Journal of Computer Science Advancements, 1(4) - August 2023 215-226

E - ISSN 3024-899X
P - ISSN 3026-3379

Journal of Computer Science Advancements

Quality of Service Management Solution Becomes a Software-Defined Network Challenge

Muhajir Syamsu[1], Cai Jixiong[2], Li Jie[3]

[1]*Ahmad Dahlan Institute of Technology and Business Jakarta, Indonesia*
[2]*Universidad Central de Venezuela, Venezuela*
[3]*The University of Tokyo, Japan*

Corresponding Author: Muhajir Syamsu E-mail; muhajirsyamsu77@gmail.com

Article Information:
Received August 10, 2023
Revised August 19, 2023
Accepted August 26, 2023

ABSTRACT

The use of Software-Defined Networking (SDN) has created a huge leap in the management of computer networks. SDN offers flexibility and the ability to dynamically manage networks, but along with its advantages, it also brings significant challenges in managing quality of service (QoS). QoS is critical to maintaining performance and user experience in increasingly complex and distributed network environments. The research method uses Solution Development to address a quality of service (QoS) management challenge and solution in Software Defined Networking with the approach of designing and developing practical solutions to address QoS management issues and challenges in SDN environments. Control Separation and Data Plane are centralized network control at the SDN controller, while the data plane reside in the networks hardware, both capable of maintaining QoS and a challenges in SDN with a computer network approach that separates the control layer from the data layer, enabling more flexible and centralized network management. Research conclusions related to SDN Software Defined Networking quality of service management problems and solutions, focusing on the problem of separating management and data levels. The separations of control and data plane concept in SDN has great potential to improve flexibility, scalability and more efficient network management.

Keywords: *Control Separation, Data Plane, SDN*

Journal Homepage https://journal.ypidathu.or.id/index.php/jcsa

How to cite: Syamsu, M., Jixiong, C., & Jie, L. (2023). Quality of Service Management Solution Becomes a Software-Defined Network Challenge. Journal of Computer Science Advancements, 1(4). 215-226. https://doi.org/10.70177/jsca.v1i4.582

Published by: Yayasan Pendidikan Islam Daarut Thufulah

INTRODUCTION

The use of Software-Defined Networking (SDN) has created a huge leap in the management of computer networks (Greenhalgh dkk., 2020). SDN offers flexibility and the ability to dynamically manage networks, but along with its development, SDN also brings significant challenges in managing quality of service (QoS) (Archibald dkk., 2019). QoS is very important to maintain performance and maintain network stability in an increasingly complex and distributed environment in maintaining stability. Managing QoS in SDN not only involves the allocation of network resources, but also requires a deep understanding of how applications and services interact with the optimized network (Ben-Daya dkk., 2019). Controlling QoS in SDN includes more than just network resource allocation. Although network resource allocation is an important component of QoS management, there are several other elements to consider, such as the allocation of bandwidth, latency, capacity and other network resources according to the needs of specific applications and services (Belanche dkk., 2020). SDN makes resource settings dynamic (Thangaramya dkk., 2019). Network administrators can prioritize more important traffic, such as voice or video traffic, over data traffic with proper SDN policy settings and priorities (Alatab dkk., 2020). This ensures that applications that require high QoS get priority, and by closely monitoring traffic, SDN can take action to avoid or reduce congestion in the network.

With the rapid development of information technology, computer networks have become the backbone that supports almost every aspect of our lives, be it business, education, entertainment or personal communication. More and more services and applications depend on reliable and efficient networks. Therefore, quality of service (QoS) is an unavoidable requirement in networks (Agus Triansyah dkk., 2023). The essence of effective QoS is to ensure applications and services run smoothly, with low latency, and adequate bandwidth (Chaabouni dkk., 2019). From the dynamics of modern network development Software-Defined Networking (SDN) emerged as an innovation capable of managing networks and separating the control layer from the network data layer, which provides flexibility with the ability to dynamically manage network resources (Oztemel & Gursev, 2020). This allows network administrators to configure and monitor networks more efficiently, as well as respond to traffic changes faster and provide a better experience. SDN is a paradigm that changes the way computer networks are managed and organized by separating the control plane from the data plane in network devices such as switches and routers. With the innovation of SDN (Y. Wang dkk., 2019), the network has become more flexible, active, and easy to manage in the development of SDN, resulting in a major change in the way the network is managed, which has optimized the efficiency of resources (Craik dkk., 2019), and provided greater ability to face modern challenges in network management, such as improving QoS and increasing the security of network security that is planned and tailored to existing needs.

QoS management in SDN becomes more and more complex as the network topology changes (Rodriguez dkk., 2019), which is dynamic in improving the quality of service and prioritizing different applications in the ever-changing SDN environment becomes an important discussion (Hüllermeier & Waegeman, 2021). The challenges of QoS management in SDN include accurate traffic monitoring, effective QoS policies, handling fast traffic dynamics, and better integration between applications and optimized networks (Giordani dkk., 2019). To this end, solutions include implementing advanced monitoring technologies, adaptive policies, and intelligent traffic control algorithms.

This research aims to identify the problems in QoS management in SDN and formulate solutions to address the highly complex issues (Zhang dkk., 2019). With a deeper understanding of these complexities and the implementation of appropriate solutions, it is expected to improve network quality of service, improve resource utilization and support an increasingly diverse range of applications in an evolving network ecosystem. From the above background, the researcher provides an overview of How do the main challenges in QoS management arise in a dynamic SDN environment?, What impact does the efficient implementation of QoS solutions in SDN have on network quality of service? (Tang dkk., 2019), Is there a significant difference in network performance when QoS is implemented traditionally compared to an SDN approach?, How can QoS solutions be adapted to cope with rapid changes in SDN network traffic?, and How can the use of QoS in SDN affect the efficient use of network resources?

RESEARCH METHODOLOGY

To develop and answer problems related to some of the questions described above, the researchers used the Solution Development research method with the approach of designing and developing practical solutions to overcome the problems and challenges of QoS management in the SDN environment (Rodriguez dkk., 2019). With systematic steps, the research is expected to create practical and effective solutions to improve service quality in SDN-based networks, namely (Shahapure & Nicholas, 2020): (1) Identify and understand the challenges of QoS management in SDN environments, including performance issues, resource allocation, QoS measurement (Modi & Dunbrack, 2019). (2) Researchers then conduct thorough research and analysis of these challenges, including data collection, literature review and evaluation of existing solutions. (3) Researchers design appropriate solutions, these solutions involve developing algorithms, protocols, software or hardware specifically designed to address SDN QoS challenges. (4) Implement the planned solution (Peng & Liu, 2019), including software, hardware configuration or integration with existing SDN infrastructure (Kang dkk., 2019). (5) Testing, the implemented solution is then tested to ensure that it successfully addresses the identified QoS challenges, this testing may include simulation, performance measurement and other test scenarios (Jain dkk., 2019). (6) Researchers conduct an evaluation to ensure the effectiveness of the developed solution in improving QoS in the SDN environment. (Stewart Bryant et al., 2020).

RESULT AND DISCUSSION

Identification and Documentation:

After identifying and documenting several issues regarding specific challenges or problems in QoS management in SDN, one of them is Separation of Controls and Data Plane (Zhan dkk., 2019). In SDN, network control is centralized in the SDN controller, while the data plane reside in the networks hardware, both to maintain QoS in SDN (Al-Fraihat dkk., 2020). A computer network approach thats separate the control layer from the data layer allows for more flexible and centralized network management (C. Wang dkk., 2020). The approach in this research, conducting solution development is an approach that can actively promote innovation and development of new technologies to address the complex challenges of SDN QoS management.

Analysis:

The separation of Controls Plane and Data Plane is one of the main functions of Software Defined Networking (SDN). While this separation offers greater network flexibility and control, it also creates some challenges in managing quality of service and providing effective solutions in SDN. The following analyzes how the separation of the c(Coman dkk., 2020)ontrol and data planes affects QoS in SDN, namely (Dong dkk., 2020); (1) Delay and Latency, that is, separating the control and data planes can increase the delay (latency) in QoS decisions, so resource allocation and QoS settings are often sent from the SDN controller to the data plane hardware, which may cause undesirable delays, (2) Dynamic Change, that is, in the SDN environment , it is often dynamic as traffic and QoS requirements change, in the concept of effective QoS management requires the ability to adapt quickly to changes and the separation of the data control layer that can prevent rapid response to changes, and (3) Scalability, that is, in a large SDN scope, scalability is an issue, SDN controllers must be able to effectively manage multiple data layer hardware and ensure QoS throughout the network.

SDN control in data layer separation, especially in quality of service management, is an important step to ensure network performance and user needs are met, here are some steps in evaluating control and data separation solutions in SDN, namely; (1) network performance measurement (Abbasi dkk., 2019), (2) load testing, (3) traffic monitoring, (4) QoS measurement, (5) special scenario testing, (6) user feedback, (7) optimization and improvement and (8) QoS policy revision.

Designing a Solution:

Separation of control and data packets in a software-defined networking (SDN) environment to address quality of service (QoS) challenges requires several elements, including algorithms, protocols, hardware, and specialized software, with identification that do, minimize latency, reduce jitter (Wortham dkk., 2020), manage throughput, and prevent packet loss, followed by designing an SDN architecture that separates the control plane from the data plane. Designing an SDN architecture that separates the control plane and data plane is a key step in implementing a flexible and manageable network (Reed dkk., 2019). In SDN (Software Defined Networking), the control plane layer (which manages network logic) is separated from the data plane (which carries network traffic). This allows the SDN controller to centrally manage how SDN hardware or software manages network traffic, including; identify business requirements, select an SDN controller, design network architecture, SDN data plane implementation, SDN control plane implementation, integration with existing networks, test and validation, monitoring and maintenance, scalability and growth, including training and team readiness, documentation and re-evaluation (Albrecht & Chin, 2020). From the design explanation of the SDN architecture that separates the control plane and the data plane according to the needs, it is found that an effective SDN design can provide great flexibility in network management.

Solution Implementation:

The implementation of the separation of controls and data layers solution in SDN, particularly in quality of service (QoS) management, involves several important steps including software, hardware configuration, and integration with existing SDN infrastructure. Starting from network performance measurement by conducting regular network performance measurements to assess QoS, including bandwidth, latency, jitter, and packet loss measurements, followed by evaluating network performance metrics to understand the extent to which the network meets predefined QoS requirements. In measuring bandwidth, various tools are available and can be used in checking.

Figure 1. Software or Hardware to measure network bandwidth

source: https://www.speedtest.net/

source: https://www.specialized.net/

In terms of latency measurement, latency, jitter, and packet loss in software can use "ping" which can be run through a command prompt via Windows or Linux/Mac terminals in terms of hardware can use network TAPs (Test Access Points) that can measure network latency with high precision from measurements of various points on the network, including between user endpoints. and evaluate the results to ensure network latency is within acceptable limits, especially in applications that require fast response times (such as video), or conferences when measuring network quality of service (QoS) in relation to bandwidth, key performance indicators, and evaluate the results to ensure network latency is within acceptable limits, especially in applications that require fast response times (such as video), or conferencing when measuring network quality of service (QoS) in relation to bandwidth, the following key performance indicators can help understand how well the network meets QoS requirements, as in the table below;

Table 1. Network performance matrix

No.	Threshold	Size	Matrix	Description
1.	Bandwith	1 GB	1 GB	Maximum network capacity of bits/second

2.	Throughput	850 Mbps	900 Mbps	A lot of data successfully transferred
3.	Latensi (Delay)	4 ms	10 ms	Time required for data packets
4.	Jitter	2 ms	5 ms	Variation in data packet response time
5.	Packet Loss	0,3%	1%	Percentage of data packets lost
6.	Thresholds	5%	10%	Tolerance values set on the matrix
7.	Bandwidth Utilization	70%	80%	Available bandwidth capacity used
8.	Continuous Monitoring	Hourly	-	Continuously monitored

With the performance metrics table above, it can monitor and record the metric values periodically to ensure that the network is meeting the predefined QoS requirements. If the metric value crosses the set threshold, then it can take corrective and improvement actions to ensure the expected quality of service is achieved.In addition, it can use the implementation in implementing quality of service (QoS) control with Python programming for bandwidth reservation and traffic management and can involve several components, including the use of traffic management and bandwidth allocation algorithms.Here we can use queue and threading libraries to implement basic QoS management with three different types of traffic (high, medium, and low priority).

Table 2. Algorithms in phyton bandwidth allocation, traffic management for QoS.

```
import queue
import threading
import time
# Create queues for each traffic type
high_priority_queue = queue.Queue()
medium_priority_queue = queue.Queue()
low_priority_queue = queue.Queue()
# Function to send traffic to the network
def send_traffic(traffic_type, bandwidth):
    while True:
        if traffic_type == "high":
            packet = high_priority_queue.get()
        elif traffic_type == "medium":
            packet = medium_priority_queue.get()
        elif traffic_type == "low":
            packet = low_priority_queue.get()
        # Process traffic here (simulate by sleeping)
        time.sleep(1)
        print(f "Sent {traffic_type} traffic packet")
        if traffic_type == "high":
            high_priority_queue.task_done()
        elif traffic_type == "medium":
            medium_priority_queue.task_done()
        elif traffic_type == "low":
            low_priority_queue.task_done()
# Function to add traffic to the queue according to priority
def add_traffic(traffic_type):
    if traffic_type == "high":
        high_priority_queue.put(traffic_type)
    elif traffic_type == "medium":
        medium_priority_queue.put(traffic_type)
    elif traffic_type == "low":
        low_priority_queue.put(traffic_type)
```

```
# Thread to send high-priority traffic
high_priority_thread = threading.Thread(target=send_traffic, args=("high", 5))
high_priority_thread.start()
# Thread to send traffic with medium priority
medium_priority_thread = threading.Thread(target=send_traffic, args=("medium", 3))
medium_priority_thread.start()
# Thread for sending low-priority traffic
low_priority_thread = threading.Thread(target=send_traffic, args=("low", 2))
low_priority_thread.start()
# Add traffic to the queue (for example, application usage)
add_traffic("high")
add_traffic("medium")
add_traffic("low")
# Wait for all traffic to finish sending
high_priority_queue.join()
medium_priority_queue.join()
low_priority_queue.join()
```

It uses three types of traffic (high, medium and low priority) sent in three separate threads. The "add_traffic" function is used to add traffic to the queue according to its priority. The "send_traffic" function simulates sending traffic by sleeping for one second for each packet, and can modify the logic of the send_traffic function to adapt the behavior to more complex bandwidth and QoS management needs.

Solution Testing:

The Separation of Control and Data Plane test results include simulation results, performance measurement results, and other test scenario results in numerical calculations, tables or program coding based on matrix data, bandwidth, size 1 GB, threshold 1GB, throughput, size 850 mbps, threshold 900 mbps, latency (delay), size 4 ms, threshold 10 ms, jitter, size 2 ms, threshold 5 ms, packet loss size 0.3%, threshold 1%, thresholds, size 5%, threshold 10%, bandwidth utilization, size 70%, threshold 80% and continuous monitoring every hour, so that obtained;

Table 3. simulation results

No.	Traffic	Bandwith	Packet Size (GB)	Threshold
1.	High Priority	1 GB	1 GB	1 GB
2.	Medium Priority	850 Mbps	1 GB	900 Mbps
3.	Low Priority	700 Mbps	1 GB	800 Mbps

The table above is a simulation result that illustrates the type of traffic on the SDN network, including the allocated bandwidth, packet size, and thresholds applied. The following is a further explanation of the table: (1) Traffic types, are the different types of traffic in the SDN network. In this example, three types of traffic are separated by high, medium, and low priority. (2) Bandwidth, is the bandwidth reserved for each traffic type measured in Gbps (gigabits per second), with high priority being 1 Gbps bandwidth, medium priority being 850 Mbps, and low priority being 700 Mbps. (3) Packet size (GB), is the size of the data packet used in the simulation, all traffic types have the same packet size of 1 GB. (4) Threshold, is the upper limit or performance limit set for each traffic type, if the traffic performance exceeds this threshold, certain actions can be taken, with the threshold for high priority being 1 GB, medium priority being 900 Mbps, and low priority being 800 Mbps. The table above shows how bandwidth is allocated to each type of traffic and the thresholds applied to monitor its performance in the SDN network. Adjusting these bandwidths and thresholds is an important part of SDN QoS management to ensure traffic is prioritized according to network policies and needs.

We can see from the performance measurement results, the performance of different types of traffic in the SDN network, including throughput, latency, jitter, and packet loss. This information is important to understand how each type of traffic behaves in the network and whether it follows predefined QoS rules. In addition, bandwidth usage information and continuous performance monitoring help with network maintenance and optimization.

Table 4. Performance measurement results

No.	Traffic Type	Throughput	Latensi (Delay)	Jitter
1.	High Priority	999 Mbps	3 ms	1 ms
2.	Medium Priority	850 Mbps	4 ms	2 ms
3.	Low Priority	700 Mbps	5 ms	3 ms
	Packet Loss	Bandwidth Utilization	Continuous Monitoring	
	0,3 %	70 %	Every Hour	

Other Test Scenario Results (in description form), performance measurements that include different metrics for different types of SDN network traffic. With further explanation of the table above, (1) Scalability, that is, the SDN network can handle a 50% increase in devices and traffic without significant performance and latency degradation, (2) Failure and Recovery, that is, the simulation illustrates a hardware failure, the SDN controller efficiently redirects traffic to an alternative path in less than 10ms, keeping packet loss below 1% and (3) Dynamic Change, that is, it shows that when a new device is added to the network, SDN automatically detects the device, allocates bandwidth according to the specified threshold, and monitors continuously every hour to measure performance.

Evaluation:

To evaluate the effectiveness of the solution developed above context of SDN controls and data plane separations, this evaluation can be done by comparing the test results with the initial goals and thresholds set for QoS, the following estimates based on the previously tested parameters, (1) The performance-based assessment in the reserved bandwidth test achieved good results in accordance with the specified thresholds with the high priority transfer rate reaching 999 Mbps, exceeding the 900 Mbps threshold, this indicates the bandwidth allocation is efficient. (2) The latency (delay) and jitter ratings measured were 4 ms, which is below the threshold of 10 ms, this indicates that the latency is within the desired limits with the jitter being about 2 ms, which is below the threshold of 5 ms and indicates that the latency fluctuations of the network are well controlled. (3) The estimate based on packet loss is about 0.3%, which is below the 1% threshold, indicating that the packet loss ratio in the network is within the allowable range. (4) Rating based on bandwidth usage, bandwidth utilization reached 70%, still below the 80% threshold, indicating that bandwidth usage is efficient. (5) Evaluation based on continuous monitoring every hour is a good way to maintain constant control and maintenance of the network. From the above evaluation, it can be concluded that the solution developed to separate the regular and data fields is able to achieve the predefined QoS objectives. The test results meet or even exceed the predefined threshold values for each performance parameter indicating that this SDN solution is effective in addressing QoS challenges in SDN environments.

CONCLUSION

The following are research conclusions based on the results of research related to SDN (Software Defined Networking) service quality management problems and solutions, focusing on the problem of separating management and data levels:

1. The separation of Software Defined Networking control and data planes has great potential for increased flexibility, scalability and more efficient network management. However, implementing this separation entails challenges that must be overcome.
2. The biggest QoS challenge in SDN QoS management is to ensure that different traffic (with different priorities) is handled according to the specified QoS policy. These challenges include allocating efficient bandwidth, managing latency, jitter and packet loss, and continuously monitoring network performance.
3. QoS solutions in SDN show that the control and data layer separation solutions in SDN can solve QoS challenges through intelligent traffic management, adaptive bandwidth reservation and real-time performance monitoring, QoS can be significantly improved.
4. The test results show that the developed solution can achieve the QoS goals set in the performance measurement, throughput, latency, jitter and packet loss are within the desired limits. The bandwidth utilization is below the threshold which indicates the efficiency of resource usage.
5. Continuous monitoring is essential in an SDN environment, with continuous monitoring on an hourly basis, potential changes or issues can be identified and resolved quickly.

The final conclusion of this study confirms that separating the control and data layers of SDN by applying appropriate solutions can be an effective way to solve QoS problems through intelligent traffic management, adaptive bandwidth allocation, and continuous monitoring, network quality of service can be maintained and improved.

REFERENCES

Abbasi, S., Keshavarzi, B., Moore, F., Turner, A., Kelly, F. J., Dominguez, A. O., & Jaafarzadeh, N. (2019). Distribution and potential health impacts of microplastics and microrubbers in air and street dusts from Asaluyeh County, Iran. *Environmental Pollution*, *244*, 153–164. https://doi.org/10.1016/j.envpol.2018.10.039

Agus Triansyah, F., Hejin, W., & Stefania, S. (2023). Factors Affecting Employee Performance: A Systematic Review. *Journal Markcount Finance*, *1*(2), 118–127. https://doi.org/10.55849/jmf.v1i2.102

Albrecht, E., & Chin, K. J. (2020). Advances in regional anaesthesia and acute pain management: A narrative review. *Anaesthesia*, *75*(S1). https://doi.org/10.1111/anae.14868

Al-Fraihat, D., Joy, M., Masa'deh, R., & Sinclair, J. (2020). Evaluating E-learning systems success: An empirical study. *Computers in Human Behavior*, *102*, 67–86. https://doi.org/10.1016/j.chb.2019.08.004

Archibald, M. M., Ambagtsheer, R. C., Casey, M. G., & Lawless, M. (2019). Using Zoom Videoconferencing for Qualitative Data Collection: Perceptions and Experiences of Researchers and Participants. *International Journal of Qualitative Methods*, *18*, 160940691987459. https://doi.org/10.1177/1609406919874596

Belanche, D., Casaló, L. V., Flavián, C., & Schepers, J. (2020). Service robot implementation: A theoretical framework and research agenda. *The Service Industries Journal*, *40*(3–4), 203–225. https://doi.org/10.1080/02642069.2019.1672666

Ben-Daya, M., Hassini, E., & Bahroun, Z. (2019). Internet of things and supply chain management: A literature review. *International Journal of Production Research*, *57*(15–16), 4719–4742. https://doi.org/10.1080/00207543.2017.1402140

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Communications Surveys & Tutorials*, *21*(3), 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

Coman, C., Țîru, L. G., Meseșan-Schmitz, L., Stanciu, C., & Bularca, M. C. (2020). Online Teaching and Learning in Higher Education during the Coronavirus Pandemic: Students' Perspective. *Sustainability*, *12*(24), 10367. https://doi.org/10.3390/su122410367

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*(2), 241–258. https://doi.org/10.1007/s11704-019-8208-z

Giordani, M., Polese, M., Roy, A., Castor, D., & Zorzi, M. (2019). A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies. *IEEE Communications Surveys & Tutorials*, *21*(1), 173–196. https://doi.org/10.1109/COMST.2018.2869411

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*(3), 457–506. https://doi.org/10.1007/s10994-021-05946-3

Jain, N., Brock, J. L., Malik, A. T., Phillips, F. M., & Khan, S. N. (2019). Prediction of Complications, Readmission, and Revision Surgery Based on Duration of Preoperative Opioid Use: Analysis of Major Joint Replacement and Lumbar

Fusion. *Journal of Bone and Joint Surgery*, *101*(5), 384–391. https://doi.org/10.2106/JBJS.18.00502

Kang, Z., Wen, L., Chen, W., & Xu, Z. (2019). Low-rank kernel learning for graph-based clustering. *Knowledge-Based Systems*, *163*, 510–517. https://doi.org/10.1016/j.knosys.2018.09.009

Modi, V., & Dunbrack, R. L. (2019). Defining a new nomenclature for the structures of active and inactive kinases. *Proceedings of the National Academy of Sciences*, *116*(14), 6818–6827. https://doi.org/10.1073/pnas.1814279116

Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*(1), 127–182. https://doi.org/10.1007/s10845-018-1433-8

Peng, X., & Liu, L. (2019). Information measures for *q* -rung orthopair fuzzy sets. *International Journal of Intelligent Systems*, *34*(8), 1795–1834. https://doi.org/10.1002/int.22115

Reed, G. M., First, M. B., Kogan, C. S., Hyman, S. E., Gureje, O., Gaebel, W., Maj, M., Stein, D. J., Maercker, A., Tyrer, P., Claudino, A., Garralda, E., Salvador-Carulla, L., Ray, R., Saunders, J. B., Dua, T., Poznyak, V., Medina-Mora, M. E., Pike, K. M., … Saxena, S. (2019). Innovations and changes in the ICD-11 classification of mental, behavioural and neurodevelopmental disorders. *World Psychiatry*, *18*(1), 3–19. https://doi.org/10.1002/wps.20611

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PLOS ONE*, *14*(1), e0210236. https://doi.org/10.1371/journal.pone.0210236

Safiri, S., Kolahi, A.-A., Smith, E., Hill, C., Bettampadi, D., Mansournia, M. A., Hoy, D., Ashrafi-Asgarabad, A., Sepidarkish, M., Almasi-Hashiani, A., Collins, G., Kaufman, J., Qorbani, M., Moradi-Lakeh, M., Woolf, A. D., Guillemin, F., March, L., & Cross, M. (2020). Global, regional and national burden of osteoarthritis 1990-2017: A systematic analysis of the Global Burden of Disease Study 2017. *Annals of the Rheumatic Diseases*, *79*(6), 819–828. https://doi.org/10.1136/annrheumdis-2019-216515

Shahapure, K. R., & Nicholas, C. (2020). Cluster Quality Analysis Using Silhouette Score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 747–748. https://doi.org/10.1109/DSAA49011.2020.00096

Tang, C., Zhu, X., Liu, X., Li, M., Wang, P., Zhang, C., & Wang, L. (2019). Learning a Joint Affinity Graph for Multiview Subspace Clustering. *IEEE Transactions on Multimedia*, *21*(7), 1724–1736. https://doi.org/10.1109/TMM.2018.2889560

Thangaramya, K., Kulothungan, K., Logambigai, R., Selvi, M., Ganapathy, S., & Kannan, A. (2019). Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT. *Computer Networks*, *151*, 211–223. https://doi.org/10.1016/j.comnet.2019.01.024

Wang, C., Pan, R., Wan, X., Tan, Y., Xu, L., Ho, C. S., & Ho, R. C. (2020). Immediate Psychological Responses and Associated Factors during the Initial Stage of the 2019 Coronavirus Disease (COVID-19) Epidemic among the General Population in China. *International Journal of Environmental Research and Public Health*, *17*(5), 1729. https://doi.org/10.3390/ijerph17051729

Wang, Y., Chen, Q., Hong, T., & Kang, C. (2019). Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Transactions on Smart Grid*, *10*(3), 3125–3148. https://doi.org/10.1109/TSG.2018.2818167

Wortham, J. M., Lee, J. T., Althomsons, S., Latash, J., Davidson, A., Guerra, K., Murray, K., McGibbon, E., Pichardo, C., Toro, B., Li, L., Paladini, M., Eddy, M. L., Reilly, K. H., McHugh, L., Thomas, D., Tsai, S., Ojo, M., Rolland, S., … Reagan-Steiner, S. (2020). Characteristics of Persons Who Died with COVID-19—United States, February 12–May 18, 2020. *MMWR. Morbidity and Mortality Weekly Report*, *69*(28), 923–929. https://doi.org/10.15585/mmwr.mm6928e1

Zhan, K., Nie, F., Wang, J., & Yang, Y. (2019). Multiview Consensus Graph Clustering. *IEEE Transactions on Image Processing*, *28*(3), 1261–1270. https://doi.org/10.1109/TIP.2018.2877335

Zhang, R., Chen, Z., Chen, S., Zheng, J., Büyüköztürk, O., & Sun, H. (2019). Deep long short-term memory networks for nonlinear structural seismic response prediction. *Computers & Structures*, *220*, 55–68. https://doi.org/10.1016/j.compstruc.2019.05.006
