Contents available at the publisher website: G A F T I M . C O M

International Journal of Theory of Organization and Practice (IJTOP)

Journal homepage: https://journals.gaftim.com/index.php/ijtop/index

Java-Powered AI: Using Code to Implement Intelligent Systems – Systematic Review

Sara AlAraj[1], Meera AlAraj[1]

[1]*United Arab Emirates*

A R T I C L E I N F O

Keywords:
JAVA, Advanced Business Programming, Natural Language Processing, Java Programming, Artificial Intelligence (AI), Machine Learning, Deep Learning.

Received: Dec, 22, 2023
Accepted: Jan, 17, 2024
Published: Feb, 12, 2024

A B S T R A C T

The combination of Java programming and artificial intelligence (AI) creates a potent synergy that allows developers to design intelligent systems and applications that are scalable, reliable, and efficient. This paper examines how to combine the flexibility of Java with the cognitive powers of AI, offering a range of approaches, tools, and frameworks that make use of Java's advantages for developing AI-driven solutions. An introduction of AI ideas and Java's place in AI development is given at the outset of the paper. It explores basic AI techniques like computer vision, reinforcement learning, machine learning, and natural language processing (NLP), and explains how to implement them in Java using frameworks like Deep learning, Weka, and Apache Open NLP. It also goes over how to use Java to create intelligent agents and how to design expert systems, heuristic-driven algorithms, and autonomous decision-making systems. It emphasizes how Java may be integrated with AI-enabled tools and stresses the significance of feature engineering, model deployment, and data pretreatment. Additionally, the study looks at the difficulties and possibilities in developing AI using Java, addressing issues with interoperability of AI modules, compatibility with a variety of data sources, and performance optimization. The study culminates with a futuristic view of Java-powered artificial intelligence, imagining progressions in Java libraries, frameworks, and techniques that will encourage the development of increasingly complex and intelligent systems.

1. INTRODUCTION

Combining Java programming with artificial intelligence (AI) has opened up new creative possibilities by enabling programmers to create intelligent systems that are not limited by convention. The combination of AI's cognitive capabilities and Java's resilience shows how endless the possibilities are when it comes to creating intelligent apps, especially in this era of swift technological progress. This article is a thorough investigation into the field of Java-powered AI, where we examine the mutually beneficial link between the range of AI approaches and Java's adaptable environment. It explores the field of artificial intelligence (AI) techniques and algorithms, explaining how they are implemented in the Java framework so that developers can use AI for a variety of purposes. From computer vision and reinforcement learning to machine learning and natural language processing, the first sections of the book provide a strong foundation by demystifying key AI ideas. By using this lens, we reveal how Java's scalability and flexibility allow it to easily integrate with these AI paradigms, giving developers a strong foundation on which to create intelligent systems. Additionally, this article explores several Java-based AI frameworks and libraries, highlighting programs such as Weka, Apache OpenNLP, Deeplearning4j, and others. Through analyzing their features and presenting

real-world applications, developers learn how to best utilize these frameworks to efficiently produce AI-driven solutions. Beyond algorithmic skill, the study investigates Java's application in the development of intelligent agents, self-governing decision-making systems, and expert systems. It sheds light on the process of creating complex AI models, highlighting the significance of feature engineering, data pretreatment, and the best ways to deploy models inside the Java environment. Nonetheless, this investigation does not downplay the difficulties that come with developing AI using Java. Although there are many obstacles to overcome, there are also many chances for creative solutions and breakthroughs. These obstacles include performance optimization, compatibility with different data sources, and the smooth integration of AI modules. Essentially, this paper serves as a guide to fully utilize Java's potential for creating intelligent systems, making it a valuable resource for both developers and AI aficionados. Looking ahead, we can see that Java libraries, frameworks, and approaches will continue to advance, opening the door for even more complex AI-driven developments.

When developing intelligent systems with code, Java-powered AI plays a vital role in multiple ways: Flexible Development Environment: Java provides a cross-platform, flexible environment for developing artificial intelligence. Because of its adaptability, numerous intelligent systems in a variety of industries, including e-commerce, healthcare, and finance, can be developed. Robustness and Scalability: When building AI algorithms and models, Java's robustness and scalability come in handy. This guarantees the created systems' ability to efficiently handle complex computations and large-scale data processing. Integration of AI Libraries and Frameworks: Deeplearning4j, Weka, and Apache OpenNLP are just a few of the many AI libraries and frameworks that can be easily integrated with Java. These tools facilitate the smooth integration of computer vision, natural language processing, machine learning, and other AI methods into Java programs. Creation of Autonomous Decision-Making Systems and Intelligent Agents: Java enables the development of autonomous decision-making systems and intelligent agents. This capacity is essential for developing systems that can analyze data, draw conclusions, and adjust to shifting conditions. Effective Data Processing and Model Deployment: The effectiveness of AI systems is greatly enhanced by Java's skills in feature engineering, data preprocessing, and model deployment. This guarantees efficient data pipelines and successful AI model deployment in real-world settings. Java places a strong emphasis on compatibility and performance optimization across many platforms and data sources, which is essential for guaranteeing the dependability and effectiveness of AI systems built with Java. Interoperability and extensibility: Integrating AI modules into current systems is made easier by Java's compatibility with a variety of other programming languages and technologies. Furthermore, as new needs arise, Java's extensibility enables the improvement and modification of AI systems. Encouraging Innovation and Upcoming Development: The combination of AI and Java stimulates creativity and opens the door to the creation of increasingly complex and intelligent systems. Java's ongoing development and the progress made in AI algorithms portend a time when more intelligent applications will be the norm.

Visualizing Hadoop Execution Flow via Mapping Java Event Traces

Java event traces in Hadoop execution are usually represented visually by diagrams or graphs that highlight the order and relationships between Java-based events in the Hadoop ecosystem. The sequence of events, including MapReduce stages, Task and Job Tracker activities, logging and debugging procedures, API calls, job execution flow, resource utilization, exceptions, and failures, are shown visually in this diagram. It offers a thorough rundown of the sequence of activities involving Java that take place when Hadoop jobs are being executed.

2. LITERATURE REVIEW

2.1. The schematic could show:

Event Flow: A graphic representation of the series of actions that a Hadoop job goes through from start to finish. Job submission, work scheduling, data processing, and job termination may all fall under this category.

Component Interactions*:* This diagram illustrates how different Hadoop components, such as the Nam Node, Data Node, Resource Manager, Node Manager, Task Tracker, and Job Tracker, interact

with one another. This demonstrates how various elements interact and cooperate while carrying out the task.

Custom Traces*:* Including particular event types or custom traces that are pertinent to the context, which may be necessary for tracking and examining the execution flow.

The goal of the visual representation is to give an understandable, organized, and thorough summary of the Java event traces in the Hadoop ecosystem. It helps analysts, administrators, or developers comprehend the complex job execution flow and maximize Hadoop task performance.

A thorough representation of the complex flow of events during Hadoop job execution in a Java context may be found in the visualization named "Visual Representation of Java Event Traces in Hadoop Execution." The order and relationships between Java-based events that take place at different phases of Hadoop job processing are depicted in this visual aid. It provides a concise synopsis of the key elements, actions, and changes that occur during Java events in the Hadoop environment.

Sequencing of Events: The flow chart illustrates the steps involved in completing a project, from the time it is submitted to its completion. It highlights important stages such as mapping, shuffle, and reduce tasks. Interactions Among Components: It shows how important Hadoop components including Job Tracker, Task Tracker, Name Node, Data Node, Resource Manager, and Node Manager interact and communicate with one another. This illustrates how well they worked together at various stages of the job's implementation.

Resource Utilization Metrics: Throughout the Hadoop cluster, graphical displays provide information on how system resources like CPU, memory, disk I/O, and network bandwidth are used. An knowledge of resource allocation and possible performance-affecting bottlenecks is provided by this depiction. These additions provide more context or focused controlling capacities. For developers, administrators, or analysts working on Hadoop task optimization, debugging, or performance enhancement, the visualization is a useful tool. It facilitates understanding of the intricate execution flow and informs decision-making to improve Hadoop job processing performance by giving a visual depiction of the Java event traces.

The use of Java-powered AI to construct intelligent systems has numerous, wide-ranging implications and advantages. Here are a few main outcomes and advantages:

Increased Automaticity and Efficiency: AI driven by Java makes it possible to create intelligent systems that automate jobs, optimize workflows, and expedite procedures. In many industries, this efficiency improvement results in lower costs and higher productivity.

Better Decision-Making: Intelligent systems that are constructed with AI and Java algorithms are able to evaluate large volumes of data, derive significant insights, and make decisions based on that data. By assisting companies and organizations in making well-informed decisions, this skill enhances strategic planning and results.

Personalization and User Experience: By analyzing user behavior and preferences, AI-powered Java systems are able to provide tailored experiences.

Increased user happiness with all applications, services, and goods results from this.

Improvements in Healthcare and medication: The creation of intelligent diagnostic tools, tailored medication, predictive analytics for patient care, and the diagnosis analysis of medical images are all made possible by Java-powered AI, which has the potential to completely transform the healthcare industry. Effective Customer Service and service: AI-powered chatbots and virtual assistants built with Java may give effective customer service, addressing regular chores, resolving problems, and providing prompt answers to inquiries. This increases customer satisfaction.

Enhanced Security Measures: The combination of AI's anomaly detection skills and Java's resilience allows for the development of complex security systems. Real-time threat detection and response capabilities of these systems improve cybersecurity protocols for a range of applications. Optimized Resource Allocation: Supply chain management, logistics, and transportation are just a few industries where AI systems running on Java can achieve optimal resource allocation. Better inventory control, route optimization, and economical resource use result from this.

Technological Advancements and creativity: The combination of AI with Java encourages creativity, resulting in the development of cutting-edge technologies like predictive maintenance,

smart cities, and autonomous cars industrial maintenance, among other things. Inclusivity and Accessibility: By addressing a range of user needs, including those of people with impairments, Java-powered AI systems can be created with an emphasis on accessibility. Sustained Evolution and Adaptability: Java's large community support combined with AI's quick development means that Java-powered AI systems can keep up with the newest AI methods and approaches while also evolving and adapting to be relevant in a constantly shifting technological environment.
In conclusion, Java-powered AI plays a critical role in the implementation of intelligent systems through code, utilizing the robustness, scalability, integration capabilities, and performance optimization of Java to produce AI-driven solutions in a variety of fields. In conclusion, there are numerous advantages and effects of Java-powered AI that are felt by organizations and society alike. These include increased productivity, creativity, better decision-making, better user experiences, and many more beneficial outcomes [1]–[4].
The development of intelligent agents is a fundamental component of the constantly changing field of artificial intelligence (AI) and its intersection with Java-based engineering. The creation of intelligent agents—entities with the ability to sense their surroundings, reason, and make decisions—has been a key focus of AI engineering as technology develops. With an emphasis on the application of intelligent agents, this study undertakes a thorough investigation of Java-based AI engineering [5]. Through the use of Java's stability, adaptability, and wide ecosystem, this project aims to clarify the concepts, procedures, and real-world uses of building intelligent agents across a range of fields. The foundation of this investigation is the incorporation of AI techniques into the Java framework [1], [2], [6], [7]. Java's abundance of libraries and frameworks, along with its programming language skills, make it an ideal platform for creating intelligent systems. The goal of the study is to explore the subtleties of exploiting Java's many features to construct intelligent agents inside this paradigm. Essentially, the paper explores basic AI ideas that are necessary to understand intelligent agents [3], [4], [8], [9]. Ideas like autonomy, learning, knowledge representation, decision-making, and self-organization serve as the foundation for developing intelligent entities that can interact and adapt to their surroundings. Additionally, the paper will go over how important Java is to making the creation and use of intelligent agents easier. The investigation highlights the useful features of creating intelligent agents in Java, from taking advantage of the language's object-orientedness to using its robust frameworks for data processing and computing. This project also discusses the ethical implications of AI and responsible engineering in the context of Java-based intelligent agents. Transparency, justice, and responsibility are important ethical factors in AI research, and this talk must include how these factors are integrated into Java-powered intelligent agents. The paper will also highlight use examples and real-world applications where Java-based intelligent agents have contributed significantly. These examples demonstrate the broad influence of intelligent agents created with Java-based AI engineering, ranging from autonomous systems in robotics to personalized recommendation engines in e-commerce [10]. This paper essentially acts as a roadmap for researchers, engineers, and enthusiasts who are interested in Java-based AI engineering. It provides methods, insights, and useful information that can help developers create intelligent agents that are both complex and flexible [11], [12].

3. DATA ANALYSIS

When building intelligent agents, Java-based AI engineering plays a crucial role in multiple important ways: Sturdy Development environment: With a large library, a thriving ecosystem, and programming support for AI engineering, Java offers a stable programming environment. This makes it possible for engineers to create scalable, dependable codebases for intelligent agents. Versatility and Flexibility: Java's adaptability enables the use of different AI methods and algorithms in intelligent agents [1], [2], [7]. Its adaptability to a wide range of AI applications stems from its agility in processing various data formats and integrating diverse functionalities. Object-Oriented Method: Java's object-oriented design makes it easy to represent intelligent agents and makes it possible to create modular, reusable, and organized components [3], [4], [8], [9]. In agent development, this method

encourages code organization and maintainability. AI Algorithm Integration: Intelligent agent decision-making, learning, reasoning, and planning algorithms are smoothly integrated with Java [13]–[15]. Agents may now independently assess data, adjust to changing situations, and make well-informed judgments thanks to this connection [16]–[18]. Efficient Data Processing: Intelligent agents rely on Java's data handling and manipulation capabilities to process and manage their data sources. It permits effective feature extraction, manipulation, and preparation of data vital to the functioning of the agent [19].

Resource Management and Optimization: Java-based AI engineering gives programmers the ability to efficiently manage system resources, which is essential for intelligent agents with limited resources [6], [20]–[22]. This includes optimizing memory use and guaranteeing effective performance. Platform Independence: Intelligent agents may operate on a variety of devices and systems without requiring major changes because to Java's platform independence. This guarantees that Java-based intelligent agents can be widely deployed and work in a variety of situations. Ethical AI Development: AI engineering based on Java promotes moral behavior in AI development. By emphasizing responsibility, transparency, and justice, developers can include moral considerations into intelligent agents and advance responsible AI.

Real-World Applications: Intelligent agents with Java capabilities find use in a variety of industries, including robotics, the Internet of Things, banking, healthcare, and more. Their application in real-world settings demonstrates the usefulness and adaptability of AI engineering based on Java [23].

Proceeded Innovation and Evolution: Java-based AI engineering is continuously innovative due to Java's continuing evolution and advances in AI research [24]–[26]. This makes it easier to create intelligent agents that are more complex and flexible. Java-based AI engineering, in its essence, leverages Java's capabilities in flexibility, scalability, robustness, and ethical considerations to enable the design, development, and deployment of intelligent agents [16]–[18], [27]. Using Java-based AI engineering to implement intelligent agents has a number of advantages and consequences across a range of applications and domains: Enhanced Automation and Efficiency: Java-based intelligent agents reduce human intervention and increase productivity by automating jobs, streamlining workflows, and enhancing overall efficiency across a variety of areas. Better Decision-Making: By analyzing large datasets, drawing conclusions, and making data-driven choices, intelligent agents created using Java-based AI engineering improve decision-making processes in sectors including banking, healthcare, and logistics [28]. Personalized experiences are made possible by these agents, which are capable of learning from user interactions and data patterns. Examples of these applications include content curation, recommendation systems, and adaptive user interfaces. Predictive Capabilities: Businesses may make proactive decisions and create strategies by utilizing Java-powered intelligent agents that employ predictive analytics to foresee trends, behavior, and consequences [13]–[15]. Optimal Resource Allocation: In domains like logistics and supply chain management, intelligent agents effectively distribute resources, manage inventories, streamline routes, cut down on waste, and save money [11], [12]. Improvements in Healthcare: Intelligent agents help with medication discovery, patient monitoring, tailored therapy regimens, and medical diagnostics, all of which improve patient outcomes. Enhanced Security Measures: Java-based intelligent agents support cybersecurity efforts in a variety of systems and networks by spotting irregularities, spotting dangers, and reacting quickly. Effective Customer Support: Chatbots and virtual assistants created using Java-based AI engineering provide round-the-clock assistance, quickly responding to questions from clients, resolving problems, and raising client satisfaction levels [11], [12], [29]. Java's emphasis on organized coding techniques and ethical concerns guarantees the responsible development of intelligent agents, hence fostering accountability, transparency, and justice in AI systems. This leads to the development of ethical AI systems. Technological Advancements: The future of interconnected systems and smart surroundings is being shaped by the integration of Java-based AI engineering in fields like as robotics, autonomous systems, smart cities, and IoT devices. Constant Evolution and Adaptability: Because of Java's extensibility and the constantly changing AI landscape, intelligent agents based on Java can

easily integrate new AI techniques and breakthroughs and adapt to changing requirements [20]–[22], [27].

4. ADVANTAGES

In conclusion, the advantages and impacts of using Java-based AI engineering to create intelligent agents include increased productivity, better judgment, tailored experiences, efficient use of resources, advancements across multiple industries, and a constant push for creative and moral AI development [29]. To sum up, the benefits and effects of developing intelligent agents through Java-based AI engineering include higher productivity, improved judgment, customized experiences, effective resource management, advancements in a variety of industries, and a persistent drive for ethical and creative AI development.

5. CONCLUSION

The creation of intelligent systems has reached previously unheard-of heights thanks to the combination of artificial intelligence with Java programming. We have explored the intersection of code and intelligence through this investigation of Java-powered AI, opening up a world of opportunities in various fields and sectors. Because of Java's stability, scalability, and adaptability, AI approaches have been easily incorporated, allowing programmers to create complex solutions for a wide range of technological applications. The exploration of basic AI algorithms, such as computer vision, natural language processing, and machine learning, additionally, demonstrates how flexible Java is in putting these strategies into practice. With the help of frameworks and libraries like Deeplearning4j, Weka, and Apache OpenNLP, developers are now able to fully utilize AI within the Java environment. These technologies have made it easier to create intelligent agents, self-governing decision-making systems, and sophisticated AI models, especially when combined with Java's natural advantages. The benefits of AI driven by Java are numerous and varied, ranging from stronger security measures to better decision-making, tailored experiences, and efficiency gains to healthcare improvements. Looking ahead, the combination of AI and Java promises ground-breaking breakthroughs and ongoing evolution. Java-powered AI is a paradigm change in technology that will propel us into a future in which intelligent systems powered by Java continue to reinvent human capabilities. It is more than just the confluence of code and intelligence [24]–[26].

REFERENCES

[1] S. Khadragy *et al.*, "Predicting Diabetes in United Arab Emirates Healthcare: Artificial Intelligence and Data Mining Case Study," *South East. Eur. J. Public Heal.*, vol. 5, 2022, doi: https://doi.org/10.56801/seejph.vi.406.

[2] R. Ravikumar *et al.*, "Impact of knowledge sharing on knowledge Acquisition among Higher Education Employees," *Comput. Integr. Manuf. Syst.*, vol. 28, no. 12, pp. 827–845, 2022, doi: 10.24297/j.cims.2022.12.58.

[3] M. Salameh *et al.*, "The Impact of Project Management Office's Role on Knowledge Management: A Systematic Review Study," *Comput. Integr. Manuf. Syst.*, vol. 28, no. 12, pp. 846–863, 2022, doi: 10.24297/j.cims.2022.12.59.

[4] F. Shwedeh, S. Malaka, and B. Rwashdeh, "The Moderation Effect of Artificial Intelligent Hackers on the Relationship between Cyber Security Conducts and the Sustainability of Software Protection: A Comprehensive Review," *Migr. Lett.*, vol. 20, no. S9, pp. 1066–1072, 2023, doi: 10.59670/ml.v20iS9.4947.

[5] F. Bu, H. wu, H. A. Mahmoud, H. M. Alzoubi, N. K. Ramazanovna, and Y. Gao, "Do financial inclusion, natural resources and urbanization affect the sustainable environment in emerging economies," *Resour. Policy*, vol. 87, p. 104292, 2023, doi: 10.1016/j.resourpol.2023.104292.

[6] S. Salloum *et al.*, "Understanding and Forecasting Chatbot Adoption: An SEM-ANN Methodology," *Migr. Lett.*, vol. 20, no. S11, pp. 652–668, 2023, doi: https://doi.org/10.59670/ml.v20iS11.5717.

[7] F. Shwedeh, N. Hami, S. Z. Abu Bakar, F. M. Yamin, and A. Anuar, "The Relationship between Technology Readiness and Smart City Performance in Dubai," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 29, no. 1, pp. 1–12, 2022, doi: https://doi.org/10.37934/araset.29.1.112.

[8] R. Ravikumar *et al.*, "The Impact of Big Data Quality Analytics on Knowledge Management in Healthcare Institutions: Lessons Learned from Big Data's Application within The Healthcare Sector," *South East. Eur. J. Public Heal.*, vol. 5, 2023, doi: https://doi.org/10.56801/seejph.vi.309.

[9] F. Shwedeh, "Harnessing digital issue in adopting metaverse technology in higher education institutions: Evidence from the United Arab Emirates," *Int. J. Data Netw. Sci.*, vol. 8, no. 1, pp. 489–504, 2024, doi: 10.5267/j.ijdns.2023.9.007.

[10] K. Liu *et al.*, "Exploring the Nexus between Fintech, natural resources, urbanization, and environment sustainability in China: A QARDL study," *Resour. Policy*, vol. 89, p. 104557, 2024, doi: 10.1016/j.resourpol.2023.104557.

[11] F. Shwedeh, N. Hami, and S. Z. Abu Bakar, "Dubai smart

city and residence happiness: A conceptual study," *Ann. Rom. Soc. Cell Biol.*, vol. 25, no. 1, pp. 7214–7222, 2021.

[12] B. M. Dahu *et al.*, "The Impact of COVID-19 Lockdowns on Air Quality: A Systematic Review Study," *South East. Eur. J. Public Heal.*, vol. 5, 2022, doi: https://doi.org/10.11576/seejph-5929.

[13] A. El Nokiti, K. Shaalan1, S. Salloum2, A. Aburayya, F. Shwedeh, and B. Shameem3, "Is Blockchain the answer? A qualitative Study on how Blockchain Technology Could be used in the Education Sector to Improve the Quality of Education Services and the Overall Student Experience," *Comput. Integr. Manuf. Syst.*, vol. 28, no. 11, pp. 543–556, 2022, doi: 10.24297/j.cims.2022.11.039.

[14] S. A. Alimour *et al.*, "The quality traits of artificial intelligence operations in predicting mental healthcare professionals' perceptions: A case study in the psychotherapy division," *J. Auton. Intell.*, vol. 7, no. 4, 2024, doi: 10.32629/jai.v7i4.1438.

[15] F. Shwedeh, "THE IMPACT OF SMART CITY POLICY TIMELINESS AND TECHNOLOGY READINESS ON SMART CITY PERFORMANCE IN DUBAI: THE MODERATING EFFECT OF FINANCIAL AVAILABILITY," 2021.

[16] N. Yas, M. N. I. Elyat, M. Saeed, F. Shwedeh, and S. Lootah, "The Impact of Intellectual Property Rights and the Work Environment on Information Security in the United Arab Emirates," *Kurd. Stud.*, vol. 12, no. 1, pp. 3931–3948, 2024, doi: 10.58262/ks.v12i1.282.

[17] F. Shwedeh, A. Aburayya, and M. Mansour, "The Impact of Organizational Digital Transformation on Employee Performance: A Study in the UAE," *Migr. Lett.*, vol. 20, no. S10, pp. 1260–1274, 2023, doi: https://doi.org/10.59670/ml.v20iS10.5710.

[18] F. Shwedeh, N. Hami, and S. Z. Abu Baker, "Effect of leadership style on policy timeliness and performance of smart city in Dubai: a review," in *Proceedings of the International Conference on Industrial Engineering and Operations Management Dubai, UAE, March 10-12, 2020*, 2020, pp. 917–922.

[19] C. Leng *et al.*, "An empirical assessment of the effect of natural resources and financial technologies on sustainable development in resource abundant developing countries: Evidence using MMQR estimation," *Resour. Policy*, vol. 89, p. 104555, 2024, doi: 10.1016/j.resourpol.2023.104555.

[20] F. Shwedeh *et al.*, "Entrepreneurial innovation among international students in the UAE: Differential role of entrepreneurial education using SEM analysis," *Int. J. Innov. Res. Sci. Stud.*, vol. 6, no. 2, pp. 266–280, 2023, doi: https://doi.org/10.53894/ijirss.v6i2.1328.

[21] F. Shwedeh, T. Aldabbagh, A. Aburayya, and H. Uppilappatta, "The Impact of Harnessing Total Quality Management Studies on the Performance of Smart Applications: A Study in Public and Private Sectors in the UAE," *Migr. Lett.*, vol. 20, no. S11, pp. 934–959, 2023, doi: https://doi.org/10.59670/ml.v20iS11.5892.

[22] F. Shwedeh *et al.*, "SMEs' Innovativeness and Technology Adoption as Downsizing Strategies during COVID-19: The Moderating Role of Financial Sustainability in the Tourism Industry Using Structural Equation Modelling," *Sustainability*, vol. 14, no. 23, p. 16044, 2022, doi: https://doi.org/10.3390/su142316044.

[23] Q. Hassan *et al.*, "The renewable energy role in the global energy Transformations," *Renew. Energy Focus*, vol. 48, p. 100545, 2024, doi: https://doi.org/10.1016/j.ref.2024.100545.

[24] A. Aburayya *et al.*, "SEM-machine learning-based model for perusing the adoption of metaverse in higher education in UAE.," *Int. J. Data Netw. Sci.*, vol. 7, no. 2, pp. 667–676, 2023, doi: 10.5267/j.ijdns.2023.3.005.

[25] M. Alkashami *et al.*, "AI different approaches and ANFIS data mining: A novel approach to predicting early employment readiness in middle eastern nations," *Int. J. Data Netw. Sci.*, vol. 7, no. 3, pp. 1267–1282, 2023, doi: 10.5267/j.ijdns.2023.4.011.

[26] S. Salloum *et al.*, "Sustainability Model for the Continuous Intention to Use Metaverse Technology in Higher Education: A Case Study from Oman," *Sustainability*, vol. 15, no. 6, p. 5257, 2023, doi: 10.3390/su15065257.

[27] S. Abdallah *et al.*, "A COVID19 Quality Prediction Model based on IBM Watson Machine Learning and Artificial Intelligence Experiment," *Comput. Integr. Manuf. Syst.*, vol. 28, no. 11, pp. 499–518, 2022, doi: 10.24297/j.cims.2022.11.037.

[28] B. Li, S. Mousa, J. R. R. Reinoso, H. M. Alzoubi, A. Ali, and A. D. Hoang, "The role of technology innovation, customer retention and business continuity on firm performance after post-pandemic era in China's SMEs," *Econ. Anal. Policy*, vol. 78, pp. 1209–1220, 2023, doi: 10.1016/j.eap.2023.05.004.

[29] S. Khadragy *et al.*, "Predicting Diabetes in United Arab Emirates Healthcare: Artificial Intelligence and Data Mining Case Study," *South East. Eur. J. Public Heal.*, vol. 5, 2022, doi: https://doi.org/10.56801/seejph.vi.406.