

Kubernetes and Docker Containers: A Survey

Devang Bhatnagar
Bachelor of Technology (CSE)
Amity University Greater Noida
Greater Noida, India
devangbhatnagar12@gmail.com

Abstract—The application software was massive, immense, and a rigid not too long ago. They were kept in enormous blocks of data, lying lonesomely within chunks of silicon and steel, ensuring that they would remain there, resisting change and unable to leave their position. This presented a challenge for the company because it needed to act quickly. Therefore, it wasn't a big deal when virtual machines (VMs) became popular. After that, apps were no longer connected with these bits of devices, which made everything move quickly. But regrettably, these VMs are also extremely complicated. They disguise an entire system within a different system, yet this virtual machine is nonetheless quite complex and requires management. The manner in which software is created, delivered, and maintained has been revolutionised by containers. Docker and Kubernetes have become two of the most well-known containerization techniques on the market. In-depth analysis of the fundamental ideas, features, and architectural elements of Docker and Kubernetes containers is provided in this study's report. Along with showcasing their unique functions in the container shipping environment, it goes on to investigate the distinctions and resemblance between Docker and Kubernetes. Furthermore, the report examines actual applications and used cases that heavily rely on both Kubernetes and Docker containers in order to demonstrate upon their useful advantages. In general, this research intends to give readers a good grasp of Kubernetes and Docker containers, empowering them to choose wisely whether to embrace and use them.

Keywords— *docker, Kubernetes, containers, application, technology, containerization, orchestrating, device.*

I. INTRODUCTION

Nowadays, Docker is a popular and efficient container technology. By using it, you may "build, transport, and operate any app, anyplace." It has accomplished quite a lot in an astonishingly short amount of duration and is now regarded as an accepted method for resolving one of the most expensive aspects of software: deployment.

Currently, Docker is the leading service for software containers worldwide. It is a container-based technology that is open source. When working on code with coworkers, it helps engineers avoid "functions at my device" issues. In order to achieve an accurate volume, operators use the Docker platform to run and manage apps concurrently in separate containers. The businesses utilise this sort of technology to create quick and simple computer software transport routes to release fresh capabilities for the Windows Server and Linux Server apps more quickly, privately, reliably, and effectively.

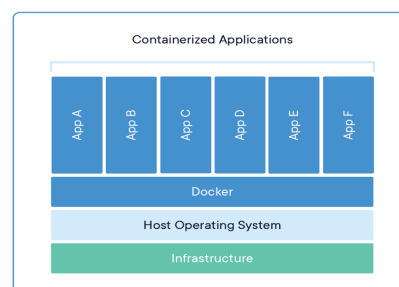
Through the aid of Docker containers, you can divide up your massive applications into a number of simpler "small apps" and confine each one in its own container. Using the most

appropriate equipment for their task, numerous small teams would be capable to collaborate on their own applications in this fashion (you are not need to use the identical technologies as the full programme). Additionally, the concept of micro service delivery as a whole and this have a tremendous compatibility. Similar to Docker, a different platform is the Kubernetes. It belongs to an open sourced platform that is utilised to administer apps and software that have been contained across a number of hosts, providing the fundamental tools for application installation, expanding and administration. Greek helmsman or pilot is the literal translation of the word Kubernetes. Those eight words that lie between the "K" & the "s," respectively, are what lead to the name "K8s." In 2014, Google made the Kubernetes project open-source.

A. What are Containers ?

All of the components required for a small piece of software in order to operate and work is housed in separate, isolated containers. Unlike virtual machines (VMs), containers only include the parameters and frameworks needed to ensure that the software works as intended. This results in effective, not heavy-weight, independent systems and guarantees that application software will operate consistently, wherever it is installed.

It allows the programmer to package a programme with all of its parts, including the whole stack of code it runs on top of and any dependencies that are associated with it. The programme has all it needs to operate within this particular containers under a deserted setting.



II. HISTORY

'Andrea Luzzardi' & 'Francois-Xavier Bourlet' were two developers who worked for DotCloud Company and initiated an internal project called Docker with Solomon Hykes in France. Additionally, Jeff Lindsay participated in the project as a separate collaboration. The DotCloud patented tech grew

into Docker, and when it did, it turned so well-known that DotCloud changed its name to Docker Inc.

In March 2013 as a project based on open source, Docker was first made available. By replacing LXC as the standard operating surroundings with the company's own libcontainer library created in the Go programming language, the Docker platform replaced LXC on March 13th, 2014, through the release of version 0.9. Following the June 2014 announcement of Docker 1.0, Docker Inc. decided that the corresponding container technology was sufficiently advanced and ready for deployment in generation, with businesses and partners providing options for sponsored service. Docker was first made accessible as a freely available project in March 2013. This initiative is still developing quickly, as evidenced by changes that are released on a regular basis and which add new attributes and fix bugs whenever they are discovered. Despite frequent changes, the endeavour has succeeded in separating "ship" through "run," allowing pictures generated by any of the edition of the Docker to be usable about any subsequent version (with reverse and forward compliance). This creates a solid basis for Docker utilisation. The project had a whopping twenty-five thousand GitHub shines as of October 24th, 2015, more than six thousand eight hundred forks, and over a thousand supporters, making it the twentieth highly rated GitHub repository.

Brendan Burns, Craig McLuckie, and Joe Beda founded Kubernetes. They were quickly connected to various other Google developers involving Tim Hockin and Brian Grant. Google originally announced it during the middle of 2014. The project's creation and architecture were heavily influenced by the company's Borg a system, and many of the initiative's most prominent participants had previously engaged on Borg. Kubernetes' initial name at Google was "Project Seven," which was an apparent allusion to a 'friendlier' Borg from the television series Star Trek. The 7 spokes on the design's round wheel, which also appears on the Kubernetes emblem, stand in for the name.

Kubernetes latest v1.0 was launched on the twenty-first of July 2015. In addition to that, the Google and the Foundation for Linux merged and created the "Cloud Native Computing Foundation (CNCF)," with Kubernetes being promoted as its primary tech.

III. DOCKER'S PRESPECTIVE

Currently, Docker, or Dock is arguably the fastest-evolving software initiative ever. Just lately, Docker had received an additional seven thousand forks in the previous year plus earned more than thirty k GitHub ratings. It has recognised a remarkable volume of pull requests from major corporations like the International Business Machines Red Hat, the Cisco Corporation, Microsoft, Google, even VMware, Inc. "Docker has reached this essential number by addressing a crucial demand for numerous software organisations: the capacity to develop applications in an accessible and adaptable manner and afterwards execute it repeatedly and dependably in various circumstances. You can create, distribute, and execute portable programmes without having to study a

brand-new programming language, invest in pricey devices, or perform a lot of deployment or config. Future potential for it is enormous. The manner in which we create and distribute software applications as well as the method by which we deliver technical support are both being completely transformed by this latest innovation.

IV. REAL WORLD APPLICATION AND USED CASES

The firm Docker Inc. has created a clear roadmap for development regarding the fundamental features (named "libcontainer"), cross service handling (called "libswarm"), and interaction between containers (called "libchan"). The corporation has previously shown a desire to ingest its personal surroundings in the intervening time with the purchase of "Orchard Labs." There is a distinct focus of technological executive leadership for the corporation and the initiative with CTO Solomon Hykes acting as a benign ruler. Since the project's inception, it has demonstrated the potential to advance quickly just by using its own outputs, and currently there aren't any indications that its capacity is waning. "Several entrepreneurs are examining the functionality matrices for VMware's system of ESX/vSphere platforms over more than ten years ago in order to observe the differences (and possibilities) among corporate ambitions fueled by the increasing number of VMs and the current Docker environment. Currently underrepresented by the current Docker environment, these areas include networking purposes, preservation, and extremely fine release maintenance (for the data inside of containers), and they create potential possibilities for both new businesses and established players. The 'run' component of Docker, which somewhat distinguishes amongst virtual machines and containers, is expected to lose significance as time passes, that will draw consideration to both the 'build' and 'ship' components.

A. Case Studies Utilising Both Containers

The two systems are frequently used together in the following situations:

a. *Arrangement and Delivery of Applications:* Apps and their associated dependencies have been placed into movable containers using Docker. Kubernetes could subsequently execute and oversee such containers. The orchestrating and expanding tools required to execute applications in containers dependably and effectively are provided by Kubernetes.

b. *Structure for Microservices:* With the help of Docker, miniature services may be separately created, executed, and expanded by being encapsulated into separate containers. Those microservices are deployed and managed via Kubernetes, which offers functions including discovering services, balancing the load, and autonomous scaling.

c. *Implementations of Hybrid Cloud:* Hybrid cloud infrastructures, which combine privately owned infrastructure with commercial cloud providers, are frequently adopted by businesses. The applications may be packed and executed uniformly throughout different domains thanks to Docker and Kubernetes, offering mobility and being portable.

d. *Continuous Deployment and Integration*: A important component of CI and CD (Continuous Integration and other one Continuous Deployment) pathways is the use of Docker and Kubernetes. In contrast to Kubernetes, which oversees the setup and scalability of app containers within an organised way, Docker containers provide stable surroundings for installation and testing.

e. *Analytics for Big Data*: Large-scale dataset handling and evaluation are frequent components of big data apps. To install and operate different parts of big data architectures like Apache Hadoop, Apache Spark, and Apache Kafka, containers which include Docker and Kubernetes are employed. These distributed platforms can function in accurate and repeatable contexts thanks to containers, while Kubernetes manages the orchestration and scalability of containers, enabling quick data processing and analysis of big datasets.

These can be considered merely a few instances of the many different uses and applications for which Kubernetes and Docker containers are put to use. The containerization solutions are adaptable and flexible enough to be used in a variety of circumstances and sectors, allowing businesses to improve the speed, scalability, and rapidity of their software creation and installation procedures.

f. *Internet of Things(IOT)*: IoT applications frequently call for the deployment and administration of several devices on the edge and detection devices. Internet of Things services and programmes are packaged and deployed using Docker containers, which are portable and flexible. These packaged Internet of Things (IoT) applications can be managed and scaled with Kubernetes, which enables effective information handling and current insights at the the brink.

V. COMPARISON OF DOCKER AND KUBERNETES

Addressing the distinctive features of Docker and Kubernetes is crucial since they play various functions in the containerization environment. The following are some aspects that set both of these systems apart:

A. Objective

Creating, the packing, and supplying containerized software are the main purposes of the Docker platform. It concentrates on developing small, transportable containers which hold a programme and all of its dependents. In contrast, Kubernetes is a free platform for container orchestration that optimises the installation, expansion, and administration of container-based apps amongst computer clusters.

B. Features

Docker offers resources for handling, deploying, and building containers. It has functionality including collaborating potential, picture administration, and container isolation. Software Developers may create containers with Docker which can operate in a variety of circumstances and systems. On the different side, Kubernetes is in charge of

managing the full lifespan of units and orchestrating the use of containers. It optimises processes like roll-out, scaling, balance of loads, and healing itself, eventually Kubernetes offers cutting-edge functionalities.

C. Structure

The Docker daemon, which is in charge of creating and managing containers, interacts alongside the client via an architecture based on client-server communication. A multilayered picture structure used by Docker allows for effective sharing of pictures while requiring less storage. A master worker structure is used by Kubernetes. The entire cluster is overseen by the master node, which also handles container schedule and orchestrating. The containers are managed and communicated therewith by nodes that work for them, also referred to as minions. Kubernetes provides a declaratory technique to specify the framework's ideal state and makes reassured that the real state corresponds to the ideal state.

VI. TECHNOLOGY OF THE CONTAINER

What additional obstacles are there, and how does it integrate into the cloud?

A container system for management is Docker. It essentially optimises the creation of containers for actively operating applications or their components. A group of APIs that can be created from layouts or instructions are used to control containers. Any the operating system that supports container-like segmentation can be used to build container-based structures, however Docker only encourages Linux container tools. Therefore, Docker containers operate apps and technologies that are based on linux specifically. It is also able to operate on a virtual machine (VM) that is powered by another operating system, but in order to process the containers, the VM must have a guest operating system of LINUX.

Despite the fact that non-Linux host operating systems can be configured to execute Docker containers, only Linux applications can be launched in them, thus employing Docker on non-Linux platforms is challenging. Users who have substantial needs for servers running Windows and want to integrate Linux-based apps into them may find value in the containers who operate on Windows operating systems. Businesses may run containers on open-source cloud virtual machines in their information locations, and the readily accessible Docker utilities can assist in the deployment of container-based portions, assist processes, and make it easier to employ hybrid clouds by providing simple backup support. For cloud services that are public, the benefits of VM-based computerization include privacy.

VII. BENEFITS AND CHALLENGES

A. Advantages of Docker and Kubernetes containers

a. Simple and Straightforward Configuration: Similar to virtual machines, Docker provides a unique environment in which you can execute codes, however unlike virtual machinery, there is no requirement for installation of a computer's operating system. To segregate the place of operation for several apps so they can be utilised from prototype to release without additionally any changes, adopt a container. These days, Docker is supported by all major sectors involved in "Infrastructure as a Service" and "Platform as a Service."

b. Ecosystem That Is Reliable: One of the main issues, creating a stable infrastructure for controlling code across various machines, is solved by Docker. The programme may pass across several machines as it moves from design to execution throughout app advancement. Docker provides a consistent environment on all of them, making it considerably simpler and faster to build and distribute code.

c. Isolating Apps Quickly & Security: Docker facilitates and minimises the expense of software encapsulation. Different applications can be segregated using containers, allowing them to reside in unique configuration surroundings independent of those used by apps launched within different containers. By imposing segmentation of the network, accessibility limits, and containers safety standards, Kubernetes improves security. Additionally, containers allow for rapid and simple programme restore in the event of security flaws or problems.

B. Limitations and challenges associated with Docker and Kubernetes

a. Complicatedness and Learning curve: For operations teams and development teams, using Kubernetes and Docker containers involves an initial learning curve. It can be difficult at first to comprehend the fundamentals of containerization, how to create Docker files, how to orchestrate containers, and how to set up Kubernetes.

b. Complications in Connectivity and Storage: In systems using containers, connectivity and memory setup might be complicated. Proper preparation and design are needed to set up container connections, service identification, and load balancing. In dynamic container systems, handling storage for state-driven apps can be difficult.

c. Cost of Resources: Although running a lot of containers continues to require a lot of facilities, containers are lighter than conventional virtualization. To prevent shortages of resources and productivity depletion, appropriate resource strategy and supervision are crucial.

Docker is useful, but it can't completely replace virtual machines (VMs). There are just a handful apps that won't be installed successfully within containers. Micro-services can benefit from Docker. Organisations may maximise the advantages of containerization while minimising possible obstacles by overcoming issues through effective preparation, coaching, utilising suitable tools, and following best practises.

VIII. CONCLUSION

Currently, Docker and container technologies are widely accepted and are generating significant consumer demand across the globe. Traditional merchants were compelled to offer a minimum of one initial container commodity as a result, but it will be important to consider how these innovations may be simply included and meet technical criteria in the years to come.

I have provided a thorough yet condensed analysis of the Docker-related container technologies throughout my research project. It has been dealt with in terms of many different applications, advantages, and uses. Identical opened sourced application container "Kubernetes" is a distinct technology that is used for improved application administration and implementation. This study primarily concentrated on explaining Docker. I've listed the positives and a few drawbacks, but in all honesty, Docker is a significantly superior technology that comes with many intuitive features.

Future developments, wider acceptance, and interaction with cutting-edge technology are all possible with Kubernetes and Docker containers. A more simplified and effective atmosphere for engineers and data scientists will be provided through greater cooperation of Kubernetes and Docker containers alongside artificial intelligence (AI) and machine learning algorithms.

In a nutshell the development and deployment of apps has been revolutionised by Kubernetes and Docker containers. Organisations may adopt current software building practises, gain expansion and flexibility, and remain at the leading edge of advancements in technology in the containerizing environment by utilising their expertise.

REFERENCES

- [1] P. K. Kushwaha and M. Kumaresan, "Machine learning algorithm in healthcare system: A Review," 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 2021, pp. 478-481, doi: 10.1109/ICTAI53825.2021.9673220.
- [2] P. K. Kushwaha, V. Bibhu, B. P. Lohani and D. Singh, "Review on information security, laws and ethical issues with online financial system," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Greater Noida, India, 2016, pp. 49-53, doi: 10.1109/ICICCS.2016.7542350.
- [3] G. Gulati, B. P. Lohani and P. K. Kushwaha, "A Novel Application Of IoT In Empowering Women Safety Using GPS Tracking Module," 2020 Research, Innovation, Knowledge Management and Technology Application for Business Sustainability (INBUSH), Greater Noida, India, 2020, pp. 131-137, doi: 10.1109/INBUSH46973.2020.9392193.
- [4] D. Pareta, I. N. Verma, B. P. Lohani, P. K. Kushwaha and V. Bibhu, "IoT Enabled Smart and Efficient Musical Water Fountain," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 369-373, doi: 10.1109/ICIPTM54933.2022.9754129.
- [5] B. P. Lohani, M. Trivedi, R. J. Singh, V. Bibhu, S. Ranjan and P. K. Kushwaha, "Machine Learning Based Model for

- Prediction of Loan Approval," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2022, pp. 465-470, doi: 10.1109/ICIEM54221.2022.9853160.
- [6] Bibhu, A. Kumar, B. P. Lohani and P. K. Kushwaha, "Robust Secured Framework for Online Business Transactions over Public Network," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 555-560, doi: 10.1109/ICIEM51511.2021.9445380.
- [7] Bibhu, P. K. Kushwaha and B. P. Lohani, "A review of security of the cloud computing over business with implementation," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Greater Noida, India, 2016, pp. 192-198, doi: 10.1109/ICICCS.2016.7542342.
- [8] Amardeep Gupta and Ranjeet Kumar Rout, "ROTEE: Remora Optimization and Tunicate swarm algorithm-based Energy-Efficient cluster-based routing for EH-enabled heterogeneous WSNs," International Journal of Communication System (Q1), vol. 33, no. 6, pp. 1-23, 2022. DOI: <https://doi.org/10.1002/dac.5372>. (IF: 2.0)
- [9] A. D. Gupta and R. K. Rout, "An Effective Optimization Method for Energy Efficient Clustering in EH Wireless Sensor Networks," 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 2021, pp. 699-702, doi: 10.1109/ICTAI53825.2021.9673312.
- [10] S. S. N. Challapalli, P. Kaushik, S. Suman, B. D. Shivahare, V. Bibhu and A. D. Gupta, "Web Development and performance comparison of Web Development Technologies in Node.js and Python," 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 2021, pp. 303-307, doi: 10.1109/ICTAI53825.2021.9673464.
- [11] A. D. Gupta, S. Suman, S. S. N. Challapalli, P. Kaushik and V. Bibhu, "Survey Paper: Comparative Study of Machine Learning Techniques and its Recent Applications," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 449-454, doi: 10.1109/ICIPTM54933.2022.9754206.
- [12] Priyanka, Inderveer Chana, and Ajay Rana. 2012. Empirical evaluation of cloud-based testing techniques: a systematic review. SIGSOFT Softw. Eng. Notes 37, 3 (May 2012), 1–9. <https://doi.org/10.1145/2180921.2180938>
- [13] A. Rana, A. Reddy, A. Shrivastava, D. Verma, M. S. Ansari and D. Singh, "Secure and Smart Healthcare System using IoT and Deep Learning Models," 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2022, pp. 915-922, doi: 10.1109/ICTACS56270.2022.9988676.
- [14] Rana A, Goyal N, Ahlawat A, Jamwal S, Reddy B, Sharma S. Mechanisms involved in attenuated cardio-protective role of ischemic preconditioning in metabolic disorders. *Perfusion*. 2015;30(2):94-105. doi:10.1177/0267659114536760
- [15] Ahlawat, A., Rana, A., Goyal, N. et al. Potential role of nitric oxide synthase isoforms in pathophysiology of neuropathic pain. *Inflammopharmacol* 22, 269–278 (2014). <https://doi.org/10.1007/s10787-014-0213-0>
- [16] A. R. Yeruva, P. Choudhari, A. Shrivastava, D. Verma, S. Shaw and A. Rana, "Covid-19 Disease Detection using Chest X-Ray Images by Means of CNN," 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2022, pp. 625-631, doi: 10.1109/ICTACS56270.2022.9988148.
- [17] Ghosh, S., Rana, A., & Kansal, V. (2020). A benchmarking framework using nonlinear manifold detection techniques for software defect prediction. *International Journal of Computational Science and Engineering*, 21(4), 593-614.
- [18] Raghavendra, M. S., Chawla, P., & Rana, A. (2020, June). A survey of optimization algorithms for fog computing service placement. In 2020 8th international conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO) (pp. 259-262). IEEE.
- [19] Gupta, S., Rana, A., & Kansal, V. (2020). Optimization in wireless sensor network using soft computing. In *Proceedings of the Third International Conference on Computational Intelligence and Informatics: ICCII 2018* (pp. 801-810). Springer Singapore.
- [20] Kunwar, V., Agarwal, N., Rana, A., & Pandey, J. P. (2018). Load balancing in cloud—a systematic review. *Big Data Analytics: Proceedings of CSI 2015*, 583-593.
- [21] Chawla, P., Chana, I., & Rana, A. (2015). A novel strategy for automatic test data generation using soft computing technique. *Frontiers of Computer Science*, 9, 346-363.
- [22] Walia, H., Rana, A., & Kansal, V. (2017, September). A Naïve Bayes Approach for working on Gurmukhi Word Sense Disambiguation. In 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 432-435). IEEE.
- [23] Dash, Y., Dubey, S. K., & Rana, A. (2012). Maintainability prediction of object oriented software system by using artificial neural network approach. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(2), 420-423.
- [24] Dubey, S. K., & Rana, A. (2010). A comprehensive assessment of object-oriented software systems using metrics approach. *International Journal on Computer Science and Engineering*, 2(8), 2726-2730.
- [25] S. Gupta, A. Rana and V. Kansal, "Comparison of Heuristic techniques: A case of TSP," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 172-177, doi: 10.1109/Confluence47617.2020.9058211.
- [26] Ghosh, S., Rana, A., & Kansal, V. (2018). A nonlinear manifold detection based model for software defect prediction. *Procedia computer science*, 132, 581-594.
- [27] Chawla, P., Chana, I., & Rana, A. (2016). Cloud-based automatic test data generation framework. *Journal of Computer and System Sciences*, 82(5), 712-738.
- [28] Bhardwaj, M., & Rana, A. (2016). Key Software Metrics and its Impact on each other for Software Development Projects. *International Journal of Electrical & Computer Engineering* (2088-8708), 6(1).
- [29] Rana, A., & Sharma, S. (2016). Mechanism of sphingosine-1-phosphate induced cardioprotection against I/R injury in diabetic rat heart: Possible involvement of glycogen synthase kinase 3 β and mitochondrial permeability transition pore. *Clinical and Experimental Pharmacology and Physiology*, 43(2), 166-173.
- [30] G. Dubey, A. Rana and N. K. Shukla, "User reviews data analysis using opinion mining on web," 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Greater Noida, India, 2015, pp. 603-612, doi: 10.1109/ABLAZE.2015.7154934.
- [31] Ghosh, S., Rana, A., Kansal, V. (2017). Predicting Defect of Software System. In: Satapathy, S., Bhateja, V., Udgata, S., Pattnaik, P. (eds) *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*. Advances in Intelligent Systems and

- Computing, vol 516. Springer, Singapore.
https://doi.org/10.1007/978-981-10-3156-4_6
- [32] Sanjay Kumar Dubey, Ajay Rana, and Yajnaseni Dash. 2012. Maintainability prediction of object-oriented software system by multilayer perceptron model. SIGSOFT Softw. Eng. Notes 37, 5 (September 2012), 1–4. <https://doi.org/10.1145/2347696.2347703>
- [33] S. Chawla, G. Dubey and A. Rana, "Product opinion mining using sentiment analysis on smartphone reviews," 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2017, pp. 377-383, doi: 10.1109/ICRITO.2017.8342455.
- [34] Dubey, S. K., Rana, A., & Sharma, A. (2012). Usability evaluation of object oriented software system using fuzzy logic approach. International Journal of Computer Applications, 43(19), 1-6.
- [35] Saini, Rimmi, Sanjay Kumar Dubey, and Ajay Rana. "Analytical study of maintainability models for quality evaluation." Indian Journal of Computer Science and Engineering 2.3 (2011): 449-454.
- [36] Ghosh, Soumi, Ajay Rana, and Vineet Kansal. "A statistical comparison for evaluating the effectiveness of linear and nonlinear manifold detection techniques for software defect prediction." International Journal of Advanced Intelligence Paradigms 12.3-4 (2019): 370-391.
- [37] A. Singh, M. Chaudhary, A. Rana and G. Dubey, "Online Mining of data to generate association rule mining in large databases," 2011 International Conference on Recent Trends in Information Systems, Kolkata, India, 2011, pp. 126-131, doi: 10.1109/ReTIS.2011.6146853.
- [38] N. Tyagi, A. Rana and V. Kansal, "Creating Elasticity with Enhanced Weighted Optimization Load Balancing Algorithm in Cloud Computing," 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 600-604, doi: 10.1109/AICAI.2019.8701375.
- [39] Dubey, Sanjay Kumar, and Ajay Rana. "A fuzzy approach for evaluation of maintainability of object oriented software system." International Journal of Computer Applications 49.21 (2012).