (*Naja mossambica*) and puff adder (*Bitis arietans*), an elapid and viperid respectively, both of which have potent cytotoxic venom. The black mamba (*Dendroaspis polylepis*) and various cobra (*Naja*) species are elapids possessing potent neurotoxic venom and muscle weakness. The boomslang (*Dispholidus typus*), a colubrid with a haemorrhagic venom, can cause potentially fatal bleeding.

The case study [10] presented that the subtropical low-lying northeast of KZN accounted for the majority of snakebites, in keeping with other studies showing hot, humid climates in low-lying rural areas to be hotspots for snakebite for snakes (puff adder, boomslang, and Mozambique spitting cobra). The 3 districts representing this region (uMkhanyakude, Zululand, and uThungulu) are all underdeveloped and have primarily rural subsistence populations of KZN.

Drones allow farmers to observe their fields from the sky, which can reveal many issues on the farm, common among which is irrigation-related problems, soil variations, and fungal and pest infestations [11]. In their study, researchers discuss the different types of Drones, and their application in pest control, crop irrigation, health monitoring, animal mustering, geo-fencing, and other agricultural-related activities. The author further shares the advantages and potential benefits of Drones in agriculture. The study does manage to present four major types of Drones. Though the multi-rotor drones, with their ability to hover on the spot and take off and land vertically, seem suited for agriculture, their limited flight time is a major concern.

Cloud computing can be defined as utilizing the internet to provide technology-enabled services to people and organizations [3]. Cloud computing has emerged over the past years and has become the most common and helpful source of information between partners and people who are needy and who want to share certain information. If agriculture were practiced with the latest technologies in MRAs, we wouldn't face challenges such as famine, poverty, unsafe farming and wildlife killings, crime, and rural-to-urban migration.

Robotic services are systems, devices, and robots with three functions: Sensation, actuation, and control [4]. Providing cloud-based robotic services in agriculture will curb many challenges faced by farmers and farm workers in developing countries. About 5.4 million snake bites occur yearly, resulting in 1.8 to 2.7 million cases of envenoming [5]. They further say there are between 81 410 and 137 880 deaths and around three times as many amputations and other permanent disabilities each year. In their findings, it is also discovered that most of these occur in Africa, Asia, and Latin America. In Africa, WHO researchers found that there are an estimated 435 000 to 580 000 snake bites annually that need treatment. 70% of this envenoming affects farmers (both young and old) in poor rural communities in low- and middle-income countries.

Snakebite is a neglected tropical disease and one of the major causes of mortality in developing countries [6]. They further state that deaths due to snakebites are 2.8% of total deaths, and most cases are during monsoon 55%, and from rural areas, 93%.

MRAs farmers and farm workers are not safe due to dangerous snakes that roam around farms. Maintaining a high level of biodiversity is important to all life on earth, including humans, and snakes are an important part of that biodiversity. Snakes make up a significant proportion of the middle-order predators that keep our natural ecosystems working. Killing them also creates an imbalance in the ecosystem. A cloud-based Robotics model that harnesses a cluster of drones to detect and track stationary and motion-based snakes that pose a danger to farmers and farm workers in the context of day navigation for better safety of both farmers and snakes can be a solution.

In this paper, we therefore present findings of a preliminary study of how farmers from MRAs conduct farming amid snakes, the type of technologies they are using to detect snakes, what common snakes are mostly found in MRAs, and what type of agriculture is mostly practiced, mechanism mostly used to prevent snakebites, killing of snakes and reasons killing them in MRAs, among other demographic information. We collected data through close-ended questionnaires given to farmers in the Zululand district in the KwaZulu-Natal province of South Africa. Due to the high likelihood of illiteracy among MRAs, we were also physically present to explain the questions in the questionnaire. We compiled and analyzed the data after it was all collected.

The plan and implementation of the proposed system will be based on the findings of this preliminary study, which provides a solution to the problem of detecting and tracking stationary and motion-based snakes in farms in MRAs to curb the envenoming and deaths of farmers and farm workers. The system will improve safe farming, maintain a high level of biodiversity and enhance yields in MRAs.

The rest of this paper is organized as follows: In Sect. 2, we present the purpose, objective, and significance of the paper. In Sect. 3, Cloud robotics and their importance are explored. In Sect. 4, explore more about Drones. In Sect. 5, we discuss Drones in



A Cloud-Based Drones' Model for Detection and Tracking of Stationary and Motion-Based Snakes in Farms in Marginalized Rural Areas: A Preliminary Study

Phumlani T. Simelane^(✉)  and Okuthe P. Kogeda 

University of the Free State, Bloemfontein 039, South Africa
{2019872668, kogedapo}@ufs.ac.za

Abstract. Marginalized Rural Areas (MRAs) practice farming as their main source of food, employment, and income, yet in most cases, they lack the basic resources and skills to improve their yields and farming techniques. Due to the lack of information and resources, farmers experience snake bites. Others even get killed without help due to the long distance they travel to obtain help from healthcare facilities. Farmers being killed by snakes lead to a reduction in productivity (since some productive & knowledgeable farmers die), and it also leads to the loss of livestock. With more frequent snake bites in MRAs, farmers surrender farms to snakes leading to fewer farms to grace in. Hence there is a need to design and develop a swarm cloud-based drone model for tracking and detection of stationary motion-based snakes in farms for better safety of both farmers and snakes. We collected data through close-ended questionnaires given to farmers from the Zululand district in South Africa. The preliminary study results show that 97% of farmers encounter snakes on their farms, and 70% of the crops are synonymous with snakes. Farmers eliminate 73% of these snakes, and 67% of the snakes fight back, which may result in hospitalization or fatality. While these snakes are eliminated, 54% of the rodents impact their agricultural products. The most common strategy or method used by farmers in farms to get rid of snakes is cats, which is at 65%.

Keywords: Drones · Tracking · Detection · CNN · PSO · Snakes · MRAs · Agriculture and food security

1 Introduction

Information and Communication Technology (ICT) in agriculture has attracted a lot of attention and research over the past few years all over the globe, for it seems to have outstanding benefits compared to traditional farming. ICT has played a major role in collecting and sharing timely and accurate information on markets, weather, inputs, and prices in developing countries [1]. The use of drones in agriculture and smart farming is very effective because drones can give farmers a bird's eye view of their fields while remaining close to the terrain and so providing more precise evaluations [2].

77. Jones, A., Vidalis, S.: Rethinking digital forensics. *Annals of Emerging Technologies in Computing* **3**(2), 41–53 (2019). <https://doi.org/10.33166/AETiC.2019.02.005>
78. Adedayo, O.M.: Big data and digital forensics, Rethinking Digital Forensics, In: 2016 IEEE International Conference on Cybercrime and Computer Forensic (ICCCF). (2016)
79. Omer, M.A., Yazdeen, A.A., Malallah, H.S., Abdulrahman, L.M.: A Survey on Cloud Security: Concepts, Types, Limitations, and Challenges. *Journal of Applied Science and Technology Trends* **3**(2), 101–111 (2022). <https://doi.org/10.38094/jastt301137>
80. Apau, R., Koranteng, F.N.: An overview of the digital forensic investigation infrastructure of Ghana. *Forensic Science International: Synergy* **2**, 299–309 (2020). <https://doi.org/10.1016/j.fsisyn.2020.10.002>

56. Soltani, S., Seno, S.A.H.: A survey on digital evidence collection and analysis, In: 7th International Conference on Computer and Knowledge Engineering, Iran: IEEE, pp. 1–7. (2017)
57. Lee, S., Kim, H., Lee, S., Lim, J.: Digital evidence collection process in integrity and memory information gathering, (2005)
58. Karagiannis, C., Vergidis, K.: Digital evidence and cloud forensics: Contemporary legal challenges and the power of disposal. *Information (Switzerland)* **12**(5), 181 (2021). <https://doi.org/10.3390/info12050181>
59. Nikkel, B.J.: Improving evidence acquisition from live network sources. *Digit. Investig.* **3**(2), 89–96 (2006). <https://doi.org/10.1016/j.diin.2006.05.002>
60. Ferguson, R.I., Renaud, K., Wilford, S., Irons, A.: PRECEPT: a framework for ethical digital forensics investigations. *J. Intellect. Cap.* **21**(2), 257–290 (2020). <https://doi.org/10.1108/JIC-05-2019-0097>
61. Allie, R.: Judgement of Hanna Cornelius: Case No: CC 04/2018. Cape Town: Western Cape High Court, pp. 1–69. (2018)
62. Masipa, J.: Oscar Pistorius Murder Charge: Case No: CC113/2013, pp. 1–26. (2016)
63. Wilson, S.D.J.: Tshegofatso Pule Murder: Case No: SS36/2021, pp. 1–10. (2022)
64. Desai, S.: Henri Christo Van Breda Murder Case: No SS17/16, pp. 1–270. (2018)
65. Makgoba, M.W.: The Report into the Circumstances Surrounding the Deaths of Mentally Ill Patients: Gauteng Province. (2017)
66. Khan, A., Wiil, U.K., Memon, N.: “Digital forensics and crime investigation: Legal issues in prosecution at national level”, in *5th International Workshop on Systematic Approaches to Digital Forensic Engineering*. SADFE **2010**, 133–140 (2010). <https://doi.org/10.1109/SADFE.2010.8>
67. van Beek, H.M.A., van den Bos, J., Boztas, A., van Eijk, E.J., Schrap, R., Ugen, M.: Digital forensics as a service: Stepping up the game. *Forensic Science International: Digital Investigation* **35**, 301021 (2020). <https://doi.org/10.1016/j.fsidi.2020.301021>
68. Jarrett, A., Choo, K.R.: The impact of automation and artificial intelligence on digital forensics. *WIREs Forensic Science* **3**(6), 1–5 (2021). <https://doi.org/10.1002/wfs2.1418>
69. Marshall, K., Rea, A.: Legal challenges in cloud forensics. In: 27th Annual Americas Conference on Information Systems, AMCIS 2021 (2021)
70. Akinbi, A.O.: Digital forensics challenges and readiness for 6G Internet of Things (IoT) networks. *WIREs Forensic Science* (2023). <https://doi.org/10.1002/wfs2.1496>
71. Choi, M., EL Azzaoui, A., Kumar Singh, S., Mohammed Salim, M., Reward Jeremiah, S., Hyuk Park, J.: The Future of Metaverse: Security Issues, Requirements, and Solutions. *Human-centric Computing and Information Sciences*, vol. 12 (2022)
72. Sun, Y., Tian, Z., Li, M., Zhu, C., Guizani, N.: Automated Attack and Defense Framework toward 5G Security. *IEEE Netw* **34**(5), 247–253 (2020). <https://doi.org/10.1109/MNET.011.1900635>
73. Pooyandeh, M., Han, K.J., Sohn, I.: Cybersecurity in the AI-Based Metaverse: A Survey. *Applied Sciences (Switzerland)* **12**(24), 129993 (2022). <https://doi.org/10.3390/app122412993>
74. Batista, D., et al.: Exploring Blockchain Technology for Chain of Custody Control in Physical Evidence: A Systematic Literature Review. *Journal of Risk and Financial Management* **16**(8), 360 (2023). <https://doi.org/10.3390/jrfm16080360>
75. Daryabar, F., Dehghantanha, A., Choo, K.K.R.: Cloud storage forensics: MEGA as a case study. *Australian Journal of Forensic Sciences* **49**(3), 344–357 (2017). <https://doi.org/10.1080/00450618.2016.1153714>
76. Martini, B., Choo, K.K.R.: Cloud storage forensics: OwnCloud as a case study. *Digit Investig* **10**(4), 287–299 (2013). <https://doi.org/10.1016/j.diin.2013.08.005>

38. Mabuto, E.K., Venter, H.S.: State of the art of Digital Forensic Techniques. *Information Security for South Africa (ISSA)* **2011**, 1–7 (2011)
39. Ahmed Ali, S., Memon, S., Sahito, F.: Challenges and solutions in cloud forensics, In: *ACM International Conference Proceeding Series, Association for Computing Machinery*, pp. 6–10. (2018). <https://doi.org/10.1145/3264560.3264565>
40. Mousa, A.N., Ithnin, N., Almolhis, N., Zainal, A.: A Consumer-Oriented Cloud Forensic Process Model, In: *2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC 2019)*, pp. 1–6. (2019)
41. Akter, O., Akther, A., Uddin, M.A., Manowarul Islam, M.: Cloud Forensics: Challenges and Blockchain Based Solutions. *International Journal of Wireless and Microwave Technologies* **10**(5), 1–12 (2020). <https://doi.org/10.5815/ijwmt.2020.05.01>
42. Montasari, R., Hill, R.: Next-Generation Digital Forensics: Challenges and Future Paradigms, In: *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, pp. 1–8. (2019)
43. Baig, Z.A., et al.: Future challenges for smart cities: Cyber-security and digital forensics. *Digital Investigation* **22**, 3–13 (2017). <https://doi.org/10.1016/j.diin.2017.06.015>
44. Montasari, R.: An overview of cloud forensics strategy: Capabilities, challenges, and opportunities. *Strategic Engineering for Cloud Computing and Big Data Analytics* (2017). https://doi.org/10.1007/978-3-319-52491-7_11
45. Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., Markakis, E.K.: A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Communications Surveys and Tutorials* **22**(2), 1191–1221 (2020). <https://doi.org/10.1109/COMST.2019.2962586>
46. Herman, M., et al.: NIST cloud computing forensic science challenges, Gaithersburg, Maryland (2020). <https://doi.org/10.6028/NIST.IR.8006>
47. Isaac Abiodun, O., Alawida, M., Esther Omolara, A., Alabdulatif, A.: Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey. *Journal of King Saud University - Computer and Information Sciences* **34**(10), 10217–10245 (2022). <https://doi.org/10.1016/j.jksuci.2022.10.018>
48. Jain, P., Mahalkari, A.: Review of Cloud Forensics: Challenges, Solutions and Comparative Analysis. *Int J Comput Appl* **178**(34), 28–34 (2019). <https://doi.org/10.5120/ijca2019919220>
49. Sharma, P., Arora, D., Sakthivel, T.: UML-based process model for mobile cloud forensic application framework - a preliminary study. *International Journal of Electronic Security and Digital Forensics* **12**(3), 262 (2020). <https://doi.org/10.1504/IJESDF.2020.108296>
50. Kent, K., Chevalier, S., Grance, T., Dang, H.: *Guide to Integrating Forensic Techniques into Incident Response*. Gaithersburg, Maryland (2006)
51. Ninawe, P., Ardhapurkar, S.: Design and Implementation of Cloud Based Mobile Forensic Tool. In: *ICIIECS'15 : DRDO sponsored 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems : 19th and 20th March 2015 : proceedings* (2015)
52. Kazim, A., Almaeeni, F., Al Ali, S.: “Memory Forensics: Recovering Chat Messages and Encryption Master Key,” In: *2019 10th International Conference on Information and Communication Systems (ICICS) : 11–13 June, 2019, Jordan University of Science and Technology, Irbid, Jordan*, pp. 1–7. (2019)
53. Guttman, B., White, D.R., Walraven, T.: *Digital Evidence Preservation*, (2022). <https://doi.org/10.6028/NIST.IR.8387>
54. Chow, K.P., et al.: *Digital Evidence Search Kit*. IEEE, Taipei Taiwan (2005)
55. Silvarajoo, V.R., Yun Lim, S., Daud, P.: Digital Evidence Case Management Tool for Collaborative Digital Forensics Investigation, In: *2021 3rd International Cyber Resilience Conference, CRC 2021, Institute of Electrical and Electronics Engineers Inc.* (2021). <https://doi.org/10.1109/CRC50527.2021.9392497>

21. Jansen, W., Ayers, R.: Guidelines on Cell Phone Forensics Recommendations of the National Institute of Standards and Technology. Nist Special Publication, **800**(101) (2007)
22. Siddaway, A.P., Wood, A.M., Hedges, L.V.: How to Do a Systematic Review: A Best Practice Guide for Conducting and Reporting Narrative Reviews, Meta-Analyses, and Meta-Syntheses. *Annu. Rev. Psychol.* **70**, 747–770 (2019). <https://doi.org/10.1146/annurev-psych-010418>
23. Khan, K.S., Kunz, R., Kleijnen, J., Antes, G.: Five steps to conducting a systematic review (2003) [Online]. Available: <http://www.ncbi.nlm.nih.gov/entrez/query/>
24. Okoli, C.: A Guide to Conducting a Standalone Systematic Literature Review. *Communications of the Association for Information Systems*, **37**, 1–33 (2015) [Online]. Available: <http://aisel.aisnet.org/cais/vol37/iss1/43>
25. Oosterwyk, G., Brown, I., Geeling, S.: A Synthesis of Literature Review Guidelines from Information Systems Journals. *Kalpa Publications in Computing* **12**, 250–260 (2019)
26. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering - A systematic literature review. *Information and Software Technology* **51**(1), 7–15 (2009). <https://doi.org/10.1016/j.infsof.2008.09.009>
27. Schryen, G.: Writing qualitative is literature reviews—Guidelines for synthesis, interpretation, and guidance of research. *Commun. Assoc. Inf. Syst.* **37**, 286–325 (2015). <https://doi.org/10.17705/1cais.03712>
28. Yadav, D., Mishra, M., Prakash, S.: Mobile forensics challenges and admissibility of electronic evidences in India, In: Proceedings - 5th International Conference on Computational Intelligence and Communication Networks, CICN 2013, pp. 237–242. (2013). <https://doi.org/10.1109/CICN.2013.57>
29. Zawoad, S., Hasan, R.: Digital Forensics in the Age of Big Data: Challenges, Approaches, and Opportunities, In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, IEEE, pp. 1320–1325. (2015). <https://doi.org/10.1109/HPCC-CSS-ICCESS.2015.305>
30. Casino, F., et al.: Research Trends, Challenges, and Emerging Topics in Digital Forensics: A Review of Reviews. *IEEE Access Institute of Electrical and Electronics Engineers Inc* **10**, 25464–25493 (2022). <https://doi.org/10.1109/ACCESS.2022.3154059>
31. Barmapsalou, K., Cruz, T., Monteiro, E., Simoes, P.: Current and future trends in mobile device forensics: A survey. *ACM Comput Surv* **51**(3), 1–31 (2018). <https://doi.org/10.1145/3177847>
32. Said, H., Yousif, A., Humaid, H.: iPhone forensics techniques and crime investigation. In: The 2011 International Conference and Workshop on Current Trends in Information Technology (CTIT 11), Dubai, United Arab Emirates, pp. 120–125 (2011). <https://doi.org/10.1109/CTIT.2011.6107946>
33. Kang, S.H., Park, K.Y., Kim, J.: Cost effective data wiping methods for mobile phone. *Multimed Tools Appl* **71**(2), 643–655 (2014). <https://doi.org/10.1007/s11042-013-1603-9>
34. Pandey, A.K., et al.: Current Challenges of Digital Forensics in Cyber Security. (2020). <https://doi.org/10.4018/978-1-7998-1558-7.ch003>
35. Almeahmadi, T., Batarfi, O.: Impact of Android Phone Rooting on User Data Integrity in Mobile Forensics, In: 2nd International Conference on Computer Applications & Information Security (ICCAIS' 2019) : 01–03 May, 2019 Riyadh, Kingdom of Saudi Arabia, pp. 1–6. (2019)
36. Chanajitt, R., Viriyasitavat, W., Choo, K.K.R.: Forensic analysis and security assessment of Android m-banking apps. *Australian Journal of Forensic Sciences* **50**(1), 3–19 (2018). <https://doi.org/10.1080/00450618.2016.1182589>
37. Janarthanan, T., Bagheri, M., Zargari, S.: IoT Forensics: An Overview of the Current Issues and Challenges. *Advanced Sciences and Technologies for Security Applications* (2021). https://doi.org/10.1007/978-3-030-60425-7_10

2. Neware, R., Khan, A.: Cloud Computing Digital Forensic challenges, In: Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018), pp. 1–3. (2018)
3. Mohay, G., Technical Challenges and Directions for Digital Forensics. (2005). [Online]. Available: www.e-evidence.info/cellular.html
4. Barmapsalou, K., Cruz, T., Monteiro, E., Simoes, P.: Mobile Forensic Data Analysis: Suspicious Pattern Detection in Mobile Evidence. *IEEE Access* **6**, 59705–59727 (2018). <https://doi.org/10.1109/ACCESS.2018.2875068>
5. Chen, S., Hao, X., Luo, M.: Research of mobile forensic software system based on windows mobile. *International Conference on Wireless Networks and Information Systems, WNIS 2009*, 366–369 (2009). <https://doi.org/10.1109/WNIS.2009.32>
6. Ayers, R., Brothers, S., Jansen, W.: Guidelines on mobile device forensics, Gaithersburg, Maryland (2014). <https://doi.org/10.6028/NIST.SP.800-101r1>
7. Hummert, C., Pawlaszyk, D.: *Mobile Forensics – The File Format Handbook*. Springer International Publishing (2022). <https://doi.org/10.1007/978-3-030-98467-0>
8. Burrows, C., Zadeh, P.B.: A Mobile Forensic Investigation into Steganography. [Online]. Available: <http://www.ericsson.com/res/docs/2015/ericsson->
9. Yusof, M.N., Mahmod, R., Abdullah, M.T., Dehghantanha.: Mobile Forensic Data Acquisition in Firefox OS, In: *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 2014 Third International Conference on : date April 29 2014–May 1 2014, pp. 691–5. (2014)
10. Humphries, G., Nordvik, R., Manifavas, H., Copley, P., Sorell, M.: Law enforcement educational challenges for mobile forensics. *Forensic Science International: Digital Investigation* **38**, 301129 (2021). <https://doi.org/10.1016/j.fsidi.2021.301129>
11. Ghafarian, A.: Forensics Analysis of Cloud Computing Services, In: *Science and Information Conference 2015*, pp. 1–5. [Online]. (2015) Available: www.conference.thesai.org
12. Sonia Akter, S., Shahriar Rahman, M.: Cloud Forensic: Issues, Challenges and Solution Models, *ArXiv*, pp. 2–23. (2023)
13. Lim, S.Y., Johan, A., Daud, P., Ismail, N.A.: Dropbox forensics: Forensic analysis of a cloud storage service. *International Journal of Engineering Trends and Technology* **1**, 45–49 (2020). <https://doi.org/10.14445/22315381/CATI3P207>
14. Choo, K.-K.R., Esposito, C., Castiglione, A.: Evidence and Forensics in the Cloud: Challenges and Future Research Directions. *IEEE Cloud Computing* **4**(3), 1–6 (2017)
15. Sibiya, G., Venter, H.S., Fogwill, T.: Digital Forensics in the Cloud: The State of the Art, In: *2015 IST-Africa Conference : 06–08 May 2015, Lilongwe, Malawi, Malawi: IEEE* (2015)
16. Shah, J.J., Malik, L.G.: Cloud forensics: Issues and challenges. In: *International Conference on Emerging Trends in Engineering and Technology, ICETET*, IEEE Computer Society, pp. 138–139. (2013). <https://doi.org/10.1109/ICETET.2013.44>
17. Dhake, B., Limaye, H., Motwani, D.: Cloud Forensics: Threat Assessment and Proposed Mitigations. In: *2022 International Conference for Advancement in Technology, ICONAT 2022*, Institute of Electrical and Electronics Engineers Inc, (2022). <https://doi.org/10.1109/ICONAT53423.2022.9725922>
18. Lutta, P., Sedky, M., Hassan, M., Jayawickrama, U., Bakhtiar Bastaki, B.: The complexity of internet of things forensics: A state-of-the-art review. *Forensic Science International: Digital Investigation* **38**, 301210 (2021). <https://doi.org/10.1016/j.fsidi.2021.301210>
19. Fernandes, R., Colaco, R.M.: A New Era of Digital Forensics in the form of Cloud Forensics: A Review July, 2020, In: *Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020)*, pp. 1–6. (2020)
20. Ayers, R., Brothers, S., Jansen, W.: Guidelines on Cell Phone Forensics Guidelines on Mobile Device Forensics. *Archived NIST Technical Series Publication Archived Publication*, vol. 1 (2007)

must adopt sophisticated techniques to bypass encryption and navigate authorization mechanisms [57].

The integration of blockchain and cryptographic technologies may redefine data integrity and chain of custody practices, enhancing forensic evidence reliability [74]. However, these innovations introduce complexities in data retrieval and analysis, necessitating continuous skill advancement in cloud architecture, network protocols, and cryptography to navigate distributed data storage and security [43].

6.3 Anticipated Challenges & Opportunities

The evolution of mobile and cloud forensics presents both challenges and opportunities. The complexity of data structures and encryption mechanisms demands refinement of data acquisition and decryption techniques. The increasing use of ephemeral messaging and transient data in both environments requires novel approaches for evidence capture and preservation [75, 76].

Though privacy-focused legislation and public awareness may limit data access, this presents potential for creating forensic investigation tactics that respect individual rights [60]. As the lines between mobile and cloud environments become less distinct, forensic professionals can employ integrated tools and procedures to produce an extensive digital narrative [77, 78].

Standardized procedures, innovative technologies, and educational materials will be made available through collaboration between academics, business, and law enforcement [79], enhancing the discipline's effectiveness and encouraging sound answers to new challenges [80].

7 Conclusion

To summarize, the field of digital forensics investigations, particularly in the domains of mobile and cloud forensics, is fraught with complexities. The constantly changing mobile device landscape, with its variety of operating systems and applications, necessitates ongoing adaptability and technical know-how. Furthermore, the dynamic and decentralized nature of cloud environments presents challenges that call for novel extraction, analysis, and preservation techniques for digital evidence. These difficulties are made more difficult by the quick speed of technical innovation, privacy concerns, and legal constraints. Collaboration between experts, the creation of cutting-edge tools, and a profound knowledge of both technological and legal issues will be vital in overcoming these obstacles and guaranteeing the integrity of digital forensic investigations in the face of the expanding digital landscape.

References

1. Venter, H.S.: Mobile Forensics using the Harmonised Digital Forensic Investigation Process, Information security for South Africa (ISSA), pp. 1–10. Johannesburg South Africa, IEEE, Sandton (2014)

5.5 Enhancing Forensic Tools & Techniques

Mobile and cloud forensics face challenges in navigating encrypted data and robust security measures. Professionals use techniques like brute-force attacks and cryptographic analysis to decrypt data, utilizing cryptographic expertise and methodologies [52]. Therefore, forensic tools and techniques have become crucial for mobile and cloud forensics evolution. Advancements in machine learning, artificial intelligence, and big data analytics improve efficiency and accuracy [68]. Furthermore, collaborative partnerships between experts and developers infuse innovation into investigative methodologies, enabling forensic professionals to navigate complex scenarios while maintaining evidentiary integrity [68].

5.6 Improving Legal & Regulatory Frameworks

Mobile and cloud forensics require legal and privacy awareness, compliance with regulations, and ethical practices. Acquiring permissions, consent, and data protection laws ensures admissibility and reliability of forensic evidence in legal proceedings [60].

Clear guidelines for evidence acquisition, admissibility, and chain of custody enhance investigative processes, promoting equitable application and public trust [69]. Improved legal and regulatory frameworks are crucial for mobile and cloud forensics. Additionally, collaborative efforts between experts, policymakers, and practitioners align legal standards with technological advancements, safeguarding privacy rights and ethical considerations [60].

6 Future Trends in Mobile & Cloud Forensics

6.1 Mobile Technology Advancements

The development of mobile technology, such as fifth generation (5G) connectivity, virtual reality (VR), augmented reality (AR), and pervasive computing also known as the internet of things (IoT) devices, will have a huge impact on the future of mobile forensics [70].

The acquisition, analysis, and interpretation of a larger variety of data formats will provide issues for forensic practitioners as mobile devices become increasingly integrated into daily life and company processes. To do this, new approaches [71], improved encryption techniques, and multimedia fusion must be developed [72].

Data analysis and pattern identification will be fundamentally influenced by artificial intelligence and machine learning, necessitating the adaptation and adoption of these technological breakthroughs by forensic professionals [73]. To successfully traverse the changing mobile world after this paradigm change, interdisciplinary collaboration and ongoing education are required.

6.2 Cloud Computing & Security Innovations

Cloud forensics is closely linked to the evolution of cloud computing and security innovations. As digital evidence spreads across complex infrastructures, investigators

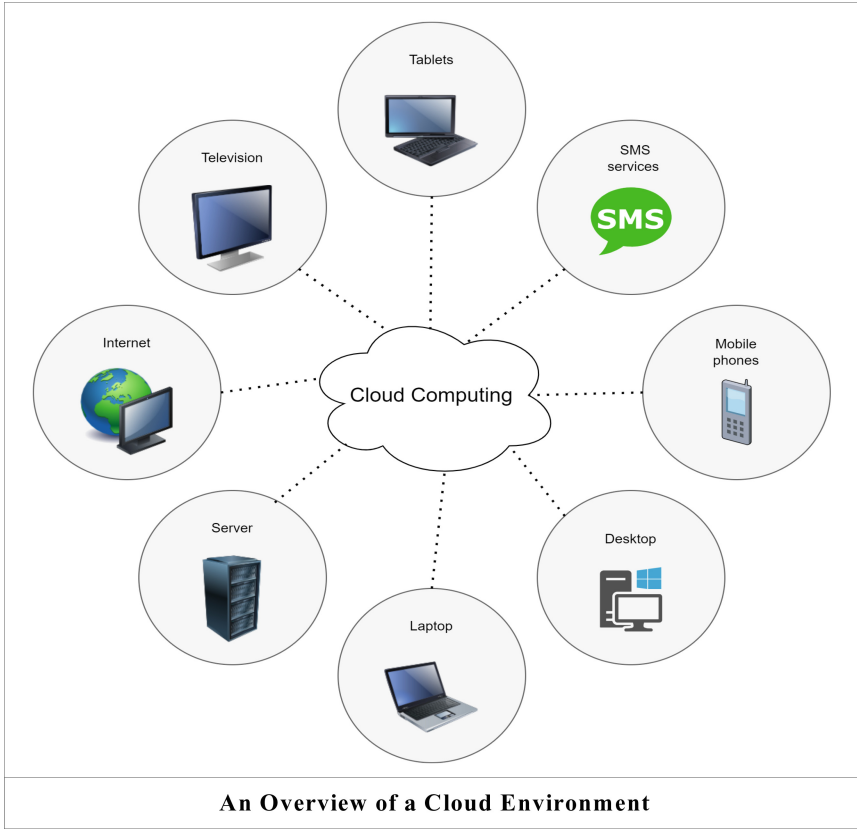


Fig. 2. An overview of a cloud environment setup.

5.3 Information Sharing & Collaboration

Collaborative engagement and information sharing are crucial for mobile and cloud forensics advancement. These platforms enable multidisciplinary expertise to fuse, fostering innovative solutions and enhancing the field's capacity to tackle complex scenarios [30]. Furthermore, collaboration plays a crucial role in enabling lawful data access, expediting the investigation process, and preserving data integrity, privacy regulations, and critical evidence acquisition [53-59].

5.4 Continuous Training & Skills Development

Continuous training and skill refinement are crucial for professionals in mobile and cloud forensics to remain competent and adapt to evolving landscapes [20]. This culture fosters adaptability and optimal investigative efficacy.

4.2 Case Study 2: Cloud Forensics

In the case of Marli van Breda Murder, the accused (Henri Christo Van Brend) was indicted for the murder of his family members, namely his parents, brother, and sister, and for an attempted murder for trying to kill his sister. He was also charged for defeating and obstructing the administration of justice [64].

Cloud forensics was conducted for the purpose of evaluating acquired digital forensic evidence from several devices, including smartphones and laptops. To establish time-frames and acquire evidence essential to the trial, cloud forensic experts were involved in the data extraction and analysis from multiple cloud-based services, emails, and other digital sources. The accused was then charged with three life sentences in prison for count one to three- and fifteen-years imprisonment for count four- and twelve-years imprisonment for count five.

Another instance was of the horrible death of mentally ill patients who were moved from one hospital facility to the other, where there were no resources for them. This case of Life Esidimeni tragedy resulted in the deaths of the moved mentally ill patients [65]. Cloud forensic was done on all the available digital records to uncover the decision-making process that led to the transfers of the patients. A commission of enquiry was established, and evidence was presented even though justice has still not been served [65].

5 Discussion and Best Practices

5.1 Forensic Acquisition & Imaging Methods

Forensic acquisition and imaging are crucial in mobile and cloud forensics, ensuring data integrity and maintaining the evidentiary chain of custody [49]. Traditional methods, like physical and logical acquisitions, are complemented by advanced techniques like live acquisitions [31]. Acquiring methods must consider device characteristics, proprietary software, security measures, and adhere to established best practices [50]. Data can also be shared across multiple platforms via the cloud environment. An overview of a general cloud environment setup is shown in Fig. 2.

5.2 Mobile Forensics Standardization

A foundational element of meticulous investigative procedures is the development of standardized methods in both mobile and cloud forensics. The processes for gathering, analyzing, and interpreting data are standardized to ensure consistency, dependability, and uniformity. The reliability and admissibility of evidence in court processes are increased when forensic experts follow established protocols because they are more effective at navigating complexity [66, 67]. A reliable and organized approach to digital investigations is fostered by the collaborative establishment of complete standards that consider legal, ethical, and technological factors.

Table 2. (continued)

Challenge	Description of Challenge
Challenges of Live Forensics in Cloud Environments	Conducting live forensics within cloud environments introduces distinct challenges due to the complexity of virtualized systems and the potential for unintended data alterations. Effectively addressing the challenges of live cloud forensics requires specialized expertise in virtual machine forensics and the implementation of non-intrusive methodologies to acquire real-time data without disrupting cloud services [48]

4.1 Case Study 1: Mobile Forensics

There are several instances of successful prosecution in the South African courts, that emanated from mobile forensics. The famous and emotional Hannah Cornelius Case which involves kidnapping, rape, murder and robbery was cracked using the evidence from a mobile device. The accused persons (Vernon Junaid Witbooi, Geraldo Parsons, Eben Van Niekerk, and Nashville Julius) were all indicted and charged with robbery with aggravated circumstances, kidnapping, attempted murder, rape, and murder [61]. When digital forensics was conducted on the devices, mobile forensics investigation retrieved the movement patterns of the suspects using Cellular tower information, GPS locations, and communication records from the mobile phones of the suspects and victims. All the accused persons were convicted of their indictment.

Another widely televised case was of the Paralympian Oscar Pistorius who shot and killed his then girlfriend River Steenkamp on the eve of valentine in 2013. The accused was initially convicted of capable homicide, but the appeal turned the charge into murder. Mobile forensic was performed on his mobile devices to reconstruct the events that may have led to the shooting. The suspect was successfully convicted and sentenced to thirteen (13) years in prison [62].

In the case of Tshegofatso Pule, the boyfriend of the deceased was charged with premeditated murder of his 8-month pregnant girlfriend. His mobile devices were seized for the purpose of digital forensic investigations. The mobile forensic revealed that there were calls, WhatsApp and text messages communication between the boyfriend and the hit man. The hitman ultimately turned himself a state witness and got a lesser charge, and the boyfriend was sentenced to life imprisonment [63].

Another example is of a South African national soccer team captain, Senzo Meyiwa, who was shot and killed in an alleged house robbery. In this case, mobile records from the suspects deduced that they communicated, and the suspects know each other. The case is still on trial, but mobile forensics has played a crucial role in uncovering clues about the hidden communications between the suspects and the late soccer star, then girlfriend.

Table 2. (continued)

Challenge	Description of Challenge
Cross-Border Data Requests and Legal Considerations	Cloud forensics involves cross-border data storage, requiring strict adherence to protocols and legal issues. Collaboration with foreign organizations and cloud service providers ensures evidence acquisition, navigating legal systems and data privacy laws [44]
Data Fragmentation in Distributed Cloud Storage	Data fragmentation results from the distributed storage architecture used in cloud environments, where data is scattered across various physical locations [16]. It is extremely difficult to meticulously piece together disparate facts to create a consistent evidentiary narrative. To facilitate the seamless integration of fragmented data into the investigative process, forensic investigators must show proficiency in data reconstruction techniques and correlation approaches [10]
Cloud Service Misconfigurations and Security Incidents	Misconfigurations and security incidents can jeopardize the integrity and confidentiality of data in cloud settings. A thorough knowledge of cloud infrastructure, networking, and security procedures is required to recognize and analyze[45] cloud service misconfigurations and security breaches. To preserve evidence and minimize potential harm, security issues must be promptly detected and remedied [2]
Dynamic IP Addressing and Tracking Network Traffic	In cloud systems, IP addresses are assigned dynamically, which makes it difficult to follow network activity and attribute it to certain sources during forensic investigations [46]]. Tracing dynamic IP addresses and identifying communication patterns within the cloud infrastructure require knowledge of network forensics and the use of advanced network analysis tools [46]
Preservation of Metadata in the Cloud	A significant problem in cloud forensics is the preservation of the metadata linked to cloud-stored data. When conducting investigations, metadata is essential for proving the legitimacy, provenance, and contextual significance of the data. To ensure the admissibility and dependability of gathered evidence, careful preservation measures must be put in place to prevent unintentional metadata alteration or loss [47]

(continued)

Table 2. *(continued)*

Challenge	Description of Challenge
Log Collection, Retention, and Analysis	Critical elements of cloud forensics include the gathering and examination of logs from cloud infrastructures. To extract crucial information about system activities, user interactions, and potential security incidents from the collection and retention of logs from various sources, advanced log analysis techniques are required [41]. To extract valuable insights from log data, forensic investigators must have a thorough understanding of log gathering procedures and analytical tools [41]
Collaboration & Cooperation with Cloud Service Providers	An essential component of cloud forensics is effective coordination and cooperation with cloud service providers. Clear communication channels and the development of cooperative partnerships with cloud providers are required to obtain the essential data access and cooperation. This kind of cooperation makes it easier to acquire data legally and securely, which improves the effectiveness and precision of forensic investigations [41]
Coping with Rapid Data Growth in Cloud Environments	The rapid growth of cloud data volumes presents challenges in handling, processing, and analysing vast amounts of information during forensic investigations [41]. Scalable forensic methodologies and efficient data handling techniques are needed to expedite investigations without compromising accuracy. Continuous research and development are necessary to ensure timely and efficient retrieval of relevant evidence [42]
Handling Transient Data in the Cloud	Due to the dynamic generation, modification, and deletion of virtual resources, forensic practitioners encounter difficulties when trying to capture and preserve volatile material in cloud systems. For effective analysis and preservation without interrupting cloud services, live forensic expertise is essential [43]

(continued)

3.2 Cloud Forensics: Major Challenges

4 Real-World Instances

There are some real-world instances where mobile and cloud forensics helped the courts in terms of successful convictions in South Africa. These real-world examples discussed in the next subsection highlight the value of digital evidence in contemporary court cases and the difficulties forensic professionals confront when working with developing technology and digital platforms.

Table 2. Tables of Cloud Forensics Challenges

Challenge	Description of Challenge
Data Location and Jurisdiction Complexities	Due to geographical dispersion and the involvement of service providers, cloud forensics encounters difficulties in locating data and identifying its jurisdictional consequences. It takes specialized legal knowledge, global collaboration, and cutting-edge technical approaches to resolve these problems [18]
Multi-Tenancy and Data Isolation	Strong data isolation methods are needed in multi-tenant cloud infrastructures to protect the integrity and confidentiality of forensic investigations. For effective data management and customer-specific data security, it is essential to properly identify and separate customer-specific data [19]
Data Encryption and Decryption	Forensic investigators trying to access and decrypt cloud-stored data face significant difficulties due to the widespread usage of strong data encryption algorithms by cloud service providers. To overcome the obstacles preventing data access, the encryption processes used to strengthen cloud data privacy call for sophisticated cryptographic knowledge and cutting-edge deciphering approaches [39]
Responsibility and Security Model in the Cloud	Cloud forensics faces challenges in delineating responsibilities and security measures between service providers and customers [40]. The shared responsibility model requires understanding security responsibilities and identifying responsible parties in case of breaches or data compromises

(continued)

Table 1. *(continued)*

Challenge	Description of Challenge
Data Fragmentation and Reconstruction	Data fragmentation, in which relevant data may be scattered across several physical sectors or file locations, is a common occurrence in mobile devices [7]. Data carving competence and precise linkage of scattered fragments are required to ensure the coherent reconstruction of fragmented data and to rebuild a coherent evidence picture [34]. To meet this challenge, data reconstruction must be approached with care and thoroughness, assuring the accuracy and validity of the evidence that is recovered
Overwriting and Data Corruption Risks	When mobile devices are handled incorrectly during forensic investigations, there is a chance that important evidence’s integrity could be jeopardized by accidental data overwriting or corruption. This problem highlights the importance of following best practices to assure data preservation and the accuracy of investigative results. It also highlights the necessity of using strict forensic procedures and preservation methods to prevent against data tampering [35]
Technological Advancements and Updates	The field of mobile forensics continues to face difficulties due to the constant evolution of mobile technology, which is characterized by regular operating system updates and hardware improvements. To ensure that forensic procedures and tools are compatible with the modern mobile landscape, constant research and development activities are required due to the quick rate of technological advancement. As a result, to continue being effective in their research, mobile forensic practitioners need to keep up with all current technology developments [36]
Legal and Privacy Issues	Legal and privacy considerations of many different types have a significant impact on how mobile forensic investigations are conducted [37]. In the context of gathering digital evidence, it is crucial to ensure strict adherence to legal requirements, obtain any appropriate warrants, and protect the privacy rights of everyone concerned [38]. The complex interplay between technology advancements and legal restrictions emphasizes the necessity of competent legal representation and in-depth familiarity with relevant legislation to negotiate this treacherous terrain with caution

Table 1. (*continued*)

Challenge	Description of Challenge
Locked and Password-Protected Devices	The ubiquity of password-protected and locked devices is a significant challenge for mobile forensic investigators. Data extraction procedures become more complex because of the strict access constraints that protect device contents, especially when dealing with reluctant or unresponsive device owners [31, 32]. To overcome these difficulties, specialist forensic methods must be used to get beyond the authentication obstacles that prevent the collection of relevant evidence
Cloud Services and Data Synchronization	The prevalence of cloud services and data synchronization in modern mobile forensics creates complex scenarios where crucial digital evidence may be stored in distant cloud repositories rather than only on the physical device. The use of cloud-based data synchronization and storage adds layers of complexity to the investigation process because it necessitates abiding by legal requirements and working cooperatively with cloud service providers to gain access to, and secure crucial data kept in off-site locations [31]
Third-Party Applications and Data Access	In their efforts to obtain and understand application-specific data, forensic practitioners face a difficult issue due to the widespread use of third-party applications within mobile ecosystems. To successfully recover crucial evidence, it is necessary to have a thorough understanding of various application frameworks. Third-party programs may differ in their encryption and data storage procedures [31, 32]. Continuous research and flexibility in response to shifting mobile application paradigms are necessary to ensure the seamless integration of various third-party applications into the forensic workflow
Recovery of Deleted Data	Recovering deleted data from mobile devices is a difficult task since contemporary smartphone architectures include tools like TRIM or similar technologies that hasten the deletion of idle storage areas [24]. These data-wiping methods necessitate forensic methodologies and specialist knowledge to identify and recreate any remaining traces of wiped data, assuring a thorough and accurate inquiry [33]

(continued)

3 Mobile and Cloud Forensics Challenges

The challenges brought about by mobile and cloud technologies in relation to forensic investigation processes have resulted in research evolving to cover these domains. The key challenges identified through literature are depicted in Table 1 and Table 2 below.

3.1 Mobile Forensics: Major Challenges

Table 1. Tables of Mobile Forensics Challenges

Challenge	Description of Challenge
Device Diversity & Fragmentation	Due to the massive diversity of mobile devices, which are characterized by a wide range of unique hardware configurations, operating systems, and proprietary applications, the field of mobile forensics faces a difficult task [28–31]. The development of forensic procedures and technologies capable of accommodating this broad variety of device types is required due to the significant fragmentation within the mobile ecosystem [31]. Considering this steadily growing device diversity, determining universal applicability becomes challenging, necessitating the need for adaptable forensic techniques to efficiently handle the range of mobile devices found throughout investigations
Encryption & Security Measures	The use of smartphones has grown tremendously, and this has increased the importance placed on using strong encryption and security measures to protect critical data. Mobile forensic practitioners have a significant barrier when trying to access and decipher encrypted data without the necessary passcodes or biometric credentials [31, 32]. To overcome the obstacles to data decryption, cutting-edge cryptographic knowledge and novel methodology must be developed. The deployment of complex encryption mechanisms to safeguard user information manifests as a barrier to conventional forensic procedures [31, 32]

(continued)

evidence, these disciplines demand the union of technical proficiency, legal grasp, and methodological rigor [19].

1.3 Research Objectives

The major goal is to explore the many complex problems that arise in the fields of mobile and cloud forensics. This study attempts to provide a thorough understanding of the challenges that forensic practitioners face by methodically investigating the complexities involved in the extraction, processing, and interpretation of digital evidence from mobile devices and cloud environments.

The study attempts to uncover developing trends, cutting-edge approaches, and prospective solutions to address these difficulties through a thorough review of the existing situation. The study also aims to highlight the significance of standard operating procedures, interdisciplinary cooperation, and ongoing skill improvement as the foundations of efficient mobile and cloud forensics. This research study intends to develop digital investigative methods and improve the integrity of forensic results by offering insight on the challenges and opportunities within these disciplines.

2 Methodology

This study followed a systematic review protocol which involved a comprehensive search for relevant literature on cloud and mobile forensics challenges [22]. Several guidelines have been developed for conducting systematic reviews [23–26]. However, to achieve its main goal of synthesizing and interpreting previous research on cloud and mobile forensics challenges, this study followed the systematic review approach by [27]. The stages involved in this synthesis are as follows: Framing, search and assessment, synthesis, and interpretation.

2.1 Framing

Digital forensics investigations have grown extremely complex due to the evolving and extensive use of mobile and cloud technologies. This has resulted in forensic practitioners encountering new challenges during the different phases of the investigation process. This study uses existing literature to highlight these challenges. It further provides real-world instances where the use of mobile and cloud forensics was able to solve cases.

2.2 Search and Assessment

The online databases used in this review were IEEE Xplore, SpringerLink, Google Scholar, and Science Direct. The search strings used were: “mobile forensics” OR “mobile forensics challenges” OR “challenges in mobile forensics”, “cloud forensics” OR “cloud forensics challenges” OR “challenges in cloud forensics”. From the results obtained, paper abstracts were read to identify relevant papers and discard irrelevant ones. Once all the irrelevant papers were filter out, full text reading of the remaining papers was done.

and mined from digital devices, to ensure that there is trust from the side of the mobile and cloud system users. A general overview of a digital forensic investigations and the types of forensic investigations is depicted in Fig. 1.

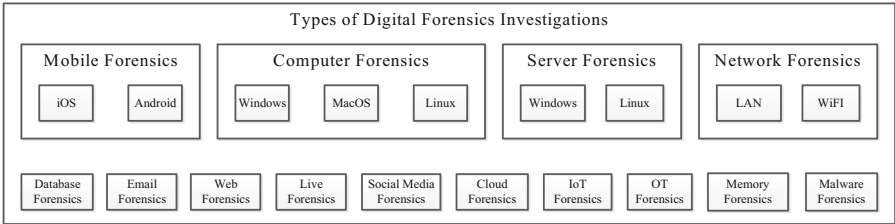


Fig. 1. Overview of the types of Digital Forensic Investigations

1.1 Importance of Mobile & Cloud Forensics in Modern Investigations

The scope and complexity of digital investigations have been redefined by the growing use of mobile devices and the migration of data to cloud systems. The digital traces left in cloud storage and on mobile devices provide priceless insights into user behavior, data transactions, and communication patterns.

These observations have been helpful in a variety of situations, including criminal investigations, civil lawsuits, cybersecurity events, business compliance audits, and more. In an increasingly digital environment, mobile and cloud forensics serve as accountability pillars by making it possible to find hidden evidence, follow data trails, and clarify the chronology of events [18]. The significance of these forensic disciplines cannot be overstated, as forensic disciplines assist in [19] building confidence, maintaining data integrity, and applying justice in technology environments that continue to expand [19].

1.2 Overview of Mobile Forensics & Cloud Forensics

A variety of approaches are included in mobile forensics that are targeted at retrieving, analyzing, and deciphering data from mobile devices. The variety of information included on these devices, from text messages and call logs to application usage history and geolocation information [6, 20, 21], offers a profound understanding of user behavior.

Conversely, cloud forensics tackles the intricate challenges posed by virtualized environments, involving the acquisition and analysis of data stored remotely in cloud service providers’ infrastructure. This includes unearthing evidence from virtual machines, deciphering encrypted data, and discerning the implications of data fragmentation across distributed cloud storage [19].

In addition, cloud forensics addresses the complex issues brought on by virtualized settings and entails the capture and analysis of data kept remotely in the infrastructure of cloud service providers. To ensure the precise and admissible presentation of



Digital Forensics Investigations: Major Challenges in Mobile and Cloud Forensics

Tanita Singano, Norman Nelufule^(✉), Boitumelo Nkwe, Kele Masemola, Daniel Shadung, Zamo Ngubane, Ntombizodwa Thwala, and Japhtalina Mokoena

Defence and Security Cluster, Information and Cybersecurity Centre (ICSC), Council for Scientific and Industrial Research (CSIR), Pretoria 0184, Brummeria, South Africa
nnelufule@csir.co.za

Abstract. Due to the extensive use of mobile devices and cloud computing, digital forensics investigations have grown increasingly complex. The main challenges that forensic investigators in the fields of mobile and cloud forensics encounter are discussed in this study. The proliferation of various devices, operating systems, and applications creates challenges for data capture, extraction, and interpretation in the context of mobile forensics. Additionally, the process is made more difficult by the usage of privacy and encryption tools. The extraction and analysis of digital evidence from distributed, frequently encrypted cloud systems is the focus of cloud forensics, on the other hand. Significant barriers include questions of jurisdiction, data ownership, and secure access. This presentation examines the evolving field of digital forensics with a particular emphasis on the complex problems that mobile and cloud technologies present.

Keywords: Digital Forensic · Digital Evidence · Mobile Forensics · Cloud Forensics

1 Introduction

The increased use of mobile devices and the general use of cloud-based services have transformed how individuals and organizations interact with and store digital data in the modern cyber environment. The important domains of mobile and cloud forensics, which entail the methodical extraction, analysis, and interpretation of digital evidence from mobile devices and cloud settings, were created due to this paradigm shift [1–3]. While cloud forensics focuses on the investigation of data stored inside distant cloud infrastructures, mobile forensics deals with the investigation of digital artifacts present within smartphones, tablets, and wearable devices [4–17].

To meet the expectations of the legal, corporate, and law enforcement sectors, the convergence of various disciplines is a crucial aspect of contemporary digital investigations. The complexities of these forensic disciplines are dynamically expanding, bringing previously unknown challenges and opportunities, as the border between mobile and cloud technology continues to blur. It is important to ensure that digital evidence can be traced

For where the clients will be deployed Raspberry PIs are advantageous, because they are 24.5% cheaper than the cheapest x86-64 client that was investigated. This provides an extra seat for every four Cloudgate seats that are deployed, in a rural area where accessibility is more important than speed that extra seat makes a big difference. Also when considering tens or hundreds of deployments those 24.5% savings start to add up.

References

1. Andrew, M.: Pinet end of life announcement (2020). <http://pinet.org.uk/blog/2020/10/27/PiNet-end-of-life.html>. Accessed 01 Oct 2021
2. Debian Community. dnsmasq (2020). <https://wiki.debian.org/dnsmasq>. Accessed 02 Oct 2021
3. Hollingworth, G.: The raspberry pi pserver tool (2018). <https://www.raspberrypi.org/blog/pserver/>. Accessed 27 June 2021
4. iPXE Community. ipxe (2021). <https://ipxe.org/>. Accessed 14 May 2021
5. Janina, A.: Teaching with Raspberry PIs and PiNet (2017). <https://www.raspberrypi.org/blog/teaching-pinet/>. Accessed 27 June 2021
6. Jing, J., Helal, A.S., Elmagarmid, A.: Client-server computing in mobile environments. *ACM Comput. Surv. (CSUR)* **31**(2), 117–157 (1999)
7. kernel.org. Squashfs 4.0 filesystem. <https://www.kernel.org/doc/html/latest/filesystems/squashfs.html>. Accessed 01 Oct 2021
8. Lee, D., Won, Y.: Booting linux faster. In: 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, pp. 665–668. IEEE (2012)
9. LTSP. Linux terminal server project, about. <https://ltsp.org/>. Accessed 14 May 2021
10. Lucy, H.: Raspberry pi 4 vs raspberry pi 3b+ (2020). <https://magpi.raspberrypi.org/articles/raspberry-pi-4-vs-raspberry-pi-3b-plus>. Accessed 03 Oct 2021
11. Maga, D., Hiebel, M., Knermann, C.: Comparison of two ICT solutions: desktop PC versus thin client computing. *Int. J. Life Cycle Assess.* **18**(4), 861–871 (2013)
12. Raspberry Pi Foundation. What is a Raspberry Pi? <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. Accessed 13 May 2021
13. Rodeh, O., Bacik, J., Mason, C.: Btrfs: the linux b-tree filesystem. *ACM Trans. Storage (TOS)* **9**(3), 1–32 (2013)
14. Siebörger, I., Terzoli, A., Hodgkinson-Williams, C.: LTSP DNS round robin clusters: green technology access enablers for telecommunication services in marginalised communities. In: Proceedings of the Southern African Telecommunication and Networks Conference (SATNAC), East London Convention Centre, pp. 393–398 (2011)
15. Terzoli, A., Siebörger, I., Tsietsi, M., Gumbo, S.: Digital inclusion: a model for e-infrastructure and e-services in developing countries. In: International Conference on e-Infrastructure and e-Services for Developing Countries, pp. 85–98. Springer (2017)
16. Thinyane, M., Dalvit, L., Terzoli, A., Clayton, P.: The internet in rural communities: unrestricted and contextualized. *ICT Africa* **13**, 15–25 (2008)
17. Uimonen, P., Hellström, J.: ICT4D donor agencies and networks. In: The International Encyclopedia of Digital Communication and Society, pp. 1–9 (2015)
18. W3.org. WebDriver, W3C Working Draft (2021). <https://www.w3.org/TR/webdriver/>. Accessed 21 Sept 2021

6.3 Cost of Workstation

As illustrated in Table 3 a Raspberry Pi 4B workstation is 24.5% cheaper than the Cloudgate client this might not look like a significant difference. However for every four Cloudgate workstations that are bought that would have provided six Raspberry Pi 4B workstations. These numbers add up quickly when considering tens or even hundreds of deployments in a lab or community centre.

6.4 Network Booting and File Management

In the duration of the research project the usage of network booting shaved off several hours in the management of the different clients, installing software, transferring files between devices, booting the systems, etc.

The Raspberry PIs came with version 2 of the Python programming language, but the Selenium automation suite works only with version 3. Therefore version 3 had to be installed on both setups, and network booting cut the installation time in half. Because the Raspberry Pi clients boot from the same image this was installed once and made available to both clients, this goes for all other software that were installed. Network booting also cut all the installation times to a quarter of the time it would have taken for the x86-64 clients. For the x86-64 clients the required programs had to be installed only in the LTSP server this made them available to all x86-64 clients.

File management is seamless, after downloading, creating or updating a file on one machine, and it does not matter whether it is a Raspberry Pi or x86-64 client. The files are automatically made available to all the other devices that login as the same user that has the file saved. Network booting therefore saves the administrator of these clients a lot of time. Considering the fact that in rural areas skills to manage the clients is limited this is a useful feature. The next section concludes the paper.

7 Conclusions

The main objective of this research project was to investigate cost-effective computing infrastructure that can be used in schools or community centres using Raspberry PIs. As a solution to this a LAN that can boot both Raspberry PIs and x86-64 clients was created. The Linux Terminal Server Project was used to implement this research project and support for Raspberry PIs was manually added.

In general Raspberry Pi clients performed poorly compared to x86-64 clients, this is because the Intel processors are well designed for desktop applications. However Raspberry Pi clients are still usable as desktop computers. Given that lightweight applications and the right OS are used they do not crash or significantly slow down. With the Raspberry Pi 4B providing a smoother experience that is comparable to x86-64 clients.

Table 3. Price comparison of Clients

Shop	Intel NUC	Cloudgate	Mecer	Pi 3B+	Pi 4B
TakeaLot	R6020.00			R3649.00	R4549.00
Makro	R5889.00				
Cloudgate		R5404.00			
TechHut			R6550.00		
BidorBuy			R7500.00		
PiShop				R3360.00	R4080.00

6 Findings and Discussion

This section presents the findings from the different tests that were run on the two types of clients and general discussion about all the factors that contributed in the results. Because the deployments will make use of Raspberry Pi4B clients, only the Pi4B client is mentioned, and compared to the cheapest x86-64 client, Cloudgate.

6.1 Difference Between ARM and X86-64

Raspberry Pi clients use an ARM processor which is designed for lightweight applications. On the other hand x86-64 clients are built for heavyweight applications. This is well illustrated in Fig. 4, where the CPU benchmark test overloads the clients with complex calculations that heavily utilise the CPU. The Raspberry Pi clients perform poorly as compared to x86-64 clients. This is because of the difference in their architecture. Figure 3 illustrates the temperature of the clients when they are overloaded but not doing a lot of CPU heavy calculations and this performance difference is not seen here.

6.2 Testing Chromium and LibreOffice

As seen on Table 1 Raspberry Pi clients performed poorly compared to x86-64 clients with the Pi 4B being 6.6 times worse than the Cloudgate client. This 6X difference is however not seen in the LibreOffice tests. The comparison of these two clients on the LibreOffice Writer test is 4.6X in favor of the Cloudgate client. For LibreOffice Calc it is also 4.6X in favor of the Cloudgate client.

The ratio of 6.6X difference is also not observed in the manual tests of Chromium, where the Cloudgate client is 2.5-5X faster depending on the website that is visited. Raspberry Pi clients performed worse at loading the “RUConnected” website than other websites. This could be because of the different versions in Chromium and its drivers that were used in the different clients. Latest versions were used for each architecture as a way to provide optimum performance.

running on the background on the client or a poor cooling system. The former is very unlikely as all running processes were stopped.

5.4 Sysbench

The CPU performance of the x86-64 clients is more than 60x better than that of the Raspberry Pi clients as illustrated on Fig. 4, and the difference comes from the two architectures being built for different types of applications. This difference in performance is discussed in detail in Sect. 6.



Fig. 4. CPU performance of clients

5.5 Cost of Clients

This subsection explores the different prices of the clients that were used. The prices are for a full workstation which includes a monitor (19”) at the price of R2000.00 and a keyboard + mouse combo at R150.00. These are shown in Table 3.

To ensure that prices are fair and markup on the different clients does not influence the comparisons, we considered the prices from vendors that specialise in each client be used. Vendors that specialise in Intel NUC and Mecer clients were not found, two vendors for each were used as an indication of the range of their cost. This does not affect the conclusions that were made from the cost comparisons. The clients that are compared are the Cloudgate and Raspberry Pi 4B. Vendors that specialise in these two clients were found cloudgate.co.za and pishop.co.za respectively.

LibreOffice Calc. For this test the Mecer client finished the execution of the LibreOffice Calc program in an average of 1.4 s. This again represented the fastest execution time compared to all the tested clients. The execution time of the Raspberry Pi 3B+ Rev 1.3 was 22.2 s which represented the worst execution time amongst the bunch. Same as with LibreOffice Writer and Chromium tests most of the recorded time represents the opening of LibreOffice Calc. The actual computation for the x86-64 clients is less than two seconds and less than 5 s for the PIs.

As seen in the results for the LibreOffice Writer and Calc tests (Table 2) there is a 48% and 24.8% difference respectively between the Raspberry Pi 3B+ and Raspberry Pi 4B test results. No conclusive reason was reached for this difference in performance as it is not observed in the Chromium tests and both clients use the same OS. Further investigation needs to be made to identify what the reason for this difference is.

5.3 Stress Test User Interface (s-tui)

Because of the Pi4Bs temperature issue that was experienced at the beginning of implementation stress tests were executed on the clients. This was done to test the performance of the clients under 100% utilisation and see if any temperature issues arise, especially on the Raspberry Pi clients. To mitigate any unintended damage on the Raspberry Pi clients the stress tests were only run for a maximum of 10 min. After the 10 min the utilisation percentage of the Pi4B was 99.3% and that of Pi3B+ was 98.9% as illustrated in Fig. 3.

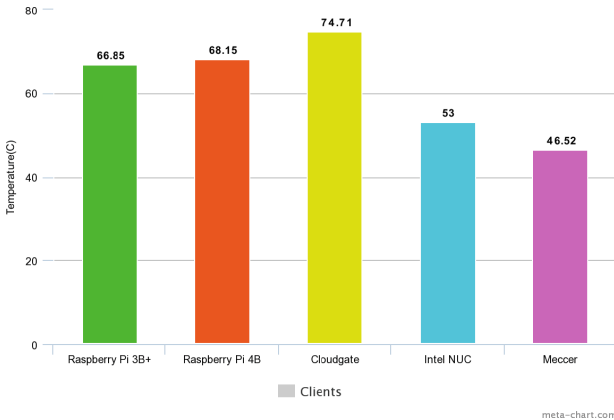


Fig. 3. Stress test results

The performance of the Cloudgate client was not expected with a recorded temperature of 74.71 °C. This represented the highest temperature after 10 min of 100% utilisation. This could be due to other, heavy, processes that might be

5.1 Chromium Tests

The Chromium web browser was chosen because firstly it comes with both installations of Raspberry Pi OS and that of Ubuntu OS, and secondly no drivers were found for Raspberry Pi for Firefox. The FireFox web browser was used by Ingrid Sieborger [14] in her paper to test her LTSP Round Robin cluster to be used in a rural area.

Table 1 shows the comparison of the clients for the Chromium test.

Table 1. Chromium test comparisons

Clients	Intel NUC	Cloudgate	Mecer	Pi 3B+	Pi 4B
Time(s)	13	12.9	11.7	87.5	85.9

As can be seen from Table 1, the clients performed comparatively as expected with the Mecer client and out performing the rest with an execution time of 11.7s. The Raspberry Pi 3B+ had the slowest execution time in the bunch on average taking 87.5s. Some of the challenges that occurred with this test were the following error messages `NoSuchElementException` and `NoSuchWindowException`. Explicit waits were used to wait for the DOM to load all of the required elements, these waits make up the bulk of the noted times.

5.2 LibreOffice Tests

LibreOffice was chosen because it is widely used in the Linux ecosystem and for the fact that it is open source. One of the inhibitors of deployments to rural communities is lack of funds. Therefore the usage of free and open source products is essential to reducing costs.

Table 2. LibreOffice Writer and Calc test comparisons

LibreOffice	Intel NUC	Cloudgate	Mecer	Pi 3B+	Pi 4B
Writer(s)	2.8	2.8	1.6	24.8	12.9
Calc(s)	3.6	3.6	1.4	22.2	16.7

LibreOffice Writer. As was the case with the Chromium browser tests, the Mecer clients out performed all the other clients as expected. However the difference between the execution times of the different x86-64 clients is minuscule, which is 1.3 and 1.1s on the Intel NUC and Cloudgate clients respectively. The Raspberry Pi 3B+ Rev 1.3 had the worst execution time of 12.9s. In the LibreOffice tests the actual computation time was instant in x86-64 clients being under a second and for Raspberry PIs under 6s. Most of the recorded time is made up of the time it takes to open the LibreOffice program.

Chromium Test Script. The Selenium framework was used to create a test script for the chromium browser. Webdrivers had to be manually installed for both systems, but a ChromeDriver was sufficient for the x86-64 clients and for the Raspberry PIs, the Chromium WebDriver was used. A WebDriver is a remote control interface that enables the control of user agents. A WebDriver provides a protocol that can be used in multiple programming languages as a way for programs to remotely control the behaviour of a web browser [18].

The script simulates a user using the Chromium browser, where firstly a timer is started. The web browser is then opened in incognito mode and loads all the required elements. It then opens google.com then on the search bar types “RUConnected¹”. It then opens the Rhodes University’s RUConnected website and inserts login details in the presented login form, in this case my login details are used. It then navigates to the chosen module page and downloads all past exam question papers. After the files are downloaded the timer stops.

LibreOffice Test Script. The LibreOffice test programs were also implemented in Python using an OpenOffice Python package named uno. The Python-UNO package allows the usage of the standard OpenOffice.org API from within a Python script, to develop uno programs.

A timer is started, then the LibreOffice Writer script opens LibreOffice, with an open port to connect to later, using a simple Bash script. On a different thread the script connects to LibreOffice and writes a Lorem Ipsum letter. After the letter is done the timer is stopped and LibreOffice is closed.

Similar to the LibreOffice Writer script, the Calc script starts a timer and opens LibreOffice, with an open port to connect to later, using a simple Bash script. On a different thread the script connects to LibreOffice and opens a spreadsheet it then opens a csv file with student names and marks. The csv file is used to populate two columns of the spreadsheet, the timer is then stopped and LibreOffice closed. The results to these tests and for benchmarking are discussed in the following section.

5 Test Results

To test the performance of the different clients, three Python scripts were created to simulate user interaction as detailed in the previous section. Benchmark testing tools were also used to measure the performance of the CPU of the clients. Because the setting of the project research is in a rural community one of the important factors to be compared was the cost of the hardware, which will be presented at the end of this section. All tests were executed 20 times on each client and the average was used to compare results.

¹ RUConnected is the eLearning platform of Rhodes University.

mentioned in the related work section, this directory contains data for servers. In this case the PC is an LTSP server.

Operations that are not wanted for booting the Raspberry PIs are manually deactivated on the file system. These include:

- **dphyswpafile**: this script manages the PIs swap file. A swap file allows Linux to use some of the disk space as RAM. When the device starts running out of RAM, the swap space is used to swap some data from the RAM to the disk space. This frees up the RAM to do more urgent and possibly important operations. When the RAM finishes with the prioritised operations, it swaps back the content from the disk. This functionality is not needed for network booting, the PIs also don't have a disk, therefore the server will take care of all the disk related requirements.
- **resize2fs_once**: this script resizes the root file system to fill the partition of the available disk. Because the Pi does not have a disk this script may have undesired results.
- **raspi-config**: this is the Raspberry Pi configuration tool, this tool provides the user access to a number of capabilities. These include changing the password of the default/root user, enabling or disabling SSH, screen blanking, etc. Because of the type of setup this computers will be used we don't want learners to have access to such abilities. There will be a dedicated admin user that will have some of these capabilities.

On the other hand, NFS is used to enable the Raspberry Pi Clients to access the file system of the server as they don't have an SD card to locally store data.

4.2 Preparing LTSP Server

LTSP was installed and support for x86-64 clients was configured. The installation did not recognize Raspberry Pi clients, this was solved by using Organisational Unique Identifier(OUI). OUIs refer to the first three octets of a MAC address. The Raspberry Pi Foundation has three OUIs these were all listed on the server to be identified as Raspberry Pi clients.

After the server was able to identify Raspberry Pi clients the boot directory as described in Sect. 4.1 was linked to the TFTP server. This is transferred to all clients that are connected to the network and are identified as Raspberry PIs.

4.3 Testing Scripts

To simulate the usage of programs that are normally used on deployments in rural communities Python scripts were created. The scripts were written for the Chromium browser as it came with the installations of both Raspberry Pi OS and Ubuntu 20.04. Scripts for automating the testing of LibreOffice Writer and Calc were created as they are often used in such deployments.

LibreOffice Python Uno Package. There are three ways to automate LibreOffice, the first one is to control LibreOffice externally, meaning you execute LibreOffice first and run a script that will interact with that instance of LibreOffice. The second one is to run the script internally, meaning before you run open LibreOffice add the script to LibreOffice and execute it. The third way is to add an extension to LibreOffice, this is ideal for adding extra functionality to LibreOffice. The last one is what is used to add extra buttons or other fields.

The first method was chosen for our project, i.e. opening LibreOffice and connecting using a Python script. This method is ideal for the project because the time it takes the clients to open LibreOffice needs to be noted.

Stress Test and Benchmarking. The Stress Test Terminal (s-tui) was used to run stress tests on the client machine. Sysbench was used for evaluating the clients CPU performance.

4 Implementation

The server and each client have their own peripherals, this is for making the environment as close to the real world as possible. This also makes it easy to test all the clients simultaneously, and to configure and debug them.

4.1 Preparing Raspberry Pi Client Image

As illustrated in Fig. 2 the setup contains two Raspberry Pi clients. Because the server is an x86-64 machine this makes the OS that it is running incompatible with the Raspberry PIs as they are ARM based machines. Therefore the OS for the Raspberry PIs had to be manually processed and added to LTSP before these clients can be able to boot from the network.

The first step to achieve this was to choose the appropriate OS for the PIs, the first choice was to use an Ubuntu OS for Raspberry PIs. This was an ideal choice as this would make the user interface for all the clients both x86-64 and PIs the same. This would also make the tests fair for both architectures as they are running the same OS. The OS worked fine on the Raspberry Pi 4 Model B Rev 1.1, however, it was slower compared to when it was running the Raspberry Pi OS. The Ubuntu operating system made the Raspberry Pi 3 Model B Plus Rev 1.3 client unusable as it was constantly crashing. This problem was expected as Ubuntu is designed for computers with a minimum RAM of 4GB and the Raspberry Pi 3 Model B Plus Rev 1.3 that was used only has 2GB of RAM.

After the operating system was chosen the next step was to prepare the ISO file. The Raspbian OS ISO file is manually prepared for network booting, this was done by firstly creating a live loop device. A loop device is like a virtual USB or disk drive. But instead of mapping its data blocks to a physical device it maps it to a regular file in the file system. The boot directory and the rest of the Raspbian OS file system are then copied to the `/srv/ltsp/` directory. As

- **Ethernet Cables:** Five cables were used, one to connect the switch to the internet the other five to connect the clients to the switch.
- **5X Peripherals:** Four sets of peripherals were used to interface with the clients and the last one was for the server.

3.3 LTSP

As mentioned before, LTSP is a software that uses several tools to enable network booting of multiple Local Area Network(LAN) clients from a single LTSP server [9]. A server has a Linux OS and the clients boot from an “identical” copy of that OS. This makes maintaining tens or hundreds of clients as easy as maintaining a single computer. Some of the tools LTSP uses to enable network booting are:

- **iPXE:** which is a leading open source network booting firmware [4]
- **dnsmasq:** which is a tool that mainly provides two services DNS forwarding and DHCP, it is well suited to providing these services to a small network. Dnsmasq supports both static and dynamic DHCP IP-address leasing and TFTP for network booting of diskless machines [2].
- **NFS:** which is a mechanism for storing files on a network. It allows client computers in turn users to access files over a network, these can then be used and as if they were stored locally.
- **Secure Shell Protocol(SSH) or Lightweight Directory Access Protocol(LPDAP)** these are used to authenticate and authorize users.
- **mksquashfs:** this is a Linux tool that is used to compress files and directories. [7].

3.4 Testing Tools

The testing of the clients was carried out in two different phases, firstly testing the execution of commonly used programs. Secondly benchmarking the hardware performance of the clients. The tools that were used to test the common programs are discussed first.

Selenium. Selenium is an open-source software that provides an automated testing framework that is used to test if websites function as expected in various browsers and platforms. The Selenium software is a suite of software systems that cater to different use cases. In this project the Webdriver tool was used to test the performance of the clients in opening, navigating and downloading files using the Chromium Web browser.

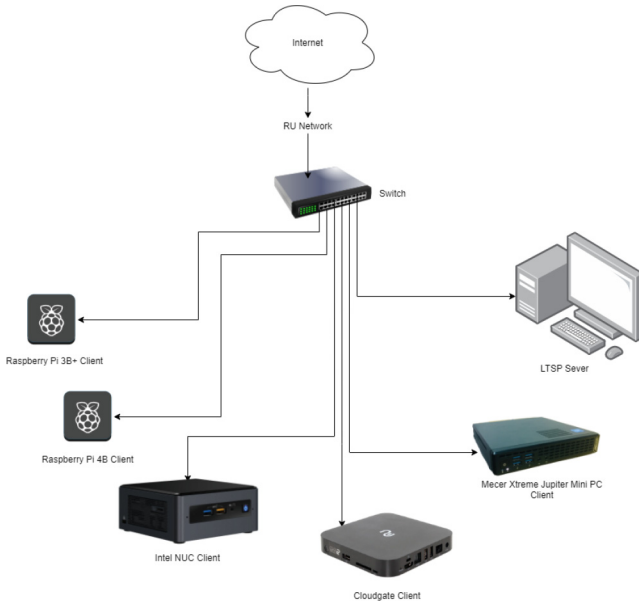


Fig. 2. Hardware setup

- **Server:** The server is running Ubuntu 20.04.1 using Linux kernel version 5.11.0, all the other x86-64 computers are running the same OS. The server has an Intel(R) Core(TM) i7-870 CPU with a base frequency of 2.93GHz. The CPU has 4 cores and 8 threads, the L1d, L1i, L2 and L3 caches are 128KiB, 128KiB, 1MiB and 8MiB respectively.
- **Raspberry Pi 4B:** The Raspberry PIs are running the Raspberry Pi OS and it is using kernel version 5.10.0, both PIs use the same OS. The Pi 4B has an ARM Cortex-A72 64-bit quad core processor with a frequency of 1.5GHz and an on-board 802.11ac WiFi. It supports full gigabit Ethernet(throughput not limited) and has 4GB of RAM.
- **Raspberry Pi 3B+:** This Pi has an ARM Cortex-A53 64-bit quad core processor with a 1.4GHz frequency and also has an on board 802.11ac WiFi. It supports gigabit Ethernet(throughput is limited to ca. 300Mbit/s).
- **Mecor Xtreme Jupiter Mini PC:** This client has an Intel(R) Celeron(R) CPU G3930 at a frequency of 2.90GHz. The CPU has 2 cores per socket, 1 thread per core and 4GiB SODIMM DDR4 Synchronous 2133 MHz system memory.
- **Intel NUC:** The Intel NUC has an Intel(R) Celeron(R) J4005 CPU at a frequency of 2.00GHz. The CPU has 2 cores per socket and 1 thread per core and has 4GiB system memory.
- **Cloudgate:** The Cloudgate client has an Intel(R) Celeron(R) J4115 CPU at a frequency of 1.80GHz. The CPU has 4 cores per socket and one thread per socket. This client also has an 8GiB system memory.

3.1 Methodology

The research we are presenting here was experimental meaning a number of experiments were carried out to determine whether Raspberry PIs can be used as computing infrastructure in schools/community centres. The following experiments were carried out to determine this:

Booting Raspberry Pi clients over a network using LTSP, because this is important when considering the setting where these clients will be deployed. The skills to maintain and manage the clients are limited, and network booting reduces the time the administrator spends on each deployment site as they only have to manage only one device.

Testing the clients was carried out automatically and manually, automatic tests were used to find the computational consistency of the clients. And manual test were used to see how the clients will perform in real time with a human across the screen.

3.2 Hardware

The hardware infrastructure design of the project is illustrated in Fig. 2, the server and all the clients are connected to the Switch with Ethernet cables. The three x86-64 clients were chosen because those are the type of computers that are used in such deployments.

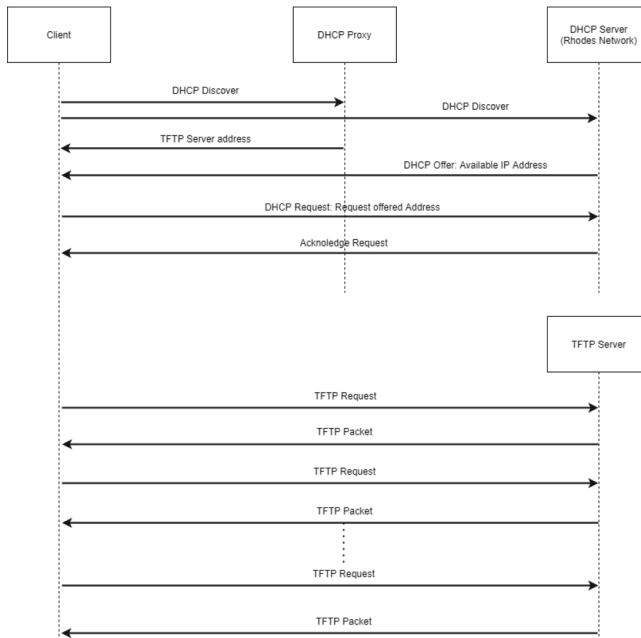


Fig. 1. Client, DHCP and TFTP Server interaction

Figure 1 illustrates the process that happens in the implemented system. The DHCP Proxy and TFTP server are hosted in the same computer as the LTSP server. Because of Rhodes University regulations the clients get their IP addresses from the Rhodes University/Hamilton building networks DHCP server.

2.6 Related Work

There are various open source Linux projects developed, for community use, that utilize network booting with easy to use disk-less computers. The two most popular ones are presented here as they relate to our project.

PiNet. PiNet is an open source and free project that was developed by Andrew Mulholland. It was developed alongside teachers from all over the world. It enables teachers to manage a whole classroom of Raspberry PIs from a single computer making administration and maintenance tasks very easy [5]. This project was phased out in October 2020 and someone else is yet to pick it up.

PiNet is based on LTSP5, and this is a problem because LTSP has been redesigned from scratch and does not support prior versions. For the project to use the later versions of LTSP it has to also be redesigned from scratch. Though this is less important and easy to fix it is worth mentioning, PiNet uses Raspbian Stretch therefore does not support the latest version of Raspberry PIs, Pi4 [1].

PiServer. PiServer was developed by the Raspberry Pi team, similar to PiNet it enables users to manage Raspberry PIs from a single x86-based server(central computer) running the x86 version of Raspbian OS [3]. PiServer enables users to network boot generic Pi clusters and is targeted to a larger audience than only schools, this the main target audience for PiNet.

PiServer does not use LTSP to boot it's clients, it does however use the same tools to enable network booting. These include DHCP for leasing IP addresses to clients, Lightweight Directory Access Protocol(LDAP) for authentication and authorization and NFS for sharing the servers file system.

Both of these projects boot over a network and this is what enables them to manage the Raspberry PIs(clients) from a single personal computer(server). The clients share a single file system and users can login any of the client devices and find their data there. However both projects only support network booting Raspberry Pi clients and have user interfaces for configuration. This prevents users from connecting other diskless computers they might have and abstracts the operations that happen under the hood making it tricky to customize. The project we are presenting in this paper solves these problems, some of the methods that went into the designs and testing of the system are discussed next.

3 Methodology and Design

This chapter looks into the design of the system we developed. The methods that were used to carry out the research, test the clients and programs that were used in the tests are discussed in this section.

2.3 Raspberry Pi

The Raspberry Pi is the name of a series of inexpensive credit-card sized computers that can be used with common peripheral devices that are normally used with desktop computers like a monitor, standard keyboard and mouse. It can do everything that you would expect a desktop computer to do from browsing the Internet to playing video games [12].

The latest version of the Raspberry Pi is the fourth-generation Raspberry Pi computer series, it has a 1.5GHz clock speed processor, with RAM that is up to 8GB, a gigabit Ethernet adapter, 2.4GHz and 5.0GHz IEEE 802.11ac wireless, Bluetooth 5.0, 2*USB 2 and 2*USB 3 ports [10]. These specifications make the Raspberry Pi ideal for creating a cost effective computing infrastructure, which can also boot over a network.

2.4 Booting

Booting refers to a sequence of events/operations that happen before the computer is ready to display content. The events are called a boot sequence, and each computer has a boot sequence. The Linux boot process is completed by copying the kernel binary image to the secondary storage on the RAM disk, and then loading it into the main memory, and finally running the kernel image. [8]. This involves a number of programs working together to achieve the final results.

These programs include the Master Boot Loader(MBR), where its main function is to identify where the OS is located and load it into the RAM. It also contains information about the GRUB. GRUB stands for GRand Unified Boot Loader, it is the most common boot loader for Linux systems. The GRUB splash screen is typically the first thing that appears when a computer is turned on, this can be used to select a kernel image. The GRUB file also starts the `init` program, this is always the first program to be run. The kernel then loads a temporary root file system using `initrd` which stands for Initial Ram Disk. At this point the system executes run level programs, afterwards the real root file system can be mounted.

2.5 Network Booting

Network booting is similar to normal booting and the key difference is that the computer gets the image from a server in the network. This enables a computer to load an operating system directly from the network without any attached local storage devices like a SSD, HDD, USB, SD card, etc. This is made possible with the help of technologies like the iPXE. Preboot Execution Environment(PXE) is a client-server interface that allows computers in a network to be booted from the server before deploying the obtained PC image, and iPXE is an open-source implementation of PXE. The other technologies that are typically used are the Dynamic Host Configuration Protocol(DHCP) server for assigning IP addresses. The Trivial File Transfer Protocol(TFTP) server to store and transfer the boot program when requested, and the NFS to enable client computers to access files in the server computer.

Fat/Thick Clients. Thick clients on the other hand do the bulk of their processing locally. Unlike thin clients the type of communication that they have with the server has to do with storage. This includes requesting or updating archival information on the server. Thick clients have better processing power than thin clients, because all processing is done locally they tend to be a little bit faster too. One of the major advantages of thick clients is the fact that they can be used independently, without the server and perform exactly the same way they did when connected to the server. This is one of the key reasons thick clients are used in the implementation of the project we are presenting in this paper. That is, to be able to use them independently in a case where the server fails.

2.2 Linux File System

File systems are designed to structure the storage of non-volatile information/data, this is done through providing a name space and metadata structure [13]. Desktop computers need to be able to store data on a hard disk or a similar type of memory, like a USB drive. Firstly, it is non-volatile meaning it does not require a constant power in order to retain the stored information. Unlike Random Access Memory(RAM) in which after power is off the stored content is deleted, in disk memory data does not get deleted. Secondly, disk storage is inexpensive compared to RAM memory.

A file system also needs an Application Programming Interface(API) that enables either the system or users to manipulate its objects like files or directories with restricted access. Some of the most popular tasks include creating or deleting a file or directory, and moving or copying them [13]. The API does this by using algorithms which efficiently determine where files are stored and how to get them. Because it is free and open source, we opted to use the Linux File System.

Linux Directory Structure. The directories in Linux are structured in a tree like hierarchy with root at the top of the tree.

Some of the noteworthy directories in the Linux file system are listed and their functions are explained below:

- **root:** is the home directory of the superuser this is typically the administrator of the system. This user has unlimited privileges.
- **boot:** This is where the files that are needed to startup the computer are stored. These includes the grub, bootloader and kernel.
- **mnt:** This is a temporary mount-point for regular file systems.
- **src:** This directory contains data for servers. For example this is where HTML(/src/http) or TFTP(/srv/tftp or /srv/www/) files are stored when a web or TFTP server from a Linux system.

The family of Linux Operating System(OS) are used in a variety of settings, one of which is in mini computers. Raspberry Pi OS which was created by the Raspberry Pi Foundation to be used on their Raspberry Pi mini computers is one of such applications.

most likely worsen as it will get harder for them to participate in the digitized environment.

The main objective of this study is to implement the support of Raspberry Pi clients on LTSP which can be used as workstations in either rural schools or community centres. When providing computing infrastructure to marginalised communities there are problems that one comes across, one of which being lack of funds [17]. Therefore it is always better to opt for open-source and affordable software and hardware alternatives. That is why Raspberry PIs and the LTSP were used in this project. Before starting the research project a study of the background and other projects that are related to it was undertaken. The next section discusses the background and related works.

2 Background and Related Work

In this section all the necessary concepts that were required to successfully achieve the objectives of the research project are reviewed and briefly discussed. The first topic to get discussed is the Client/Server model:

2.1 Client/Server Model

The client-server model is a distributed model in which the server provides a service, resource and computational power etc. to the clients. The client relies on sending requests to the server in order to gain access to services that it requires. Depending on the type of client it is, the client depends on the server to do some operations. Two of the most popular types of clients for the client/server model are thin and fat/thick clients. There are numerous similarities between thin and thick clients in both cases the client sends a request and receives responses from the server. The server, in both cases, acts as a middleman.

Because of its easiness and advantage in protecting data with access control and security policies, the client/server model has been well adopted and is used in diverse settings. Some of the most popular examples that this model is used include: Network File System (NFS), File Transfer Protocol (FTP), and Hypertext Transfer Protocol (HTTP) [6]. As mentioned above, clients can be either thin or thick and the below sections explain their difference:

Thin Client. A thin client is designed such that all, with the exception of controlling peripherals, processing is done on the server. The client functions as a stripped down terminal to the server and it requires constant communication with the server to do all computation [11]. These types of clients provide a seamless experience for the user and enable them to interact with the client as though they are working on the server computer. Thin clients do not have a disk, are inexpensive and they are typically unusable/unreliable without the server.



Investigating Cost-Effective Computing Infrastructure for Schools/Community Centres Using Raspberry PIs

Live Tembiso, Zelalem Shibeshi^(✉), and Alfredo Terzoli

Department of Computer Science, Rhodes University, Grahamstown, South Africa
{z.shibeshi,a.terzoli}@ru.ac.za

Abstract. Computing infrastructure plays a significant role in our lives. This has been made even more apparent by the Corona virus pandemic. However a large number of people still do not have access to any form of end user computing infrastructure. This paper looks at the reliability and feasibility of using Raspberry PIs as a form of computing infrastructure for communities that would otherwise not have access to computers. The Raspberry PIs are compared to other mini computers that use x86-64 processors. In the implementation of the project an LTSP (Linux Terminal Server Project) network that supports both x86-64 and Raspberry Pi clients was created in order to test both systems in the same network. LTSP makes it easy to boot LAN clients from a single image. Because LTSP does not support Raspberry Pi clients this support was added in manually. Three programs that are commonly used in deployments of this nature were tested, LibreOffice Writer, LibreOffice Calc and the Chromium browser, using custom Python scripts. To test and compare the performance of the different clients sysbench and s-tui benchmark tests were used to benchmark the CPU and I/O performance of the clients. The results show that the raw computational power of the x86-64 clients is 60x better than that of the Raspberry Pi clients. The usage of the different types of clients is comparable with the x86-64 clients being 2.5-6x faster than the Raspberry Pi clients.

Keywords: LTSP · Raspberry Pi · Linux file system · ICT4D

1 Introduction

Like many in developing nations, a large number of South Africans still don't have access to the Internet due to various reasons [15]. This is a difficult truth to come to terms with especially as we are living in the information age and connectivity is an essential component of access to information. Some of the key reasons for the lack of connectivity are cost of computing infrastructure and scarcity of skills [16]. If this problem persists to hinder marginalized communities, which are mainly poor and have below-average education, their situations will

wireless network virtualization depends on specific access technologies, and the wireless network contains many more access technologies compared to wired network virtualization and each access technology has its own unique characteristics, which makes convergence, sharing and abstraction difficult to achieve [11, 17]. How to define a VM migration method taking into account the constraints in a wireless network environment?

References

1. Alain, F.: Qu'est-ce-que la virtualisation? (2019). <https://www.piloter.org/techno/support/virtualisation.htm>. Accessed 13 July 2019
2. Benbrahim, S.E.: Migrations en temps réel des machines virtuelles interdépendantes. Ph.D. thesis, École Polytechnique de Montréal (2016)
3. Benedicte, B.: Migration informatique: le guide pour réussir (2021). <https://blog.hubspot.fr/marketing/migration-informatique>. Mis en ligne le 29 Novembre 2021. Accessed 11 Dec 2021
4. Chowdhury, N.M.K., Boutaba, R.: A survey of network virtualization. *Comput. Netw.* **54**(5), 862–876 (2010)
5. Quelle est la configuration requise pour windows xp? (2022). <https://frameboxxindore.com/fr/windows/what-are-the-minimum-requirements-for-windows-xp.html>. Accessed 12 May 2022
6. La migration informatique, qu'est-ce que c'est? (2021). <https://www.redhat.com/fr/topics/automation/what-is-it-migration>. Mis en ligne le 04 Février 2021. Accessed 11 Dec 2021
7. Qu'entend-on par migration informatique? (2021). <https://www.groupe-sl.com/2021/08/02/migration-informatique>. Mis en ligne le 02 Aout 2021. Accessed 11 Dec 2021
8. Jacques, L.: Introduction aux systèmes informatiques: architectures, composants, mise en œuvre, pp. 1–2. Dunod (2017)
9. Keshavamurthy, U., Guruprasad, H.: VM migration: a survey. *Global J. Eng. Sci. Res.* (2015)
10. Kherbache, V.: Ordonnancement des migrations à chaud de machines virtuelles. Ph.D. thesis, Université Côte d'Azur (2016)
11. Liang, C., Yu, F.R.: Wireless network virtualization: a survey, some research issues and challenges. *IEEE Commun. Surv. Tutor.* **17**(1), 358–380 (2014)
12. Passante, B.: Quelle est la difference entre la bande wifi 2.4 ghz et la 5 ghz ? (2022). <https://sogetel.com/aide/internet/quelle-est-la-difference-entre-la-bande-wi-fi-2-4-ghz-et-la-5-ghz>. Accessed 12 May 2022
13. Pham, T.S.: Autonomous management of quality of service in virtual networks. Ph.D. thesis, Université de Technologie de Compiègne (2014)
14. Popek, G.J., Goldberg, R.P.: Formal requirements for virtualizable third generation architectures. *Commun. ACM* **17**(7), 412–421 (1974)
15. Reseau, I.: Qu'est-ce-qu'un réseau informatique ? (2019). <https://primabord.eduscol.education.fr/qu-est-ce-qu-un-reseau-informatique>. Mis en ligne le 07 juin 2016. Accessed 13 July 2019
16. Venkatesha, S., Sadhu, S., Kintali, S.: Survey of virtual machine migration techniques. *Memory* (2009)
17. Wang, X., Krishnamurthy, P., Tipper, D.: Wireless network virtualization. In: 2013 International Conference on Computing, Networking and Communications (ICNC), pp. 818–822. IEEE (2013)

$$v \left\{ \begin{array}{l} c_v : 64MB \\ d_v : 1Go = 1000 MB \\ P_v : 233 MHz \\ k_v : 0no - devices \\ l_v : 10software \\ r_v : 10(LAN) \\ b_v : 0 \end{array} \right.$$

$$c_v + d_v = 64 + 1000 = 1064$$

$$p_v + k_v + l_v + r_v + b_v = 233 + 0 + 10 + 10 + 0 = 273$$

We have: $c_v + d_v > p_v + k_v + l_v + r_v + b_v$. What's more, today's computers are v e r y powerful and their RAM and hard disk capacities can reach the order of gigabytes (for RAM) or terabytes (for hard disks).

8 Conclusion and Perspectives

This paper looks at the migration of virtual machines, which is fast becoming a must in data centres, as server hosting capacities are becoming more and more elevated. We began our discussion with an overview of IT migration. Here, we presented a brief overview of the IT migration environment and its motivations. We then discussed the notion of machine migration. In particular, we have formally defined the notions of machine and machine migration. We then discussed the concept of virtual machine migration. Specifically, we have defined the formal characterisation of a virtual machine and the formal language formulation of the virtual machine migration process. We then showed that the virtual machine migration problem is an NP-hard problem. Finally, we proposed an optimal algorithm for solving this problem, and discussed our algorithm.

In summary, the migration of virtual machines is used on a daily basis to improve application performance, reduce power consumption and increase the efficiency of the system. This is a wide-ranging field, with a number of research questions and challenges coming to the fore. It is quite broad and a number of research questions and challenges are coming to the fore. Several improvements can be made to this work in future projects.

A first perspective to this work is to find a mathematical model allowing to evaluate the total migration time of a VM in accordance with our proposed migration approach. Then compare our migration approach with those in the literature.

Another perspective would be to extend our method to the simultaneous (grouped) migration of parallel and interdependent virtual machines [2].

Migrating a VM from one physical host to another in a multi-hop network must take into account different path parameters such as: bandwidth, number of nodes and distance. How can we implement a migration method that defines the optimal migration path in order to reduce the transfer time for migration packets? Virtualisation, irrespective of wired or wireless networks, can be considered as a process dividing the entire network system [11,17]. However, the distinctive properties of the wireless environment, in terms of time-varying channels, attenuation, mobility, broadcast, etc., make the problem more complicated. In addition,

$D(d_v, d_2) = D(1000, 4000) = 1$ (because $1000 < 4000$)

We have: $\eta(v) < \eta(m_1)$

The vector associated with the virtual machine has the coordinates:

$$vm = (64, 1000, 233, 0, 4, 100, 0, 128, 2000, 512, 2, 4, 10, 0)$$

Let vm' be the vector representing the data of the virtual machine vm running at a given time. it can have for coordinates:

$$vm' = (32, 512, 233, 0, 2, 100, 0, 128, 2000, 512, 2, 4, 10, 0)$$

The total weight of bag m_2 is:

$$W = 256 + 4000 + 1000 + 2 + 8 + 10 + 0 = 5276$$

Given that the various migration conditions are verified, we can carry out the migration of the vm through the algorithm by passing the parameters as follows: $MigrationVM(32, 512, 233, 0, 2, 100, 0, 128, 2000, 512, 2, 4, 10, 0)$.

7 Discussion

In practice, the migration of a running virtual machine from a physical machine to another can be summarized in the following steps:

- (1) saving the context in which the VM is run;
- (2) creation of the VM on the target machine;
- (3) copying VM data from the source machine to the destination machine;
- (4) destruction of the VM in the source machine;
- (5) restoring the execution context of the VM.

The data transfer time of a running VM relies mainly on the data of the vector v , since it is the vector that effectively represents the data transfer time of a running VM and information about the hosting physical host (vector m) need not be moved, since the VM will be hosted on the destination host with a similar architecture.

We can further reduce the migration data for the vector vm to 02 These are the main elements: **the amount of RAM used by the VM, the amount of hard disk used by the VM**, because these two parameters have a very large amount of data to transfer and are constantly modified during the VM's operation, which is not the case for the others: $c_v + d_v > p_v + k_v + l_v + r_v + b_v$. In addition, when the VM's execution context is saved, information such as the virtual processor frequency, the number of devices used by the VM, the number of software applications, the number of network connections and the virtual bandwidth is saved and can be configured on the target host responsible for hosting the migrating VM.

Proof. Let's take the minimum characteristics of a personal computer defined in Sect. 3 as being the values of v when the VM is running. Taking randomly the other values of v we have:

Algorithm 2: MigrationSend

Enter: $V[]$ an array of real numbers, of length 14, such that
 $V = (c, d, p, k, l, r, b, c', d', p', k', l', r', b')$.

Output: A stack of reals of length 14.

```

1 start
2    $V'[]$  A stack of reals of length 14;
3   for  $i$  from 7 to 1 do
4      $Stack(V', V[i]);$ 
5      $Stack(V', V[i + 7]);$ 
6   return ( $V'$ );
```

Algorithm 3: MigrationReceive

Enter: $P[]$ a stack of real numbers of length 14.

Output: An array of real numbers, length 14.

```

1 start
2    $W[]$  an array of real numbers of length 14;
3   for  $i$  from 1 to 7 do
4      $W[i] = Unstack(P);$ 
5      $W[i + 7] = Unstack(P);$ 
6   return ( $W$ );
```

NB: The $Stack()$ and $Unstack()$ functions used in these algorithms are operations in the algorithmic data structure known as the “stack”.

6.2 Illustration De Notre Solution

To illustrate our approach, let's consider the example of the migration of a VM below:

Let m_1 and m_2 be two physical machines with the following characteristics:

$$\begin{array}{ll}
 m_1 \left\{ \begin{array}{l} c_1 : 128 \text{ MB} \\ d_1 : 2 \text{ Go} = 2000 \text{ MB} \\ P_1 : 512 \text{ MHz} \\ k_1 : 2 \text{ devices} \\ l_1 : 10 \text{ softwares} \\ r_1 : 10(\text{LAN}) \\ b_1 : 0 \end{array} \right. &
 m_2 \left\{ \begin{array}{l} c_2 : 256 \text{ MB} \\ d_2 : 4 \text{ Go} = 4000 \text{ MB} \\ P_2 : 1 \text{ GHz} = 1000 \text{ MHz} \\ k_2 : 2 \text{ devices} \\ l_1 : 8 \text{ software} \\ r_2 : 10(\text{LAN}) \\ b_2 : 0 \end{array} \right.
 \end{array}$$

Let v be a virtual machine of the machine m_1 with the following characteristics:

$$v \left\{ \begin{array}{l} c_v : 64 \text{ MB} \\ d_v : 1 \text{ Go} = 1000 \text{ MB} \\ P_v : 233 \text{ MHz} \\ k_v : 0(\text{no} - \text{devices}) \\ l_v : 4 \text{ softwares} \\ r_v : 100(\text{WAN}) \\ b_v : 0 \end{array} \right.$$

this transfer is: **the algorithm by priority** because the data to be transferred during migration does not require an order of priority, for example data from the central memory is the first data to be transferred, after data from the hard disk then data from the hard disk.

6.1 Description of the Algorithm of Our Solution

To lay the foundations for this part of our work, we make the following assumptions:

- m_1 and m_2 are personal computers;
- m_1 and m_2 belong to a local unicast network;
- m_1 and m_2 are switched on and meet the conditions described in Sect. 2.1;
- the virtual machine vm is created;
- the virtual machine vm is running;
- priority is assigned to vm components during migration as follows: 1 for c_v , 2 for c_1 , 3 for d_v , 4 for d_1 , 5 for p_v , 6 for p_1 , 7 for k_v , 8 for k_1 , 9 for l_v , 10 for l_1 , 11 for r_v , 12 for r_1 , 13 for b_v , 14 for b_1 .

The vm virtual machine will be migrated as follows:

1. for each component of the vm vector currently running, their value must be stacked in decreasing order of priority, i.e. from number 14 to number 1.
2. the migration traffic containing the stack is sent to the destination machine m_2 (see Algorithm 2);
3. Once the stack arrives in the m_2 machine, it will be unstacked according to the defined priority order (see Algorithm 3).

The Algorithm 1 of complexity $O(7)$ is the main algorithm used to perform the migration, the Algorithm 2 of complexity $O(7)$ is used to send the migration packet and the Algorithm 3 of complexity $O(7)$ is used to receive the migration packet.

Algorithm 1: Migration algorithm (MigrationVM)

Enter: $T[]$ an array of reals, of length 14, representing the migrated vm , such that $T = (c_v, d_v, p_v, k_v, l_v, r_v, b_v, c_1, d_1, p_1, k_1, l_1, r_1, b_1)$.

Output: An array of reals, of length 14, representing the vm migrated.

```

1 start
2    $V[]$  a stack of reals of length 14;
3    $T'[]$  an array of reals of length 14;
4    $V = migrationSend(T)$ ;
5    $T' = migrationReceive(V)$ ;
6   return ( $T'$ );

```

We want to migrate the virtual machine vm from a machine m_1 to m_2 taking into account the execution parameters of each machine and that of the virtual machine. The problem is to find the 14-tuplets $(c_v, d_v, p_v, k_v, l_v, r_v, b_v, c_1, d_1, p_1, k_1, l_1, r_1, b_1)$ such that the following conditions are met:

- $D(d_v, d_2) = 1$;
- $\eta(v) \leq \eta(m_1)$;
- $V(vm) = vm'$ whwere $vm' = (v, m_2)$.

This problem can be reduced to the problem of the rucksack where:

- $vm = (c_v, d_v, p_v, k_v, l_v, r_v, b_v, c_1, d_1, p_1, k_1, l_1, r_1, b_1)$ is the vector of objects to be transported;
- m_2 the bag where we are going to place the objects;
- the total weight of the bag is: $W = c_2 + d_2 + p_2 + k_2 + l_2 + r_2 + b_2$;
- the content of each component of the vector vm represents the weight of each object to be transported;
- the object values are the data for the VM currently running;
- the rucksack m_2 must be filled in such a way that the transfer time of the object values is minimal while respecting the weight constraint. In other words: $\sum(a_i) \leq W$, where the $a_i, i \in [1; 14]$ are the weight of each object to be transported.

Conclusion: The VM migration problem is an NP-hard problem because the rucksack problem is an NP-hard problem.

6 Virtual Machine Migration Approach

Let m_1 and m_2 two physical machines located respectively in execution environments A and B such that : $m_1 = (c_1, d_1, p_1, k_1, l_1, r_1, b_1)$ and $m_2 = (c_2, d_2, p_2, k_2, l_2, r_2, b_2)$, with $(c_i, d_i, p_i, k_i, l_i, r_i, b_i)_{1 \leq i \leq 2} \in \mathbb{R}_+^7$ and c_i represents the capacity of the central memory, d_i the capacity of the hard disk, p_i frequency processors, k_i the number of devices, l_i the number of software, r_i the type of network to which the machine belongs, b_i the capacity of the bandwidth.

Let vm be a virtual machine of machine m such that: $vm = (v, m_1)$, where $v = (c_v, d_v, p_v, k_v, l_v, r_v, b_v)$ represents the vector associated with the states of the virtual main memory, the capacity of the virtual hard disk, the frequency of the virtual processors, the number of devices connected to the VM, the number of software applications in the VM, the number of network connections in the VM and the virtual bandwidth.

Let $V : E \rightarrow F$, be the virtual machine migration function such that V is subjective and $E, F \in \mathbb{R}_+^{14}$ respectively represent the source environment and the VM's target environment.

We want to migrate the virtual machine vm from a machine m running in the physical machine m_1 is minimal. The optimal algorithm to carry out

5 Virtual Machine Migration Problem

5.1 Description of the Problem

In a network virtualization environment, the migration of equipment must be managed efficiently. This migration is not limited to just the migration that of virtual machines from one virtual network to another, but also for the transfer of virtual routers [4].

When a host or a physical router encounters a breakdown, has a problem of maintenance, computing power, energy saving, it is necessary to start the migration of the VMs if ex running in this host in order to guarantee a good quality of service to users. The major problem of the migration of VMs resides in the need to stop the VM throughout the duration of its move. The applications and services that run on VMs are generally critical and it is therefore difficult to consider stopping them even for a short period [10]. The main issue in migration VMs is therefore to minimize this downtime in order to give the user as much illusion as possible that the machine has never stopped. Thus, transferring the migration traffic of a VM from one failing host to another while ensuring minimal downtime of that VM is an issue that requires attention [10].

Furthermore, for VM migration to be possible, the target machine must have sufficient memory space to host the migrating VM. Given that a VM can be connected to several networks then the migration must keep its various connections active in order to ensure continuity of service.

In addition, the migration of a VM from one physical host to another in a multi-hop network must take into account the various path parameters, namely: bandwidth, number of nodes and distance. In such a case, what will be the optimal migration path allowing us to reduce the transfer time of migration packets?

The research problem is the optimization of the migration time of the virtual machine in the networks.

5.2 Mathematical Formulation

Given m_1 and m_2 two physical machines located respectively in execution environments A and B such that: $m_1 = (c_1, d_1, p_1, k_1, l_1, r_1, b_1)$ and $m_2 = (c_2, d_2, p_2, k_2, l_2, r_2, b_2)$, with $(c_i, d_i, p_i, k_i, l_i, r_i, b_i)_{1 \leq i \leq 2} \in \mathbb{R}_+^7$ and c_i represents the capacity of the central memory, d_i the capacity of the hard disk, p_i frequency processors, k_i the number of devices, l_i the number of software, r_i the type of network to which the machine belongs, b_i the capacity of the bandwidth.

Let vm be a virtual machine of machine m_1 such that: $vm = (v, m_1)$, where $v = (c_v, d_v, p_v, k_v, l_v, r_v, b_v)$ represents the vector associated with the states of the virtual main memory, the capacity of the virtual hard disk, the frequency of the virtual processors, the number of devices connected to the VM, the number of software applications in the VM, the number of network connections in the VM and the virtual bandwidth.

Let $V : E \rightarrow F$, be the virtual machine migration function such that: $E, F \in \mathbb{R}_+^{14}$, respectively represent the source environment and the VM's target environment.

A VMs operating system environment can be migrated from one physical machine to another, provided that there are sufficient similarities between the system architectures of these host machines [16].

Given m_1 and m_2 two physical machines such that: $m_1 = (c_1, d_1, p_1, k_1, l_1, r_1, b_1)$ and $m_2 = (c_2, d_2, p_2, k_2, l_2, r_2, b_2)$. We can formally define the migration of a VM from an initial environment A to a final environment B by a function $V : A \rightarrow B$. A and B sont des espaces vectoriels avec $A = \mathbb{R}_+^{14}$ and $B = \mathbb{R}_+^{14}$.

We make the following assumptions:

1. $m_1, m_2 \in \mathbb{R}_+^7$ and verify the assumptions made in Sect. 3.1;
2. $c_1 \leq c_2, d_1 \leq d_2, p_1 \leq p_2$ and $b_1 = b_2$;
3. $D(d_v, d_2) = 1$, where D is a Boolean function used to say whether the hard disk d_2 contains enough space to host the VM (1=Yes et 0=No);
4. $\eta(v) \leq \eta(m)$;
5. V a subjective function;
6. $\forall vm_1 \in A, V(vm_1) = vm_2$ avec $vm_2 \in B$;
7. $V(vm_1) = V(v, m_1) = (V(v), V(m_1)) = (v, m_2) = vm_2$

Let's show that V is a subjective function:

Given $y \in B$, let's find $x = (v_1, m_1) \in A$ such that: $y = V(x)$, with $v_1 = (c_1, d_1, p_1, k_1, l_1, r_1, b_1) \in \mathbb{R}_+^7$ and $m_1 = (c_0, d_0, p_0, k_0, l_0, r_0, b_0) \in \mathbb{R}_+^7$. $y \in B \Rightarrow y = (v, m)$, where $v = (c_v, d_v, p_v, k_v, l_v, r_v, b_v) \in \mathbb{R}_+^7$ and $m = (c, d, p, k, l, r, b) \in \mathbb{R}_+^7$. $\exists c_2, d_2, p_2, k_2, l_2, r_2, b_2$ such that: $c_2 Rc, d_2 Rd, p_2 Rp, k_2 Rk, l_2 Rl, r_2 Rr$ and $b_2 Rb$, where R is the order relation in $\mathbb{R}_+^*$.

$\exists c_3, d_3, p_3, k_3, l_3, r_3, b_3$ such that: $c_3 Rc_v, d_3 Rd_v, p_3 Rp_v, k_3 Rk_v, l_3 Rl_v, r_3 Rr_v$ and $b_3 Rb_v$, où R is the order relation in $\mathbb{R}_+^*$.

As $c \leq c_0$ we can take $c_0 = c$ because $c \leq c$.

As $d \leq d_0$ we can take $d_0 = d$ because $d \leq d$.

As $p \leq p_0$ we can take $p_0 = p$ because $p \leq p$.

As $b \leq b_0$ we can take $b_0 = b$ because $b \leq b$.

We can also take $p_0 = p_2, k_0 = k_2, l_0 = l_2, c_1 = c_3, d_1 = d_3, p_1 = p_3, k_1 = k_3, l_1 = l_3, r_1 = r_3, b_1 = b_3$.

So for $x = (v_1, m_1)$ with $v_1 = (c_3, d_3, p_3, k_3, l_3, r_3, b_3)$ and $m_1 = (c, d, p, k_2, l_2, r_2, b)$, we have bel and well $V(x) = y$.

Property 4. We talk about virtual wire migration when: $r_1 = 1$ or $r_1 = 10$ or $r_1 = 11$ or $r_1 = 100$.

Property 5. We talk about virtual wireless migration when: $r_1 = 101$ or $r_1 = 110$ or $r_1 = 111$ or $r_1 = 1000$.

Popek et Goldberg [14] define a virtual machine as an environment created by a virtual machine monitor (VMM) or hypervisor. A VMM is the software layer providing virtualization¹. It virtualizes all the resources of a physical machine, thus defining and supporting the running of several virtual machines. There are several VMMs, including Vmware workstation, Xen and virtual box [13]. An essential feature of a virtual machine is that the software running in it is limited to the resources and abstractions provided by the VM [16].

4.2 Formal Definition of a Virtual Machine

Let m a machine such that $m = (c, d, p, k, l, r, b)$, where c, d, p, k, l, r, b represent respectively: the capacity of the main memory, the capacity of the hard disk, the frequency of the processors, the number of connected devices, the number of software programs, the type of network and the capacity of the bandwidth.

We assume that a VM belongs to a physical machine m and is defined by a state vector: $vm = (v, m)$, with $v = (c_v, d_v, p_v, k_v, l_v, r_v, b_v)$, representing the vector associated with the states of the virtual main memory, the capacity of the virtual hard disk, the frequency of the virtual processors, the number of devices connected to the VM, the number of software applications in the VM, the number of VM software, the number of VM network connections and the virtual bandwidth if the VM is a virtual router.

We make the following assumptions:

1. $vm \neq 0$;
2. we will consider that VM is active and that it contains data currently being executed;
3. $c_v, d_v, p_v, k_v, l_v, r_v, b_v$ verify the assumptions of Sect. 3.1;
4. $c_v < d_v$
5. $d_v < d$;
6. $c_v < c$;
7. $p_v < p$;
8. $k_v \leq k$;
9. the norm of the vector vm is defined as follows:

$$\eta(vm) = \sqrt{c_v^2 + d_v^2 + p_v^2 + k_v^2 + l_v^2 + r_v^2 + b_v^2 + \eta(m)^2}.$$

4.3 Migration of Virtual Machines

Virtual machine migration is the ability to move the operating system instance from one physical machine to another [16].

¹ The virtualization is the set of techniques making it possible to dissociate the characteristics physical characteristics of a hardware or software system user-oriented applications [1].

3.3 Machine Migrations

Given that a machine is a component of a computer system, we can define machine migration as the movement of a machine from one operating environment to another.

We can formally define migration as a function $M : A \rightarrow B$, where A denotes the initial environment and B the final environment. A and B are vector spaces.

We make the following assumptions:

1. $A, B \in \mathbb{R}_+^7$;
2. we will consider m machine that is active;
3. M est is a subjective function;
4. $\forall m \in A, M(m) = m'$ with $m' \in B$;
5. $M(m) = M(c, d, p, k, l, r, b) = (M(c), M(d), M(p), M(k), M(l), M(r), M(b))$
 $= (c', d', p', k', l', r', b') = m'$;
6. $d \leq d'$.

Let's show that M is a subjective function:

Given $y \in B$, let's find $x = (c_1, d_1, p_1, k_1, l_1, r_1, b_1) \in B$ such that: $y = M(x)$, with $c_1, d_1, p_1, k_1, l_1, r_1, b_1 \in \mathbb{R}_+^*$.

$y \in B \Rightarrow y = (c, d, p, k, l, r, b)$, with $c, d, p, k, l, r, b \in \mathbb{R}_+^*$ $\exists c_0, d_0, p_0, k_0, l_0, r_0, b_0$ such that : $c_0 R c, d_0 R d, p_0 R p, k_0 R k, l_0 R l, r_0 R r$ and $b_0 R b$, where R is the order relation in $\mathbb{R}_+^*$.

We can take $c_1 = c_0, p_1 = p_0, k_1 = k_0, l_1 = l_0, r_1 = r_0, b_1 = b_0$.

Furthermore, since $d \leq d_1$ we can take $d_1 = d$ because $d \leq d$.

So for $x = (c_0, d, p_0, k_0, l_0, r_0, b_0)$, so we get $M(x) = y$.

Property 1. We talk about software migration when the following condition is met: $c' = c$ and $d' = d$ and $p' = p$ and $k' = k$ and $r' = r$ and $l' \neq l$ and $b' \neq 0$.

Property 2. We talk about data migration when one of the following cases is true:

- $c' \neq c$ and $d' \neq d$ and $p' \neq p$ and $k' \neq k$ and $r' \neq r$ and $l' = l$ and $b' \neq 0$.
- $c' \neq c$ and $d' \neq d$ and $p' \neq p$ and $k' \neq k$ and $r' = 0$ and $l' = l$ et $b' \neq 0$.
- $c' \neq c$ and $d' \neq d$ and $p' \neq p$ and $k' \neq k$ and $r' \neq r$ and $l' \neq l$ and $b' \neq 0$.
- $c' \neq c$ and $d' \neq d$ and $p' \neq p$ and $k' \neq k$ and $r' \neq 0$ and $l' \neq l$ and $b' \neq 0$.

Property 3. We talk about migration to the cloud when owner 1 or owner 2 is checked.

4 Virtual Machine Migration Concept

4.1 Notion of Virtual Machine

A virtual machine (VM) is a software implementation of a physical machine that runs programs like a real machine [16].

5. the possible values of r are: $r = 0$ the machine is not connected to any network, $r = 1$ the network type is PAN, $r = 10$ the network type is LAN, $r = 11$ the network type is MAN, $r = 100$ the network type is WAN, $r = 101$ the network type is WPAN, $r = 110$ the network type is WLAN, $r = 111$ the network type is WMAN, $r = 1000$ the network type is WWAN.
6. $p, b \in \mathbb{R}_+$ with $p \geq 233 \text{ MHz}$;
7. the possible values of b are [12]: $2,4 \text{ GHz}$ and 5 GHz ;
8. $k \neq 0 \Rightarrow b = 0$;
9. $c < d$;
10. the norm of the vector m is defined by η as follows:

$$\eta(m) = \sqrt{c^2 + d^2 + p^2 + k^2 + l^2 + r^2 + b^2}.$$

Remark 1. If $b = 0$ then the machine is a personal computer, otherwise it is a router.

Remark 2. If $k = 0$ then the PC has no connected devices (basic devices are not included).

Remark 3. When the processor is multi-core, the frequency of the machine is the maximum of the frequencies of all the cores.

3.2 Computer Network

A computer network is a set of computer elements (computers, routers, printers, etc.) connected to each other [15]. A computer network makes it possible to share data, documents, applications and printers [15].

Depending on the context, the term network may refer to the architecture, the prototype or the IT infrastructure. With the evolution of the Internet, several networking technologies have emerged to overcome the difficulties encountered in deploying the computer network and to meet the growing needs of businesses. Network virtualization was thus born. A network environment supports network virtualization if it allows the coexistence of several virtual networks on the same physical infrastructure [4].

A virtual network is a set of virtual equipment (computers, routers, etc.) and virtual links interconnected with each other. Virtual equipment and virtual links are created thanks to a software layer called the Virtual Machine Monitor (VMM) or hypervisor [9, 16]. It virtualizes all the resources of a physical machine, thus defining and supporting the running of several virtual machines.

IT migration must therefore take account of the network technology used and the characteristics of the network.

2.3 Advantages of Migration

IT migration has several advantages, including [9]:

- improving system performance: the aim is to increase the performance of an application or business by adding new functions to the system;
- energy savings: the migration is carried out in such a way as to support applications running on a minimum number of servers. This is done to maximize the use of resources and for energy management. This is done to maximize resource utilization and for energy management;
- ease of maintenance: migration simplifies maintenance tasks, helping to reduce downtime due to system maintenance. Migration can also be used when you want to replace one computer system with another, where the information relating to the latter is migrated to another environment and returned after the new system has been installed;
- fault tolerance: migration is carried out in the event of a system fault, to increase the availability of the services provided by the system.

3 Machine Migration Concept

3.1 Notion of Machine

A machine can be defined as an electronic and programmable device capable of automatically and rationally processing information. A machine here is either a personal computer or a router. The elements constituting the operating context of a machine are characterized by:

- the state of its central memory;
- the state of its hard disk;
- the state of its processors;
- the state of its connected devices (basic devices are not counted);
- the state of its softwares (installed applications and system software);
- the state of its active network connections;
- the state of its bandwidth if the machine is a router.

We can formally define a machine as a state vector: $m = (c, d, p, k, l, r, b)$ where c, d, p, k, l, r, b represent capacity of main memory, hard disk capacity, processor frequency, number of connected devices, number of software applications, type of network and bandwidth capacity.

We make the following assumptions:

1. $m \neq 0$;
2. we take as the minimum values for a personal computer those verifying the minimum characteristics required for the Windows XP system [5];
3. $c, d, k, l, r \in \mathbb{N}$ with $l \neq 0$;
4. $c \geq 64 \text{ MB}$, $d \geq 2 \text{ GB}$, et k, l, r are binary numbers;

- internally: this refers to any modification of system variables and requirements aimed at improving the existing IT system. Example: upgrading an operating system or an application;
- externally: this involves the replacement of a computer system or the abandonment of one infrastructure in favor of another. It may also involve moving the system from one physical location to another.

To facilitate these migrations, it may be useful to implement careful planning and an infrastructure automation strategy [6].

2.1 Migration Conditions

As a general rule, before starting any IT migration, you need to make sure that:

- the storage space of the final environment is greater than or equal to the storage space of the initial environment;
- the architecture of the target environment is fairly close to or at least has the same characteristics as the source environment;
- the applications are compatible with the operating system.

2.2 Types of Migration

Depending on the project, IT migration may involve one or more types of move. There are therefore 03 main types of IT migration [3, 6, 7]:

- data (or storage) migration: this involves moving data from one type of storage system to another. This process is often carried out as part of an upgrade aimed at increasing storage capacity, improving performance, reducing costs, reducing footprint or adding new capacity. During migration, data must be moved between two database engines. The challenge of any database migration is to implement it without affecting the data language or reading protocol. A database migration is successful when the tools implemented manage to modify the data without altering the structure of the database;
- Software migration: this can involve either the operating system or application software. Software migration involves moving software from one computer system to another. It can be time-consuming and involve a number of risks, including downtime, incompatibility between applications and the loss of customized settings;
- migration to the cloud: this involves moving IT systems from traditional on-site data centers to cloud environments, or from one cloud environment to another.

process that can take several forms [3]: a transfer of data from one storage space to another, a transformation from one data format to another, or a conversion to make raw data usable on a particular type of system.

Salah-Eddine [2] affirms in his thesis work that a few years ago, the migration of virtual machines was not done in real time, but only after a complete stop of the virtual machines. This could be explained by the lack of automatic tools for real-time migration of virtual machines. However, in response to the rapid growth in the number of virtual machines and virtual networks, automated tools for real-time migration of virtual machines have become essential. In addition, given the current economic context, virtual machine networking service providers have an interest in carrying out rigorous migrations in order to retain their customers by offering them continuous services that are better than their competitors, thus enabling them to gain more market share [2]. Virtual networks and virtual machines are becoming increasingly popular, and the number of users is growing all the time. Consequently, providers of these services have no choice but to increase their investments in order to respond effectively to the growing needs of their customers [2]. Despite the potential vision of virtual machine migration, several important research challenges have been addressed and remain to be tackled.

For some years, several scientific studies have focused on the informal study of the virtual machine migration problem. In this article, we present a formal definition of the virtual machine migration problem. Our contributions are summarized in three points:

- formally define the migration environment for a machine;
- formally define the migration environment for a virtual machine;
- show that virtual machine migration is a rucksack problem

The rest of this article is organized as follows. In Sect. 2, we present a general overview of IT migration. Section 3 describes the concept of machine migration. Section 4 describes the migration of a virtual machine. The problem of migrating a virtual machine is presented in Sect. 5. Our virtual machine migration method is presented in Sect. 6. Our method is discussed in Sect. 7, and Sect. 8 concludes the paper.

2 General Overview of IT Migration

IT migration can be defined as the passage of an IT system from an initial execution environment to a final environment.

A computer system is a set of hardware and software computing and telecommunications resources whose purpose is to collect, process, store, route and present data [8].

The execution environment can be defined as the set of elements required for a computer system to function properly. Examples of environments include: a room, a server, a machine, software, a network, the cloud.

IT migration projects typically have many company-specific variables and requirements [6]. IT migration can be done at two levels:



Proposal for a Formal Definition of the Virtual Machine Migration Problem

Thomas Djotio Ndie¹(✉) , Joel Casimir Tagne¹ , and Karl Jonas²

¹ University of Yaounde I, Yaoundé, Cameroon
tdjotio@gmail.com, joelcasimirt@gmail.com

² Bonn-Rhein-Sieg University of Applied Sciences, Rheinbach, Germany
karl.jonas@h-brs.de

Abstract. Thanks to the development of New Information and Communication Technologies (NICT), computer tools are increasingly in demand in almost all sectors of activity to meet to the needs of people, property, companies... Thus, IT migration is therefore a process that can affect all companies that host data relating to their customers, suppliers, partners or even general statistics.

In this article, we consider the virtual machine migration problem as a rucksack problem. As server hosting capacities have become increasingly ‘elevated’, the need for consolidation and load balancing has led to a strong interest in virtual machine migration.

We have used the following approach: we begin by presenting the computer system migration environment. Next, we propose a formal definition of the virtual machine migration environment. We then show, using the formal language, that virtual machine migration is an NP-hard problem of the rucksack type. We then propose the MigrationVM algorithm of complexity $O(7)$ as an optimal solution to the problem of migration virtual machines in networks. We have discussed our algorithm and we believe that virtual machine migration data can be reduced to two main elements: the amount of RAM used by the virtual machine and the amount of hard disk used by the virtual machine.

Keywords: Virtual machine · Network · Service · Migration · Migration time

1 Introduction

Thanks to the development of New Information and Communication Technologies (NICT), computer tools are increasingly in demand in almost all sectors of activity to meet the needs people, property, companies... Thus, IT migration is therefore a process that can affect all companies that store customer data, relating to their customers, suppliers, partners or even general statistics [3]. Migrating data may be necessary when opting for a larger storage space or when implementing a new database management strategy. IT migration is a data transfer

Systems and Cloud Computing

Cybersecurity and Privacy

The State of Data Breaches in the African Cyberspace: A Trend Analysis Using Social Media and Research Literature	259
<i>Jabu Mtsweni, Muyowa Mutemwa, Mfundo Masango, Samson Chishiri, and Siwe Moyakhe</i>	
Advancing Mobile Money Payments Through Blockchain and Interoperability Protocols	274
<i>Edem Kodjo Agbezoutsu, Pascal Urien, and Toundé Mesmin Dandjinou</i>	
Blackhole Attack Detection and Countermeasure Solution in RPL	288
<i>Fatiè Daoud Idriss Siéba, Hamadoun Tall, Amado Illy, and Tiguiane Yélémou</i>	
Social Engineering Attacks on the Cyber-Physical System: Human Cyber and Physical Impacts	296
<i>Robert Makila Beni</i>	
Proposal of Honeypot-Based Data Mining Methods for the Discovery of Intrusions in Big Data Databases	312
<i>Koffi Kanga, Beman Hamidja Kamagaté, Raogo Kabore, and Souleymane Oumtanaga</i>	
Potential Cyber Threats to the National Elections in the Digital Age in Africa	333
<i>Thuli Mkhwanazi, Avuya Shibambu, Vhuthu Nefale, Jabu Mtsweni, Jackie Phahlamohlaka, Muyowa Mutemwa, and Norman Nelufule</i>	
Intersection of Electronic Security and Digital Forensics: Data Protecting Techniques and Uncovering Data Clues	351
<i>Norman Nelufule, Boitumelo Nkwe, Daniel Shadung, Kele Masemola, Taniita Singano, Japhtalina Mokoena, Zamo Ngubane, and Ntombizodwa Thwala</i>	
Emerging Phishing Attack Trends: A South African Case Study	368
<i>Jabu Mtsweni, Precious Maduma, Vhuthu Nefale, Alex Ramantswana, Mfundo Masango, and Muyowa Mutemwa</i>	
5G Network Security: Unraveling Vulnerabilities and Innovating Defense Mechanisms	383
<i>Mamoon M. Saeed, Elmustafa Sayed Ali, Othman O. Khalifa, and Rania A. Mokhtar</i>	
Author Index	393