Journal of Frontiers in Multidisciplinary Research

# Improving Healthcare Data Intelligence through Custom NLP Pipelines and Fast API Micro services

**Oyejide Timothy Odofin [1*], Bolaji Iyanu Adekunle [2], Ejielo Ogbuefi [3], Jeffrey Chidera Ogeawuchi [4], Oluwasanmi Segun Adanigbo [5], Toluwase Peter Gbenle [6]**

[1] DXC Technology, Poland
[2] Department of Data Science, University of Salford, UK
[3] University of Massachusetts amherst And Soodle Technology
[4] Megacode Company, Dallas Texas. USA
[5] Remis Limited, Lagos, Nigeria
[6] Soft Switch, Roswell, Georgia, USA

* Corresponding Author: **Oyejide Timothy Odofin**

## Article Info

**E-ISSN:** 3050-9726
**P-ISSN:** 3050-9718
**Volume:** 04
**Issue:** 01
**January-June** 2023
**Received:** 14-03-2023
**Accepted:** 16-04-2023
**Published:** 17-05-2023
**Page No:** 390-397

**Abstract**
The healthcare industry is increasingly relying on data-driven insights for improving patient care and clinical decision-making. However, the complexity of healthcare data, particularly unstructured clinical notes and electronic health records (EHRs), presents significant challenges for traditional data processing systems. This paper introduces a novel approach that leverages custom Natural Language Processing (NLP) pipelines, integrated with FastAPI microservices, to enhance healthcare data intelligence. By utilizing domain-specific models like ClinicalBERT, the system efficiently processes unstructured text, extracting critical medical information such as diagnoses, symptoms, and treatments, and enabling real-time decision-making. The proposed architecture offers scalability, modularity, and low-latency performance, addressing the shortcomings of legacy systems. Furthermore, containerization using Docker and continuous integration/continuous deployment (CI/CD) pipelines ensure seamless deployment and system maintenance. Performance metrics, including latency, throughput, and NLP accuracy, are evaluated to demonstrate the efficacy of the system across various healthcare datasets. A real-world use case, such as clinical note triage, is presented to illustrate the practical impact of the system. Despite its promising results, limitations such as model generalizability and multilingual support are discussed, with directions for future research aimed at expanding the system's applicability. This paper contributes to the field of healthcare data intelligence by offering a scalable, flexible solution that improves clinical decision support and patient outcomes.

DOI: **https://doi.org/10.54660/.JFMR.2023.4.1.390-397**

**Keywords:** Healthcare Data Intelligence, Natural Language Processing (NLP), FastAPI Microservices, Clinical Decision Support, Electronic Health Records (EHRs), Real-Time Data Processing

## 1. Introduction
### 1.1 Background and Context
The modern healthcare ecosystem generates an immense volume of data daily, including patient records, diagnostic reports, clinical notes, prescriptions, and laboratory results [1, 2]. Much of this data is unstructured and exists in free-text formats, making it difficult to process using traditional data management systems [3, 4].

The heterogeneity and variability of language used in medical documentation add further complexity, particularly when clinical terms, abbreviations, and contextual nuances are present. Efficient extraction of relevant information from such data is essential for timely decision-making and improving patient outcomes [5-7].

Real-time processing of healthcare data is increasingly crucial, particularly in emergency care settings, remote diagnostics, and population health management. However, legacy systems are often ill-equipped to handle the velocity and variability of incoming data [8]. They also tend to lack the flexibility to adapt to evolving healthcare protocols and terminologies. This situation highlights the urgent need for intelligent systems that can parse unstructured content and convert it into structured, actionable insights [9, 10].

In response to these challenges, advanced natural language processing has emerged as a key enabler of health data intelligence. These technologies enable automated interpretation of clinical narratives, identification of patient conditions, and categorization of symptoms and treatments. However, such solutions must be designed to operate in dynamic, high-stakes environments [11-13]. They require not only linguistic accuracy but also architectural agility to support secure and efficient deployment within distributed healthcare infrastructures. This context forms the foundational landscape for exploring custom-built, real-time NLP solutions [14, 15].

### 1.2 Research Problem and Motivation

Despite recent advances in computational linguistics and machine learning, many existing healthcare NLP applications remain constrained by limited adaptability and scalability. Off-the-shelf models often perform poorly in clinical contexts because they are trained on general-domain corpora and fail to capture domain-specific terminologies. Additionally, integrating such models into real-time healthcare systems remains a technical challenge due to their computational overhead and lack of modularity. These constraints hinder the full realization of intelligent, responsive, and secure healthcare analytics platforms.

Moreover, healthcare environments demand strict compliance with regulatory and ethical standards. Data privacy, latency, and interoperability are critical requirements that cannot be overlooked. Unfortunately, many commercial and open-source NLP tools are not designed with these constraints in mind. Furthermore, healthcare institutions often operate in siloed IT environments with minimal capacity for end-to-end data orchestration, making it difficult to embed NLP solutions that require high availability and low latency.

Given this landscape, there is a pressing need for custom-built NLP pipelines that are not only tailored to the healthcare domain but also optimized for integration into modern microservices architectures. Such solutions should facilitate efficient deployment, real-time inference, and secure communication across systems. This research is motivated by the opportunity to bridge these critical gaps by leveraging domain-specific NLP models and lightweight service frameworks that can be readily adopted within diverse healthcare ecosystems.

### 1.3 Objectives and Contribution

The primary objective of this paper is to develop and evaluate a modular framework that enhances healthcare data intelligence through the integration of domain-specific NLP pipelines and lightweight service-based architectures. The solution aims to support real-time data ingestion, parsing, and classification of unstructured medical content. It also seeks to enable efficient deployment in cloud and edge environments while ensuring compliance with data governance standards. The core premise is that a tightly coupled design of custom NLP components and microservices can yield a robust, scalable, and adaptable analytics infrastructure.

To achieve this objective, the study proposes a multi-tiered solution architecture that combines fine-tuned clinical language models with a high-performance web framework for service orchestration. The use of asynchronous I/O, dependency injection, and auto-generated documentation within the chosen microservice environment allows for seamless scalability and ease of integration into existing health IT systems. Additionally, emphasis is placed on performance benchmarking, deployment automation, and API-based interoperability.

The key contributions of this paper include the design of a healthcare-specific NLP pipeline capable of handling entity recognition, context interpretation, and summarization tasks, and its deployment through a microservice framework that supports asynchronous processing and secure data exchange. The paper further contributes by presenting a detailed evaluation of system performance, real-world use cases, and future scalability prospects. Together, these components aim to advance the state-of-the-art in healthcare data intelligence by offering a customizable, efficient, and deployable solution architecture.

## 2. Related Work

### 2.1 NLP in Healthcare Data Processing

Natural language processing has a long-standing history in healthcare, with early implementations focused on rule-based systems and keyword matching for extracting data from clinical documentation [16, 17]. These systems, while useful for narrow applications, lacked contextual understanding and struggled with the ambiguity of medical language [18]. They often required substantial manual effort for rule creation and maintenance, making them unsustainable for broader clinical deployments. Despite these limitations, such techniques laid the groundwork for the automation of data abstraction in electronic health records [19-21].

Modern NLP approaches leverage deep learning, particularly transformer-based architectures like BERT and its domain-specific variants such as BioBERT and ClinicalBERT. These models have significantly improved performance in entity recognition, relation extraction, and text classification tasks within healthcare. They can capture semantic nuances and contextual relationships, enabling more accurate parsing of complex clinical narratives. Their pretraining on biomedical corpora allows them to outperform general-purpose models when applied to patient records and diagnostic summaries [22-24].

Despite these advancements, challenges persist in deploying NLP at scale in real-world healthcare settings. Variability in documentation practices, lack of annotated data, and the need for high accuracy in life-critical scenarios continue to hinder widespread adoption [25, 26]. Moreover, the computational demands of modern models require infrastructure that many healthcare systems lack. These factors underscore the importance of custom-built solutions tailored to specific clinical domains and integrated with high-performance serving architectures to meet operational requirements [27].

## 2.2 Microservice Architecture in Health IT

Microservice architectures have emerged as a transformative paradigm in health information technology, offering a modular approach to application development and deployment [28, 29]. Unlike monolithic systems, microservices enable independent development, testing, and scaling of application components, which is particularly beneficial in healthcare where systems must adapt to rapid regulatory, technological, and clinical changes [30, 31]. This architectural style supports greater flexibility, resilience, and maintainability in complex software ecosystems [32, 33].

FastAPI has gained significant traction in this domain due to its speed, ease of use, and support for asynchronous processing. It allows developers to build RESTful APIs with automatic documentation generation, type checking, and dependency injection [32, 34, 35]. These features reduce development time and improve code maintainability. In healthcare, FastAPI is increasingly used to expose machine learning models, process patient data, and serve decision support tools with minimal latency and high throughput—qualities essential for real-time healthcare applications [33, 36].

The decoupled nature of microservices aligns well with health IT needs, such as interoperability, privacy, and continuous integration. Services can be deployed across cloud and on-premise environments, encapsulating sensitive data while maintaining compliance with regulatory standards [37-39]. Moreover, the scalability inherent in microservice designs allows systems to handle varying loads, such as spikes in patient data during public health crises. This adaptability makes them ideal for embedding AI and NLP tools into clinical workflows without disrupting existing systems [40-42].

## 2.3 Gaps in Existing Systems

Despite notable advances in both NLP and microservice-based architectures, existing systems often fall short in their ability to deliver robust, real-time healthcare intelligence [40, 43]. One major limitation is speed; many NLP solutions rely on batch processing or heavyweight inference engines that introduce latency unsuitable for time-sensitive clinical decisions. This hampers their usability in urgent care, diagnostics triage, or remote monitoring contexts where immediacy is critical [44, 45].

Customization is another pressing issue. Generic NLP platforms are seldom optimized for the linguistic peculiarities of medical documentation. While some domain-specific models exist, they are often rigid and not easily fine-tuned to local dialects, institutional documentation practices, or specialist language [46, 47]. This lack of adaptability reduces model effectiveness and limits deployment across diverse healthcare environments. Additionally, most pre-built solutions are black boxes, offering little transparency or control to system integrators or clinicians [48, 49].

Deployment constraints further hinder adoption. Many legacy systems are incompatible with the infrastructure demands of contemporary NLP models and service-oriented architectures. Regulatory compliance requirements like HIPAA or GDPR introduce further complexity, especially when handling identifiable patient data in cloud environments [50, 51]. Moreover, most solutions lack the interoperability and orchestration mechanisms needed to integrate seamlessly into electronic health record systems or hospital information systems. These gaps highlight the need for lightweight, modular, and configurable systems that are explicitly designed for clinical application and scalable deployment [52-54].

# 3. Methodology

## 3.1 Design of Custom NLP Pipelines

The design of the custom NLP pipeline is central to transforming unstructured healthcare data into structured, actionable insights. The first step in the pipeline is data preprocessing, which involves tokenization, removing stopwords, and normalizing text. Tokenization is particularly important in healthcare, where clinical terms, abbreviations, and medical jargon must be correctly parsed. Following this, specialized techniques such as Named Entity Recognition (NER) are applied to extract key clinical entities like diseases, medications, dosages, and procedures from the text. A domain-specific model like ClinicalBERT, pretrained on clinical data, is then used to improve the accuracy of these tasks [55, 56].

Entity recognition is followed by relationship extraction, which identifies connections between entities. For instance, it can link a medication to its prescribed dosage or a patient to a diagnosis [57, 58]. After these tasks, the extracted data is fed into a downstream analysis system for summarization or classification tasks, such as identifying high-risk patient categories or flagging urgent conditions. The entire pipeline is designed to handle large-scale data input, ensuring efficiency and minimal latency in real-time applications [59, 60].

Customizing the pipeline for healthcare involves tailoring models to specific subdomains, such as oncology or cardiology. This fine-tuning ensures higher accuracy and relevance of the extracted data, as general NLP models may not understand the nuances of medical language. Additionally, regular updates to the model are incorporated to adapt to evolving clinical practices and new medical terminologies [61, 62].

## 3.2 Integration with FastAPI Microservices

FastAPI is chosen as the microservice framework due to its speed, flexibility, and seamless integration with modern Python-based NLP libraries. Its asynchronous capabilities are crucial in healthcare environments, where systems need to process multiple patient records in parallel without delays. FastAPI's non-blocking I/O model ensures that while one request is being processed, others can be handled simultaneously, making it an ideal choice for real-time, high-throughput systems such as healthcare data platforms [63, 64].

The microservice architecture is designed with clear separation of concerns, where each service is responsible for a specific task in the NLP pipeline. For example, one service may be dedicated to entity recognition, while another handles entity linking or text summarization [65]. FastAPI allows for efficient routing between these services, enabling seamless communication and data flow. This modular approach ensures that the system remains maintainable and scalable, allowing new NLP modules or services to be added without disrupting existing functionality [66, 67].

FastAPI's support for automatic generation of OpenAPI documentation also streamlines integration with external systems and facilitates testing and validation. Performance optimization is achieved through FastAPI's support for asynchronous request handling and background task management. This allows for non-blocking NLP model inference, ensuring that real-time predictions can be made without waiting for lengthy processing times, which is critical in clinical settings [66].

### 3.3 Data Handling and Security Protocols

In healthcare, ensuring the privacy and security of sensitive patient data is paramount. The custom NLP pipeline and FastAPI integration adhere to strict data security protocols, with encryption and anonymization techniques applied at every stage of the data flow [68]. Data is encrypted in transit using SSL/TLS protocols to ensure secure communication between the client, microservices, and backend systems. At rest, patient data is stored in encrypted databases, and any personally identifiable information (PII) is anonymized or pseudonymized before processing, in compliance with HIPAA and other privacy regulations [69, 70].

The data flow between components is designed with secure APIs and role-based access control to limit access to sensitive data. Each microservice in the FastAPI architecture is configured with strict authentication mechanisms, such as OAuth2 with JWT tokens, to verify the identity of users and ensure that only authorized personnel can access certain resources. Additionally, services communicate through secure, encrypted channels to prevent unauthorized access or data leakage [71, 72].

To maintain HIPAA compliance, all patient data that passes through the NLP pipeline is anonymized, and no personally identifiable information is retained in the model outputs [73]. The system also incorporates regular security audits, logging, and data access monitoring to ensure compliance with regulatory requirements and to identify any potential security breaches quickly. This secure handling of healthcare data builds trust with stakeholders and ensures that sensitive patient information is protected throughout the processing pipeline [74, 75].

## 4. Implementation and Evaluation

### 4.1 System Architecture and Deployment

The system architecture for this solution employs a microservices-based approach, where individual services are responsible for specific NLP tasks such as entity recognition, relationship extraction, and text summarization [76, 77]. These services are containerized using Docker, which ensures that each component can be independently deployed, tested, and scaled without dependency conflicts. Containerization also enhances the portability of the system, allowing it to be deployed across different environments (cloud or on-premise) with minimal configuration changes [78, 79].

For deployment, continuous integration/continuous deployment (CI/CD) pipelines are utilized to automate the testing and deployment of updates to the microservices. These pipelines ensure that new code changes are validated, ensuring system stability and security before production deployment [80]. The use of CI/CD allows for rapid iterations and the ability to address emerging healthcare needs in real-time. The system is deployed in hybrid cloud environments, leveraging both cloud resources for scalability and on-premise infrastructure for compliance with data sovereignty regulations [81, 82].

This architecture enables easy scaling of individual services based on workload demands. For instance, the NLP service for entity recognition can be scaled up when processing large volumes of clinical notes, while other services can be scaled independently [83]. This flexibility ensures that the system can handle varying traffic loads, providing high availability and fault tolerance, which are critical in healthcare applications where downtime can impact patient care [84].

### 4.2 Performance Metrics and Testing

Performance testing is critical to ensuring that the system meets the high demands of real-time healthcare data processing. Key metrics such as latency, throughput, scalability, and NLP accuracy are evaluated throughout the system [85, 86]. Latency is measured to assess the time it takes from receiving a request to delivering a processed result. Given the real-time nature of healthcare applications, maintaining low latency is essential, especially for time-sensitive tasks like clinical decision support or emergency care triage [87-89].

Throughput is another important metric, which measures how many requests the system can handle concurrently. The system is tested under various load conditions to simulate peak usage scenarios, ensuring that it can handle large volumes of data without performance degradation [90]. Scalability testing further evaluates the system’s ability to grow and adapt to increased data volumes, especially during public health emergencies or periods of high patient activity [91, 92].

Additionally, the accuracy of the NLP models is measured across various healthcare datasets, including clinical notes and lab reports. Accuracy metrics such as precision, recall, and F1 score are used to evaluate the performance of named entity recognition and text classification tasks. These metrics are essential to ensure that the system can accurately extract and interpret medical data, which directly impacts clinical decision-making [93, 94].

To demonstrate the real-world applicability of the system, a focused use case is explored: clinical note triage. In this scenario, the system automatically analyzes incoming clinical notes and triages them based on urgency [95]. The NLP pipeline extracts relevant entities, such as symptoms, diagnoses, and procedures, and classifies the severity of the patient's condition. This information is then sent to healthcare providers in real time, allowing them to prioritize high-risk cases promptly [96, 97].

For this use case, anonymized clinical data is used to ensure patient confidentiality and regulatory compliance. The NLP model processes various types of clinical notes, ranging from simple checkups to detailed diagnostic summaries, to showcase the system’s ability to handle diverse clinical documentation formats [98, 99]. The results are evaluated based on how accurately and quickly the system can triage the notes, providing actionable insights that improve patient care workflows. Furthermore, the system can be extended to lab result summarization, where incoming lab reports are automatically parsed, and key findings such as abnormal test results are highlighted. This reduces manual effort and ensures faster communication of critical health information.

The use case demonstrates how custom NLP pipelines, integrated with FastAPI microservices, can enhance healthcare data intelligence and improve decision-making efficiency in clinical settings [69, 100].

## 5. Conclusion

This paper demonstrates how custom NLP pipelines, integrated with FastAPI microservices, significantly enhance healthcare data intelligence compared to legacy systems. The modular architecture allows for scalable, flexible deployments tailored to specific healthcare needs, ensuring that each microservice can be optimized for particular tasks, such as entity recognition or text summarization. Unlike traditional monolithic systems, this approach supports continuous updates, real-time processing, and better resource management, thereby reducing operational overhead and improving system efficiency.

One of the key improvements over legacy systems is the increased speed and accuracy of NLP-based data extraction. By leveraging domain-specific models such as ClinicalBERT, the system ensures high-quality entity recognition and relationship extraction, facilitating better decision-making in clinical workflows. Furthermore, the integration of FastAPI enables seamless microservices communication, ensuring low latency and high throughput for real-time healthcare applications.

Despite the promising results, several limitations remain that warrant further research and development. One major challenge is the generalizability of the NLP models. While domain-specific models like ClinicalBERT excel in clinical settings, they may not perform as well in other healthcare domains or with data from different regions or hospitals that use distinct terminology or documentation practices. The need for ongoing model retraining and fine-tuning is essential to address this issue and ensure broader applicability.

Another limitation is the lack of multilingual support. While the current system is optimized for English-language clinical notes, expanding the pipeline to support multiple languages would increase its global applicability. This challenge is particularly relevant in multicultural environments where patient data is often provided in various languages. The addition of multilingual models and translation capabilities will be a crucial area for future development. Finally, while the FastAPI microservices architecture offers scalability and modularity, managing and orchestrating a large number of services as the system grows can become complex. Advanced techniques in service orchestration and monitoring will be required to handle the increasing number of services without compromising performance or reliability.

**References**


1. Rehman A, Naz S, Razzak I. Leveraging big data analytics in healthcare enhancement: trends, challenges and opportunities. Multimedia Systems. 2022;28(4):1339–71.
2. Haleem A, Javaid M, Singh RP, Suman R. Medical 4.0 technologies for healthcare: Features, capabilities, and applications. Internet of Things and Cyber-Physical Systems. 2022;2:12–30.
3. Shafqat S, Kishwer S, Rasool RU, Qadir J, Amjad T, Ahmad HF. Big data analytics enhanced healthcare systems: a review. The Journal of Supercomputing. 2020;76:1754–99.
4. Bohr A, Memarzadeh K. Current healthcare, big data, and machine learning. In: Artificial Intelligence in Healthcare. Elsevier; 2020. p. 1–24.
5. Viswanadham N. Ecosystem model for healthcare platform. Sādhanā. 2021;46(4):188.
6. Dash S, Shakyawar SK, Sharma M, Kaushik S. Big data in healthcare: management, analysis and future prospects. Journal of Big Data. 2019;6(1):1–25.
7. Dimitrov DV. Medical internet of things and big data in healthcare. Healthcare Informatics Research. 2016;22(3):156–63.
8. Khang A, Rana G, Tailor R, Abdullayev V. Data-centric AI solutions and emerging technologies in the healthcare ecosystem. 2023.
9. Jagadeeswari V, Subramaniyaswamy V, Logesh R, Vijayakumar V. A study on medical internet of things and big data in personalized healthcare system. Health Information Science and Systems. 2018;6(1):14.
10. Tresp V, Overhage JM, Bundschus M, Rabizadeh S, Fasching PA, Yu S. Going digital: a survey on digitalization and large-scale data analytics in healthcare. Proceedings of the IEEE. 2016;104(11):2180–206.
11. Gonzalez-Hernandez G, Sarker A, O'Connor K, Savova G. Capturing the patient's perspective: a review of advances in natural language processing of health-related text. Yearbook of Medical Informatics. 2017;26(01):214–27.
12. Kothinti RR. Advancements in natural language processing for automated phenotyping and predictive analytics in oncology EHRS. Iconic Research and Engineering Journals. 2021;8(8):245–52.
13. Zhou B, Yang G, Shi Z, Ma S. Natural language processing for smart healthcare. IEEE Reviews in Biomedical Engineering. 2022;17:4–18.
14. Juhn Y, Liu H. Artificial intelligence approaches using natural language processing to advance EHR-based clinical research. Journal of Allergy and Clinical Immunology. 2020;145(2):463–9.
15. Wong A, Plasek JM, Montecalvo SP, Zhou L. Natural language processing and its implications for the future of medication safety: a narrative review of recent advances and challenges. Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy. 2018;38(8):822–41.
16. Najafabadipour M. Understanding clinical narratives in oncological domain. ETSI Informatica. 2021.
17. Demner-Fushman D, Elhadad N, Friedman C. Natural language processing for health-related texts. In: Biomedical Informatics: Computer Applications in Health Care and Biomedicine. Springer; 2021. p. 241–72.
18. Lin Z, *et al*. Medical visual question answering: A survey. Artificial Intelligence in Medicine. 2023;143:102611.
19. Alam F. Knowledge net: An automated clinical knowledge graph generation framework for evidence-based medicine. Oakland University; 2023.
20. Chang JZ. EMR-based computational phenotyping in multiple sclerosis incorporating natural language processing and machine learning. University of Rochester; 2022.
21. Hughes KS, Zhou J, Bao Y, Singh P, Wang J, Yin K. Natural language processing to facilitate breast cancer research and management. The Breast Journal. 2020;26(1):92–9.
22. Conway M, Hu M, Chapman WW. Recent advances in using natural language processing to address public health research questions using social media and consumer-generated data. Yearbook of Medical Informatics. 2019;28(01):208–17.
23. Michalski AA, *et al*. Supporting the diagnosis of Fabry disease using a natural language processing-based approach. Journal of Clinical Medicine.

2023;12(10):3599.
24. Raja K, Jonnalagadda S. Natural language processing and data mining for clinical text. Healthcare Data Analytics. 2015;36:219.
25. Yadav K, Sharma M, Michaelson J, Laxminarayanamma K, Yadav VK, Krishna O. BERT-driven natural language processing in electronic health records for improved clinical insights. In: 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI). IEEE; 2023. p. 1–6.
26. Bacco L. Exploring new technologies in healthcare: Advancing natural language processing. 2023.
27. Kalyan KS, Rajasekharan A, Sangeetha S. AMMU: A survey of transformer-based biomedical pretrained language models. Journal of Biomedical Informatics. 2022;126:103982.
28. Alshuqayran N. Static microservice architecture recovery using model-driven engineering. University of Brighton; 2020.
29. Fritzsch J, Bogner J, Wagner S, Zimmermann A. Microservices migration in industry: Intentions, strategies, and challenges. In: 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE; 2019. p. 481–90.
30. Prosper J. Microservices architecture for agile integration. 2019.
31. Kumar TV. Cloud-based core banking systems using microservices architecture. 2019.
32. Zimmermann O. Microservices tenets: Agile approach to service development and deployment. Computer Science-Research and Development. 2017;32:301–10.
33. Wolff E. Microservices: Flexible software architecture. Addison-Wesley Professional; 2016.
34. Bellemare A. Building event-driven microservices. O'Reilly Media; 2020.
35. Nadareishvili I, Mitra R, McLarty M, Amundsen M. Microservice architecture: Aligning principles, practices, and culture. O'Reilly Media, Inc.; 2016.
36. Salah T, Zemerly MJ, Yeun CY, Al-Qutayri M, Al-Hammadi Y. The evolution of distributed systems towards microservices architecture. In: 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST). IEEE; 2016. p. 318–25.
37. López MR, Spillner J. Towards quantifiable boundaries for elastic horizontal scaling of microservices. In: Companion Proceedings of the 10th International Conference on Utility and Cloud Computing. 2017. p. 35–40.
38. Stenroos K. Microservices in software development. 2019.
39. Cebeci K. Design of a queue-based microservices architecture and performance comparison with monolith architecture. Marmara Universitesi (Turkey); 2019.
40. Newman S. Monolith to microservices: Evolutionary patterns to transform your monolith. O'Reilly Media; 2019.
41. Tapia F, Mora MA, Fuertes W, Aules H, Flores E, Toulkeridis T. From monolithic systems to microservices: A comparative study of performance. Applied Sciences. 2020;10(17):5797.
42. Laisi A. A reference architecture for event-driven microservice systems in the public cloud. 2019.
43. Taibi D, Systä K. From monolithic systems to microservices: A decomposition framework based on process mining. In: International Conference on Cloud Computing and Services Science. SciTePress; 2019. p. 153–64.
44. Gias AU, Casale G, Woodside M. ATOM: Model-driven autoscaling for microservices. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE; 2019. p. 1994–2004.
45. Rossi F, Cardellini V, Presti FL. Hierarchical scaling of microservices in Kubernetes. In: 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS). IEEE; 2020. p. 28–37.
46. Hippchen B, Schneider M, Landerer I, Giessler P, Abeck S, Lavazza L. Methodology for splitting business capabilities into a microservice architecture: Design and maintenance using a domain-driven approach. In: The Fifth International Conference on Advances and Trends in Software. Valencia, Spain; 2019.
47. Hippchen B, Schneider M, Giessler P, Abeck S. Systematic application of domain-driven design for a business-driven microservice architecture. International Journal on Advances in Software. 2019;12(3&4).
48. Srirama SN, Adhikari M, Paul S. Application deployment using containers with auto-scaling for microservices in cloud environment. Journal of Network and Computer Applications. 2020;160:102629.
49. Liu G, Huang B, Liang Z, Qin M, Zhou H, Li Z. Microservices: Architecture, container, and challenges. In: 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE; 2020. p. 629–35.
50. Bjørndal N, *et al*. Migration from monolith to microservices: Benchmarking a case study. Tech. Rep.; 2020.
51. Huang K, Jumde P. Learn Kubernetes Security: Securely orchestrate, scale, and manage your microservices in Kubernetes deployments. Packt Publishing Ltd; 2020.
52. Jaramillo D, Nguyen DV, Smart R. Leveraging microservices architecture by using Docker technology. In: Proceedings of SoutheastCon 2016. IEEE; 2016. p. 1-5.
53. Khakame PW. Development of a scalable microservice architecture for web services using os-level virtualization. University of Nairobi; 2016.
54. Desai V, Koladia Y, Pansambal S. Microservices: architecture and technologies. International Journal of Research in Applied Science and Engineering Technology. 2020;8(10):679-686.
55. Silverman GM, *et al*. NLP methods for extraction of symptoms from unstructured data for use in prognostic COVID-19 analytic models. Journal of Artificial Intelligence Research. 2021;72:429-474.
56. Thatoi P, Choudhary R, Shiwlani A, Qureshi HA, Kumar S. Natural language processing (NLP) in the extraction of clinical information from electronic health records (EHRs) for cancer prognosis. International Journal. 2023;10(4):2676-2694.
57. Krishnan S, Wu E. Alphaclean: Automatic generation of data cleaning pipelines. arXiv preprint arXiv:1904.11827. 2019.
58. Pervaiz F, Vashistha A, Anderson R. Examining the challenges in development data pipeline. In: Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies; 2019. p. 13-21.

59. Manninen E. Implementing a continuous integration and delivery pipeline for a multitenant software application. 2019.
60. Chinamanagonda S. Enhancing CI/CD pipelines with advanced automation—continuous integration and delivery becoming mainstream. Journal of Innovative Technologies. 2020;3(1).
61. Wratten L, Wilm A, Göke J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. Nature Methods. 2021;18(10):1161-1168.
62. Debroy V, Miller S, Brimble L. Building lean continuous integration and delivery pipelines by applying DevOps principles: a case study at Varidesk. In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering; 2018. p. 851-856.
63. Tragura SJC. Building Python Microservices with FastAPI: Build secure, scalable, and structured Python microservices from design concepts to infrastructure. Packt Publishing Ltd; 2022.
64. Peralta JH. Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More. Simon and Schuster; 2023.
65. Chopra P, Chhapola A, Kaushik DS. Comparative analysis of optimizing AWS Inferentia with FastAPI and PyTorch models. International Journal of Creative Research Thoughts (IJCRT). 2022;10(2):e449-e463.
66. Karttunen J. Methods for API governance automation: managing interfaces in a microservice system. 2023.
67. Lahtinen S. Light-weight method for detecting API breakages in microservice architectures. 2022.
68. Gupta A, Lai A, Mozersky J, Ma X, Walsh H, DuBois JM. Enabling qualitative research data sharing using a natural language processing pipeline for deidentification: moving beyond HIPAA Safe Harbor identifiers. JAMIA Open. 2021;4(3):ooab069.
69. Sousa R. Big data and real-time knowledge discovery in healthcare institutions. 2023.
70. Al-Rumaim A, Pawar JD. Exploring the evolving landscape of API security challenges in the healthcare industry: a comprehensive review. In: 2023 16th International Conference on Security of Information and Networks (SIN). IEEE; 2023. p. 1-8.
71. Khattak WA, Rabbi F. Ethical considerations and challenges in the deployment of natural language processing systems in healthcare. International Journal of Applied Health Care Analytics. 2023;8(5):17-36.
72. Nowrozy R, Ahmed K, Wang H, Mcintosh T. Towards a universal privacy model for electronic health record systems: an ontology and machine learning approach. Informatics. 2023;10(3):60.
73. Sousa S, Kern R. How to keep text private? A systematic review of deep learning methods for privacy-preserving natural language processing. Artificial Intelligence Review. 2023;56(2):1427-1492.
74. Zuo Z, Watson M, Budgen D, Hall R, Kennelly C, Al Moubayed N. Data anonymization for pervasive health care: systematic literature mapping study. JMIR Medical Informatics. 2021;9(10):e29871.
75. Lim S. HIPAAway: developing software for de-identification and exploring bias in name detection. Massachusetts Institute of Technology; 2023.
76. Trabelsi I, *et al*. From legacy to microservices: a type-based approach for microservices identification using machine learning and semantic analysis. Journal of Software: Evolution and Process. 2023;35(10):e2503.
77. Adedugbe O. Development and evaluation of a holistic, cloud-driven and microservices-based architecture for automated semantic annotation of web documents. Staffordshire University; 2019.
78. Chandramouli R. Microservices-based application systems. NIST Special Publication. 2019;800(204):800-204.
79. Daoud M, El Mezouari A, Faci N, Benslimane D, Maamar Z, El Fazziki A. A multi-model based microservices identification approach. Journal of Systems Architecture. 2021;118:102200.
80. Li Z, Shang C, Wu J, Li Y. Microservice extraction based on knowledge graph from monolithic applications. Information and Software Technology. 2022;150:106992.
81. Grover V, Verma I, Rajagopalan P. Achieving Digital Transformation Using Hybrid Cloud: Design standardized next-generation applications for any infrastructure. Packt Publishing Ltd; 2023.
82. Oladoja T. Optimizing CI/CD in healthcare: techniques for streamlined. 2022.
83. Pais S, Cordeiro J, Jamil ML. NLP-based platform as a service: a brief review. Journal of Big Data. 2022;9(1):54.
84. Chintale P. DevOps Design Pattern: Implementing DevOps best practices for secure and reliable CI/CD pipeline (English Edition). Bpb Publications; 2023.
85. Taher NC, Mallat I, Agoulmine N, El-Mawass N. An IoT-cloud based solution for real-time and batch processing of big data: application in healthcare. In: 2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART). IEEE; 2019. p. 1-8.
86. Hartmann M, Hashmi US, Imran A. Edge computing in smart health care systems: review, challenges, and research directions. Transactions on Emerging Telecommunications Technologies. 2022;33(3):e3710.
87. Mishra L, Varma S. Performance evaluation of real-time stream processing systems for Internet of Things applications. Future Generation Computer Systems. 2020;113:207-217.
88. Sheta SV. A comprehensive analysis of real-time data processing architectures for high-throughput applications. 2022.
89. Mustafa A, Hameed N. Advanced computing techniques for real-time data processing and high-performance computing. Journal of Advanced Computing Systems. 2023;3(8):1-8.
90. Afshar M, *et al*. Deployment of real-time natural language processing and deep learning clinical decision support in the electronic health record: pipeline implementation for an opioid misuse screener in hospitalized adults. JMIR Medical Informatics. 2023;11:e44977.
91. Hasanpuri V, Diwaker C. Big data, bigger challenges: a comparative study of performance testing. In: 2023 Seventh International Conference on Image Information Processing (ICIIP). IEEE; 2023. p. 870-875.
92. Olayinka OH. Big data integration and real-time analytics for enhancing operational efficiency and market responsiveness. International Journal of Scientific Research Archive. 2021;4(1):280-296.

93. Chen W, Milosevic Z, Rabhi FA, Berry A. Real-time analytics: concepts, architectures, and ML/AI considerations. IEEE Access. 2023;11:71634-71657.
94. Sresth V, Nagavalli SP, Tiwari S. Optimizing data pipelines in advanced cloud computing: innovative approaches to large-scale data processing, analytics, and real-time optimization. International Journal of Research and Analytical Reviews. 2023;10:478-496.
95. Rabbani N, *et al*. A natural language processing model to identify confidential content in adolescent clinical notes. Applied Clinical Informatics. 2023;14(3):400-407.
96. Shah SM, Khan RA. Secondary use of electronic health record: opportunities and challenges. IEEE Access. 2020;8:136947-136965.
97. Lee J, *et al*. Prevalence of sensitive terms in clinical notes using natural language processing techniques: observational study. JMIR Medical Informatics. 2022;10(6):e38482.
98. Khizra S. Using natural language processing and machine learning for analyzing clinical notes in sickle cell disease patients. 2018.
99. Dalianis H. Clinical text mining: secondary use of electronic patient records. Springer Nature; 2018.
100. Weber F. Artificial Intelligence for Business Analytics: Algorithms, Platforms and Application Scenarios. Springer Nature; 2023.