

Cloud-Based Engineering Platforms for Scalable Technical Applications

Khuzaima Daudjee

Professor, Cheriton School of Computer Science, University of Waterloo, Canada

Abstract

Cloud computing has become an important technological foundation for developing scalable engineering platforms capable of supporting complex technical applications, distributed workloads, and rapidly changing computational requirements. Traditional engineering environments frequently depend on fixed infrastructure, localized computational resources, and manually managed deployment processes, which can restrict scalability and increase operational complexity. Cloud-based engineering platforms address these limitations by integrating elastic computing, virtualized infrastructure, distributed storage, containerized applications, application programming interfaces, and automated resource management into unified technical environments. This paper examines the role of cloud-based engineering platforms in enabling scalable technical applications across engineering-oriented computational and operational contexts. The study analyzes major architectural components, scalability mechanisms, deployment approaches, resource optimization strategies, and security considerations associated with cloud-enabled engineering platforms.

A conceptual analytical methodology is used to evaluate the relationship between cloud infrastructure capabilities and technical application scalability. The analysis indicates that elastic resource allocation, container orchestration, distributed processing, automated deployment, and cloud-native service architectures can substantially improve application flexibility and resource utilization. However, scalability is influenced not only by infrastructure capacity but also by application architecture, workload characteristics, network performance, data management, security requirements, and operational governance. The paper further discusses challenges involving interoperability, cloud dependency, latency, data protection, cost management, and technical complexity. Future research opportunities include intelligent resource allocation, AI-assisted cloud engineering, edge-cloud integration, serverless engineering applications, digital twins, and autonomous cloud infrastructure. The study concludes that appropriately designed cloud-based engineering platforms can provide a flexible and scalable foundation for modern technical applications when architectural, operational, security, and economic considerations are addressed systematically.

Keywords

Cloud Computing; Engineering Platforms; Scalable Applications; Cloud-Native Architecture; Distributed Computing; Containerization; Resource Management; Technical Applications; Virtualization; Cloud Infrastructure.

1. Introduction

Engineering applications increasingly require computational environments capable of processing large datasets, executing complex simulations, supporting collaborative workflows, and adapting to rapidly

changing workloads. Conventional engineering infrastructures are often designed around predetermined hardware capacity and localized computational resources. Although such environments can provide predictable performance, they may become inefficient when application requirements fluctuate significantly. Cloud computing introduces an alternative infrastructure model in which computing, storage, networking, and software resources can be provisioned dynamically according to technical requirements. This flexibility creates opportunities for engineering organizations to develop platforms that are more responsive to changing computational demands.

Cloud-based engineering platforms combine infrastructure services with software development, data processing, simulation, analytics, and collaboration capabilities. Instead of maintaining independent physical systems for every engineering activity, organizations can use virtualized and distributed environments to provide computational services through centralized or geographically distributed cloud infrastructure. Such platforms can support engineering design applications, computational modeling, technical analytics, simulation environments, software-intensive engineering processes, and industrial monitoring systems. The ability to provision resources on demand is particularly important for workloads that experience periodic computational peaks.

Scalability represents one of the most important characteristics of cloud-enabled technical applications. Vertical scalability involves increasing the computational capacity of an individual resource, whereas horizontal scalability involves distributing workloads across multiple computing instances. Cloud platforms can support both approaches through virtualization, automated provisioning, container orchestration, distributed databases, and elastic computing services. Application architectures designed specifically for cloud environments can therefore respond more effectively to workload changes than applications tightly coupled to fixed infrastructure.

Another important advantage of cloud-based engineering platforms is the integration of development and operational processes. Modern cloud environments support automated deployment, continuous integration, continuous delivery, infrastructure automation, monitoring, and centralized resource management. These capabilities allow engineering teams to reduce manual configuration and accelerate application deployment. Collaborative cloud environments can also provide geographically distributed teams with access to common datasets, computational services, development tools, and engineering workflows. Consequently, cloud platforms can contribute not only to technical scalability but also to organizational flexibility.

Despite these advantages, cloud-based engineering platforms introduce several technical and managerial challenges. Network latency, security vulnerabilities, data governance, vendor dependency, interoperability, unpredictable operating costs, and migration complexity can affect platform effectiveness. Engineering applications may also have specialized computational requirements that are not easily accommodated by generic cloud architectures. Therefore, scalability should be considered as a multidimensional property involving computational performance, data accessibility, reliability, security, cost efficiency, and operational sustainability. This study evaluates these dimensions and develops a conceptual understanding of how cloud-based engineering platforms can support scalable technical applications.

2. Objectives

The primary objective of this study is to examine how cloud-based engineering platforms support the development and operation of scalable technical applications. The study focuses on architectural characteristics, resource management approaches, deployment models, and scalability mechanisms that

enable engineering applications to respond to changing workloads while maintaining acceptable levels of performance, reliability, and operational efficiency.

2.1 Evaluation of Cloud-Based Engineering Architectures

The first objective is to evaluate the architectural components that contribute to scalable cloud-based engineering platforms. Particular attention is given to virtualization, containers, distributed computing, cloud storage, service-oriented architectures, and automated orchestration. These components provide the technical foundation required to distribute workloads and dynamically provision resources according to application requirements.

The objective also includes examining how modular and cloud-native architectures influence application scalability. Engineering applications developed as loosely coupled services can independently scale specific functional components rather than increasing the capacity of the entire application. This approach can improve resource efficiency and provide greater flexibility when technical workloads change over time.

2.2 Assessment of Scalability and Resource Management

The second objective is to assess mechanisms used to manage computational resources within cloud-based engineering environments. This includes elastic provisioning, workload distribution, autoscaling, resource scheduling, container management, and performance monitoring. Effective resource management is necessary to prevent computational bottlenecks while avoiding unnecessary infrastructure consumption.

The study also considers the relationship between scalability and operational efficiency. Increasing resources does not automatically produce better engineering performance if application architecture, data movement, network communication, or workload scheduling remains inefficient. Therefore, scalable platforms must coordinate infrastructure resources with application-level requirements to achieve balanced performance and cost effectiveness.

3. Literature Review

3.1 Cloud Computing for Engineering Applications

Cloud computing has transformed conventional approaches to computational infrastructure by providing shared and dynamically configurable resources. Mell and Grance established a widely adopted conceptual framework for cloud computing based on characteristics such as on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. These characteristics provide a strong foundation for technical applications requiring flexible computational capacity.

Armbrust et al. identified elasticity, utility-based resource consumption, and large-scale infrastructure as important characteristics of cloud computing. For engineering applications, these characteristics are relevant because computational workloads can vary substantially between development, simulation, testing, and production stages. Cloud environments can therefore reduce dependence on permanently provisioned infrastructure while supporting temporary computational requirements.

3.2 Cloud-Native and Distributed Engineering Platforms

Cloud-native development has increasingly emphasized microservices, containers, automation, orchestration, and independently scalable application components. Burns et al. discussed the importance

of container orchestration in managing distributed application workloads, while Bernstein examined how cloud computing evolved toward more flexible infrastructure and service-oriented approaches.

Containerization is particularly useful for engineering platforms because it allows applications and their dependencies to be packaged into portable execution units. Kubernetes-based orchestration can further automate scheduling, scaling, service discovery, and workload recovery. Such mechanisms can provide engineering applications with greater deployment consistency across development, testing, and production environments.

3.3 Resource Optimization and Scalable Computing

Resource management is a central issue in cloud environments because large-scale distributed systems must continuously balance performance, capacity, and cost. Buyya et al. emphasized the importance of resource provisioning and cloud service management for achieving efficient utilization. Similar research has examined scheduling, workload distribution, virtualization, and dynamic resource allocation as mechanisms for improving cloud performance.

For engineering applications, optimization becomes more complex when workloads involve computational simulations, real-time analytics, large-scale modeling, or data-intensive processing. Such applications may require specialized combinations of CPU, memory, storage, GPU acceleration, and high-speed networking. Consequently, scalable engineering platforms must incorporate application-aware resource management rather than relying exclusively on generic infrastructure scaling.

4. Methodology

4.1 Research Design

This study uses a conceptual and analytical research design to examine cloud-based engineering platforms and their contribution to scalable technical applications. The methodology synthesizes established research on cloud computing, distributed systems, virtualization, containerization, resource management, and cloud-native architectures. The analysis focuses on identifying relationships between platform architecture and scalability outcomes rather than evaluating a single commercial cloud implementation.

The research framework considers scalability as a multidimensional construct involving computational elasticity, application performance, resource utilization, reliability, deployment flexibility, and operational efficiency. Technical characteristics of cloud platforms are examined in relation to their ability to support changing workloads. This approach provides a structured basis for comparing conventional infrastructure models with cloud-enabled engineering environments.

4.2 Architectural Analysis

The architectural analysis examines major components of cloud-based engineering platforms, including virtual machines, containers, orchestration systems, distributed storage, networking layers, APIs, monitoring services, and automation frameworks. These components are considered according to their contribution to resource flexibility, workload distribution, application portability, and operational control.

The analysis also examines cloud-native application structures in which engineering functions are divided into modular services. Such architectures can support independent scaling and fault isolation. The methodology therefore considers both infrastructure-level and application-level scalability because efficient technical applications require coordination between the two layers.

4.3 Scalability and Resource Assessment

The scalability assessment focuses on the mechanisms through which cloud platforms respond to changing technical workloads. Particular attention is given to horizontal scaling, vertical scaling, autoscaling, workload scheduling, resource provisioning, and container orchestration. These mechanisms are evaluated conceptually according to their ability to maintain application performance while controlling unnecessary resource consumption.

Key assessment dimensions include:

- Elastic resource provisioning
- Application workload distribution
- Container orchestration
- Computational resource utilization
- Storage scalability
- Network responsiveness
- Fault tolerance
- Deployment automation
- Security and access control
- Operational cost management

4.4 Analytical Framework

The analytical framework connects cloud platform characteristics with scalability outcomes. Infrastructure elasticity, automation, distributed processing, and modular application design are considered primary technical factors. Performance, reliability, flexibility, resource efficiency, and cost management are considered major outcomes. Challenges such as latency, interoperability, security, and vendor dependency are treated as moderating factors.

The framework enables a structured interpretation of how different cloud capabilities influence engineering application scalability. Rather than assuming that cloud migration automatically produces better performance, the study recognizes that scalability depends on appropriate application architecture, workload characteristics, data movement, and resource-management strategies.

5. Analysis

5.1 Cloud Architecture and Scalability

Cloud-based engineering platforms provide multiple mechanisms for scaling technical applications. Virtualization enables computational resources to be provisioned without requiring dedicated physical systems, while containers provide lightweight application-level isolation. Distributed storage allows data capacity to expand independently from computational infrastructure. Together, these mechanisms create an adaptable foundation for technical applications.

The scalability benefits become more significant when cloud resources are combined with automated orchestration. An orchestration platform can monitor application requirements and distribute workloads across available resources. This enables engineering applications to respond to changes in computational demand without extensive manual intervention.

Table 1: Major Cloud Platform Components Supporting Engineering Application Scalability

Cloud Component	Primary Function	Scalability Contribution
Virtual Machines	Provide isolated computing environments	Supports flexible computational provisioning
Containers	Package applications and dependencies	Enables lightweight and portable scaling
Container Orchestration	Manages distributed workloads	Automates deployment and scaling
Cloud Storage	Stores engineering data	Enables expandable data capacity
Load Balancing	Distributes application requests	Reduces workload concentration
APIs	Connect application services	Supports modular technical architectures
Monitoring Systems	Tracks application performance	Enables proactive resource management
Automation Tools	Automate deployment and infrastructure	Reduces manual operational processes

Interpretation: Table 1 demonstrates that scalability is supported through multiple interconnected platform components rather than through computing capacity alone. Virtual machines and containers provide execution flexibility, while orchestration and load balancing distribute workloads. Storage services support increasing data requirements, whereas monitoring and automation improve operational responsiveness. The combination of these components allows engineering platforms to respond to workload variation more efficiently than fixed infrastructure environments.

5.2 Resource Management and Application Performance

Resource management directly influences the performance of scalable technical applications. If resources are insufficient, engineering workloads may experience processing delays, service interruptions, or computational bottlenecks. Conversely, excessive resource allocation can increase operational costs without producing proportional performance improvements. Cloud platforms therefore require mechanisms that dynamically align resource availability with workload demand.

Autoscaling can improve this alignment by increasing or decreasing infrastructure resources according to predefined or intelligent policies. Workload scheduling can further improve utilization by distributing computational tasks among available resources. For engineering applications involving simulations or analytics, resource management may also consider specialized processing requirements such as high-memory workloads or accelerator-enabled computation.

Table 2: Scalability Mechanisms and Expected Engineering Outcomes

Scalability Mechanism	Engineering Application Benefit	Major Consideration
Horizontal Scaling	Supports increased concurrent workloads	Requires distributed application design
Vertical Scaling	Increases individual resource capacity	Limited by machine capacity
Autoscaling	Responds dynamically to demand	Requires accurate scaling policies
Load Balancing	Distributes application traffic	Depends on network efficiency
Container Orchestration	Automates workload placement	Adds operational complexity
Distributed Processing	Handles computationally intensive tasks	Requires coordination between nodes
Serverless Computing	Provides event-driven execution	May introduce execution constraints
Edge-Cloud Integration	Reduces latency for localized workloads	Requires distributed governance

Interpretation: Table 2 shows that different scalability mechanisms address different engineering requirements. Horizontal scaling is particularly suitable for distributed workloads, while vertical scaling can support applications that depend on larger individual computational resources. Autoscaling improves elasticity but requires carefully designed policies. Edge-cloud integration can reduce latency for geographically distributed technical applications, while serverless approaches can simplify event-driven workloads. Consequently, selecting an appropriate mechanism requires understanding application architecture and workload behavior.

5.3 Cloud Platforms for Technical Collaboration

Cloud-based engineering environments can also improve collaboration by providing shared computational and data resources. Distributed engineering teams can access common development environments, simulation resources, datasets, and technical services without maintaining identical local infrastructure. Version-controlled and centrally managed resources can support greater consistency across engineering workflows.

Cloud collaboration is particularly valuable for multidisciplinary projects involving software developers, system engineers, data specialists, and technical researchers. However, collaboration depends on appropriate identity management, access control, data governance, and communication infrastructure. Therefore, collaborative scalability should be addressed alongside computational scalability.

6. Discussion

6.1 Architectural Flexibility

Cloud-based engineering platforms provide substantial architectural flexibility because computing, storage, and application services can be configured independently. This flexibility enables organizations to adapt technical systems to changing requirements without replacing entire infrastructure environments. Microservices and container-based architectures can further improve flexibility by allowing individual components to scale independently.

However, architectural modularity also introduces additional communication paths and operational dependencies. Therefore, engineering teams must balance modularity with system complexity and ensure that distributed application components are designed for reliability, observability, and efficient communication.

6.2 Performance and Resource Efficiency

Scalability should not be evaluated solely by the amount of infrastructure that can be provisioned. Effective scalability requires resources to be allocated according to actual workload requirements. Cloud-based autoscaling, workload scheduling, and monitoring can improve utilization by dynamically adjusting infrastructure capacity.

However, poorly configured scaling policies may cause unnecessary resource consumption or delayed response to workload increases. Engineering applications with intensive simulations or large datasets may also face network and storage bottlenecks. Consequently, application-aware resource management is essential for achieving both performance and economic efficiency.

6.3 Security and Operational Governance

Security represents a fundamental consideration in cloud-based engineering environments because technical applications may process proprietary designs, industrial information, operational data, or sensitive research outputs. Identity management, encryption, access control, monitoring, and secure application development must therefore be integrated into platform architecture. Governance is equally important because distributed infrastructure can create complex responsibility boundaries.

Engineering organizations should establish clear policies for data management, resource provisioning, compliance, and incident response. Security and governance should be treated as continuous operational requirements rather than isolated implementation activities.

7. Challenges

7.1 Infrastructure and Interoperability Challenges

Cloud-based engineering platforms may encounter interoperability problems when applications depend on specialized software, proprietary interfaces, or infrastructure-specific configurations. Migrating legacy engineering applications to cloud environments can require substantial architectural modification. Differences between cloud providers may also complicate workload portability. Standardized APIs, containerization, infrastructure-as-code practices, and open technical interfaces can reduce these difficulties, but complete interoperability remains difficult for complex engineering systems.

7.2 Security and Data Management Challenges

Engineering platforms frequently process large datasets and technically valuable information. Protecting this information across distributed cloud environments requires strong identity controls, encryption,

monitoring, backup mechanisms, and governance policies. Misconfigured cloud resources can expose sensitive data, while distributed architectures can create additional security boundaries. Data residency and regulatory requirements may further constrain infrastructure choices. Consequently, security must be incorporated throughout the engineering application lifecycle.

7.3 Cost and Operational Complexity

Although cloud computing can reduce the need for dedicated infrastructure, uncontrolled resource consumption can generate significant operating costs. Over-provisioned instances, unnecessary storage, inefficient data transfer, and poorly configured services can reduce economic benefits. In addition, cloud-native environments may require specialized skills in orchestration, automation, monitoring, and distributed systems. Organizations therefore need financial monitoring, resource governance, and workforce development to maintain sustainable cloud operations.

8. Future Research

8.1 AI-Driven Cloud Resource Management

Future research can investigate artificial intelligence techniques for predicting engineering workloads and dynamically optimizing cloud resources. Machine learning models could analyze historical workload patterns, application performance indicators, and infrastructure utilization to anticipate resource requirements.

AI-driven resource management may enable cloud platforms to provision capacity before performance degradation occurs. Research should also examine explainability, reliability, and operational safety because automated infrastructure decisions can affect critical technical applications.

8.2 Edge-Cloud Engineering Architectures

The integration of edge computing with cloud platforms represents another important research direction. Engineering applications involving industrial monitoring, robotics, autonomous systems, and real-time control may require low-latency processing that cannot always be achieved through centralized cloud infrastructure. Hybrid edge-cloud architectures can distribute workloads according to latency, computational, and data requirements.

Future studies should develop adaptive workload-placement strategies capable of dynamically determining whether technical tasks should execute at the edge, in regional infrastructure, or within centralized cloud environments.

8.3 Autonomous and Sustainable Cloud Engineering

Future cloud engineering research should increasingly consider autonomous infrastructure management and environmental sustainability. Intelligent platforms could automatically optimize workload placement, resource utilization, storage policies, and computational scheduling according to performance and energy objectives.

Research can also investigate carbon-aware engineering applications that shift workloads toward more efficient infrastructure when technically feasible. Combining autonomous management with sustainability objectives could produce cloud platforms that are not only scalable but also economically and environmentally responsible.

9. Limitations

This study is primarily conceptual and analytical rather than based on a large-scale experimental deployment. Consequently, the findings describe expected relationships between cloud platform capabilities and application scalability rather than providing universally applicable performance measurements. Actual outcomes may vary according to cloud provider, application architecture, workload type, network infrastructure, and organizational practices.

The study also does not evaluate a specific commercial cloud environment or benchmark individual cloud services. Engineering applications differ considerably in their computational, storage, latency, and security requirements, which limits the ability to generalize a single scalability model across all technical domains. Future empirical research should therefore validate the proposed analytical relationships through real-world implementations and controlled performance experiments.

10. Conclusion

Cloud-based engineering platforms provide a flexible technological foundation for developing scalable technical applications. Through virtualization, containerization, distributed processing, cloud storage, automated deployment, orchestration, and elastic resource allocation, these platforms can support changing computational requirements more effectively than many fixed infrastructure environments. Their benefits extend beyond computational scalability by improving deployment flexibility, collaborative engineering, resource utilization, and operational responsiveness.

The analysis demonstrates that scalability depends on the interaction between infrastructure capabilities and application architecture. Horizontal scaling, autoscaling, load balancing, distributed processing, and container orchestration can improve application responsiveness when they are appropriately aligned with workload characteristics. However, security, interoperability, network latency, operational complexity, and cost management remain important challenges. Cloud adoption should therefore be approached as an architectural transformation rather than simply as infrastructure migration.

Future cloud-based engineering platforms are likely to become increasingly intelligent, distributed, and autonomous. AI-assisted resource management, edge-cloud integration, serverless computing, digital engineering environments, and sustainability-aware infrastructure can expand the capabilities of technical applications. By combining scalable architectures with effective governance, security, and resource optimization, organizations can establish cloud engineering platforms capable of supporting increasingly complex and dynamic technical requirements.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. <https://doi.org/10.1145/1721654.1721672>
2. Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2011). *Cloud computing: Principles and paradigms*. Wiley.
3. Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. National Institute of Standards and Technology.
4. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1, 7–18. <https://doi.org/10.1007/s13174-010-0007-6>

5. Marinescu, D. C. (2013). *Cloud computing: Theory and practice*. Morgan Kaufmann.
6. Erl, T., Puttini, R., & Mahmood, Z. (2013). *Cloud computing: Concepts, technology & architecture*. Prentice Hall.
7. Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). *Mastering cloud computing: Foundations and applications programming*. Morgan Kaufmann.
8. Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A. S., & Buyya, R. (2015). Big Data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, 79–80, 3–15. <https://doi.org/10.1016/j.jpdc.2014.08.003>
9. Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, 79, 849–861. <https://doi.org/10.1016/j.future.2017.09.020>
10. Singh, S., & Chana, I. (2016). Cloud resource scheduling: A comprehensive survey. *The Journal of Supercomputing*, 72, 2113–2142. <https://doi.org/10.1007/s11227-015-1630-2>
11. Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5), 755–768. <https://doi.org/10.1016/j.future.2011.04.017>
12. Jennings, B., & Stadler, R. (2015). Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, 23, 567–619. <https://doi.org/10.1007/s10922-014-9307-7>
13. Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2016). Challenges and opportunities in edge computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 20–26). IEEE. <https://doi.org/10.1109/SmartCloud.2016.18>
14. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39. <https://doi.org/10.1109/MC.2017.9>
15. Kratzke, N., & Quint, P.-C. (2017). Understanding cloud-native applications after 10 years of cloud computing—A systematic mapping study. *Journal of Systems and Software*, 126, 1–16. <https://doi.org/10.1016/j.jss.2017.01.001>
16. Adzic, G., & Chatley, R. (2017). Serverless computing: Economic and architectural impact. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 884–889). ACM. <https://doi.org/10.1145/3106237.3117767>
17. Shafiei, H., Khonsari, A., & Mousavi, P. (2022). Serverless computing: A survey of opportunities, challenges, and applications. *ACM Computing Surveys*, 54(11s), Article 239, 1–32. <https://doi.org/10.1145/3510611>
18. Li, Z., Guo, L., Cheng, J., Chen, Q., He, B., & Guo, M. (2022). The serverless computing survey: A technical primer for design architecture. *ACM Computing Surveys*, 54(10s), Article 220, 1–34. <https://doi.org/10.1145/3508360>
19. Deng, S., Zhao, H., Huang, B., Zhang, C., Chen, F., Deng, Y., Yin, J., Dustdar, S., & Zomaya, A. Y. (2024). Cloud-native computing: A survey from the perspective of services. *Proceedings of the IEEE*, 112(1), 12–46. <https://doi.org/10.1109/JPROC.2024.3353855>
20. Vhatkar, K. N., & Bhole, G. P. (2021). A comprehensive survey on container resource allocation approaches in cloud computing: State-of-the-art and research challenges. *Web Intelligence*, 19(4), 295–316. <https://doi.org/10.3233/WEB-210474>