International Journal of
Electrical and Computer System Design
ISSN: 2582-8134 www.ijecsd.com

A Survey- ML based Offloading in HetNets

Duraimurugan J*[1], Dr.Indra Gandhi K[2],
*[1]Department of IST, CEG Campus, Anna University, Chennai, India, dmurugan02@gmail.com
[2]Department of IST, CEG Campus, Anna University, Chennai, India, jeevaindra@gmail.com

Abstract: The deployment of ultra-dense heterogeneous networks (HetNets) composed up of macro, micro, pico, and femto cells is required due to the exponential expansion of mobile users and the critical necessity for excellent service quality. Each cell type of HetNets delivers a various level of cell range and a unique system capacity. As a result, there is an urgent requirement to balance the loads between small cells, especially given the users' erratic distribution over different mobility axes. The intelligent load balancing models, including those based on machine learning (ML) technology, that have been created in HetNets are surveyed in this work. The evaluation offers a framework and an approach to design load balancing models for future HetNets that are affordable, adaptable, and intelligent. Additionally, a general explanation of the load balancing issue is provided.

The idea of load balancing is initially introduced, followed by explanations of its goal, functionality, and assessment standards. In addition, a basic Offloading frame and its operation are given. The next step is to undertake a thorough literature review that includes approaches and fixes for the load balancing issue. To demonstrate the historical evolution of load balancing models, a thorough literature study of ML-driven load balancing methods is completed. Finally, the existing difficulties in putting these models into practice are discussed, along with the potential operational implications of load balancing in the future.

Keywords: Offloading, HetNets, HO, Optimization, ML, 5G Network

1. Introduction

The demand for mobile broadband services with faster data rates and improved quality of service (QoS) has risen dramatically as a result of the massive growth in the use of smart devices and applications, which comprise information and communication technology [1]. The sixth generation (6G) of wireless networks will need to develop enhanced broadband, huge access, ultra-reliable, and low-latency service capabilities that are stronger and smarter than those provided by the fifth generation (5G) wireless network in order to fulfil this enormous demand. The structure of the 6G network is expected to be very diverse, widely deployed, and dynamic [2]. The advantages of ultra-wideband, ultra-wide access, ultra-reliability, and low latency that 6G technology will provide will also raise important challenges, such as how such networks will be managed.

Proceeding of "Technology Integration for Sustainable Development: An International E-Conference on Electrical, Electronics, Computer Science and Mechanical Engineering. (EECM-2023)". Organized by SJUIT.

Mobile networks have already produced a significant amount of data, which can be used to make better educated management decisions for HetNets. Artificial intelligence (AI)/ML presents a wonderful possibility in this situation because it can offer insightful data analysis based on the already accessible data. The capacity of AI/ML techniques to automatically learn system experience, forecast future scenarios, and adapt to operational contexts is their promising feature [8]. Users can use AI/ML to select cells that will best serve them, manage dynamic interconnection with numerous cells, and select the most suitable HO target cells to ensure service continuity [7]. BSs can employ AI/ML to enhance system parameters like mobility parameters to provide load balancing and fortify networks. Valuable insights will be possible by using AI/ML algorithms to train the observed data. Numerous functions can be learned to enable forecasting, decision-making, and optimisation for balancing loads in 6G HetNets.

II. LOAD BALANCING OVERVIEW

Load imbalance is a problem that Future HetNets are vulnerable to. The main causes are unpredictable user access to the network, frequent modifications, and diverse business needs. The load balancing issue is briefly explained in this section.

Offloading is the equitable allocation of cell loads among neighbouring cells or the transfer of traffic from congested cells to more accessible cells in order to maintain highly efficient radio resource utilisation. Each connected user shares a cell's available bandwidth. The cell reaches a loaded condition, when the workload is above or close to the limit, when the maximum number of users per cell is reached. It also occurs when admission control begins to bar new users from joining the network in an effort to slow the increase in throughput. The scheduler should ideally be able to allocate the necessary physical resource blocks (PRBs) to users for each service that has a certain QoS.

The load balancing mechanism should eventually start to actively route users to other cells to avoid overloading or congestion as more and more users join the network and use PRB resources. In order to prevent cell overload and subsequent performance loss, a load balancing function or algorithm is used.

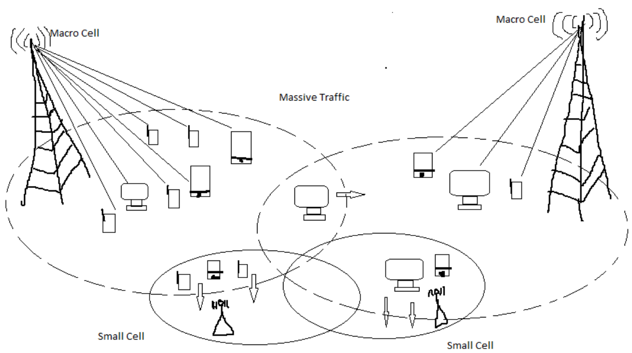
Comparing the research on load balancing that has already been done.

Survey	Year	Description
[9]	2014	• To dispel these erroneous cellular networks are once again examined in the context of a load-balanced HetNet. • Load balancing techniques for HetNets are described and contrasted. • Plans for further Offloading research have been made.
[10]	2014	• Provide suitable strategies for self-optimization in cellular networks, demonstrating various load balancing structures.
[11]	2016	• To enhance load balancing, a review of the adaptive cell selection technique is employed.
[12]	2016	• The complexity of algorithms is also explored, and a thorough analysis of load balancing methods in the cloud environment is completed.
[13]	2017	• The load balancing techniques are categorized, and their benefits and drawbacks are assessed. • There is a list of the new study areas for

		load balancing algorithms in cloud computing.
[14]	2018	• The deterministic and non-deterministic load balancing algorithms are the two main forms of load balancing utilised in SDN, and the advantages and disadvantages of each algorithm are contrasted. • By analysing the difficulties of particular algorithms, important research techniques are found for upcoming study.
[15]	2019	• IoT load balancing techniques fall into two categories. • Evaluation parameters are highlighted for the load balancing issues in IoT systems.
[16]	2019	• Algorithms for load balancing in fog computing are looked at. • A modelling tool and analysis based on various measurements of fog are presented.
[17]	2019	• There are different types of load balancing employed in cloud computing. • Each class's benefits, drawbacks, ideas, and difficulties are examined.
[18]	2018	• Details on the tasks carried out in data centres and the cloud environment, as well as the load balancing strategies used, are presented.
[19]	2020	• The load balancing technique in SDN is reviewed in a systematic manner in the literature. • Traditional or AI-based load balancing techniques are now used in SDN. • The KPIs applied to the existing approach. • Future research recommendations are discussed.
[20]	2020	• The influence of SDN and Open-Flow approaches on load balancing is described. • A description of SDN load balancing strategies is given. • The KPIs applied to the existing approach.
[21]	2020	• KPIs are utilised to offer an overall classification of load balancing in SDN. • Current approaches are discussed..
[22]	2021	• Looks at the history of HetNets' load balancing. • Examine HetNet load balancing strategies based on SDN. • Lists all of the current issues.
[23]	2021	• It offers a thorough analysis of the concept and evolution of SDN and the load balancing approach.
[24]	2021	• Provides a thorough analysis of load

		balancing techniques, taking into account the significance of fault tolerance in load balancing algorithms. • Divides the various load balancing methods into two groups: distributed and centralised. • Looks at qualitative factors including dependability, accessibility, effectiveness, etc.
[25]	2021	• Check out the load-balancing method used in the cloud environment right now. • A proposed fault-tolerant model.
[26]	2022	• Offers a thorough analysis of load balancing.
The present research		• Covers an in-depth examination of future HetNet Offloading methods based on machine learning. • Performs a thorough study of load balancing, taking into account the idea, algorithms, addressing, KPI, issues, and solutions. • Describe the SON functions' operational flow. • Outlines the historical progression and creation of various models. • Issues with current approaches are noted, and fresh lines of inquiry that can address these issues and support the development of a more stable network structure are investigated and debated.

Offloading between Cells



In the deployment of HetNets or multi-layer networks, Offloading is an essential feature. HetNets were used primarily to boost capacity and coverage in areas with unequal user distribution. In places with high user demand, small cells are typically put to provide additional capacity, while using macro cells to provide coverage in the remaining regions.

Macro cells have a higher transmission power than tiny cells. Users favour the cell with the highest received signal strength (RSS) while connecting to a cellular network. Because there will be considerably fewer user clusters connected to small cells than to macro cells, the available resources of small cells won't be fully utilised, and competition for resources in the macro cell will remain strong.

OBJECTIVE

The following objectives of load balancing are intended to be accomplished:

• Maximise cell re-selection/HO parameters to reduce the number of HOs and redirections needed to maintain load balance between neighbouring cells;

• Increase system capacity by controlling cell congestion;

• Establish efficient and effective management for optimum performance.

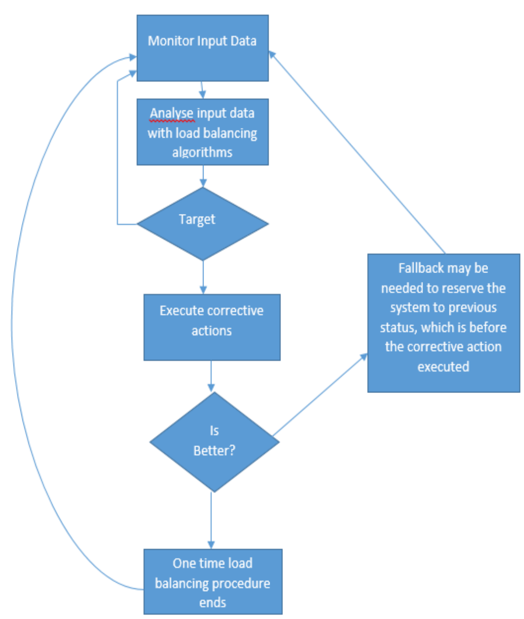
FUNCTIONALITY

The Offloading algorithm gives the UEs encamped in or linked to a cell. HO of UEs between cells might be delayed or advanced to achieve this function. The following procedures are part of load balancing:

Based on this information, an algorithm then assesses if it is required to disperse the burden across nearby cells. The load report function calculates the load for each cell covered by its own BS, and this calculation is based on the exchange of cell-specific load data via the interfaces X2 or S1 between nearby BSs.

If a change is required, the source cell notifies the neighbouring cell via a mobility change request. Whether the HO parameter settings need to be adjusted is predicted by an algorithm. When necessary, communication takes place between the participating cells to recommend adjustments to the neighbouring cell's HO trigger settings.

CRITERIA FOR EVALUATION AND OUTPUT



Flowchart of Load balancing

The following logical interpretation corresponds to the load balancing algorithm's operational process:

1. The load balancing function keeps track of cell load levels to find imbalances in input data load.

2. Cell load levels are determined, and a balanced distribution of load levels is examined.

3. The optimization algorithm is activated if a load imbalance is discovered.

4. The target cell to unload the traffic is identified through load data of the cells and the Offloading method. The destination cell is chosen based on factors such cell load levels, RSRP, and cell kinds. The required users must be Handover to the desired destination cells, which necessitates data transfer and metadata updates. Corrective activities include any HO events that are carried out for load balancing.

5. The Offloading method assesses the outcome of the manipulations that were carried out.

6. Monitoring of the input data is resumed by the load balancing functionality.

III. ML IN OFFLOADING

An extensive investigation of Machine Learning based offloading models is provided in this section. A brief overview of the ML methods utilized in the survey is provided. The literature's Machine Learning based offloading methods are displayed. Each item (which includes publication dates) is a single article. The reader can view a chronological list of articles that have been released in this field since 2013 in this section. As a result, it is simpler to track how ML-based load balancing models evolve over time. Here, a thorough analysis of each model's structure, its design process, as well as its strengths and weaknesses, has been conducted. The purpose is to emphasise the factors to take into account when developing new design.

A. AN OVERVIEW OF ML ALGORITHMS

A model that helps uncover patterns can be learned, automated, and optimised by computers using a variety of techniques known as machine learning (ML) [94].

An ML technique comprises two stages: the training stage, in this the frame model is learnt from training data, and second stage is decision, in which the trained model generates an predetermined output for each incoming input. ML techniques can be divided into three fundamental categories based on how learning is created: reinforcement learning (RL) supervised, unsupervised. Here several ML algorithms are briefly defined in terms of these three fundamental categories.

SUPERVISED LEARNING

In supervised learning, the agent is given examples of labelled state-action pairings together with a label indicating whether the action is "right" or "wrong". The cornerstone of supervised learning is creating a general rule from training data. In this way, the system's behaviours are predicted or generalised to behave appropriately in situations that are not included in the training set. The most well-known supervised learning algorithms are as follows:

a : REGRESSION

Regression is a form of statistical analysis for simulating the connection between unrelated (input) and related (output) variables. The modelling paradigm's standard error estimates assist us understand how the value of the related variable corresponding to an unrelated variable varies while other unrelated variables are maintained constant. Continuous variables are used in the regression model. This method involves first identifying the independent values. After that, the independent variable's coefficients are calculated in order to minimize the differences between the estimated and actual values. The equation is then created by including potential random mistakes.

b: SUPPORT VECTOR MACHINES (SVM)

A line that separates a group of inputs that should be classified as points in a space of high dimensions is sought after by the SVM. The margin of the hyperplane is the separation between the closest expression vector and the hyperplane separating the two classes. In order to increase its capability to forecast the classes of instances in the feature space that are not yet classified, SVM seeks to maximise the margin separating the hyperplane. Kernel function approaches are used to project data into a higher dimensional space when there isn't a good linear separator available.

c: DECISION TREES (DT)

DT classifies data items by submitting a series of queries connected to each internal node's attribute. To accommodate each potential response, each internal node is divided into a sub-node, creating a hierarchy that is encoded as a tree. An unlabeled instance is classified according to the legitimate responses by moving down the tree from the top node to the root childless node. The lower branches of DT contain purer information than the top branches. One thing to keep in mind when using DT is to keep the hight of the learned trees under the level of the training instances.

UNSUPERVISED LEARNING

Unsupervised learning algorithms seek to learn from unlabeled sequences of inputs without the assistance of a controller who can provide the correct responses or the level of error for each inspection. The most popular supervised learning implementation techniques are listed below:

a: K-MEANS

A set of unlabeled data is grouped using the K-Means technique into K separate groups based on characteristics and features. This method simply requires the original dataset and the necessary K clusters parameters. The algorithm is simple, and the following actions are taken: After selecting a random K number of centroids to cover every point in the dataset, (a) all left points are allocated to the nearest centroids using a distance function; (b) because the original centroid location was not exact, a new centroid is selected and all data is assigned to it; and (c) this process will continue until the convergence condition is met.

b: SELF ORGANIZING MAPS (SOM)

Basically, SOM comes under unsupervised model. Applications for dimension reduction and data clustering employ the SOM technique. A SOM consists of neurons, and values are regularly connected to the grid's nodes in such a way that models with higher similarity are connected to nodes closer together while models with lower similarity are attached to nodes farther apart.

RELATED STUDIES

Even though supervised learning is a useful method, it will challenging to gather training instance for offloading issues from the real world. The creation of training data using simulation programs is one of the ways suggested in the literature for compiling information that a functional network should offer in the use of supervised learning. However, the reliability of the predictions in such a system would depend on the precision of the observations and the accuracy of the simulations. Another choice is to use prior real-world data collections as a training group. The main disadvantage of this strategy is that the planning method, which was created using historical data on traffic flow, must be put into action from the very beginning of the subsequent time period.

Both two types of learning rely on static data, therefore the dataset that was given to the system is used to gauge performance. The RL approach views data as a moving target, which means that although the procedure for learning is controlled by the present policy, it may change based on how states and rewards are allocated. This ML method is popular because it may be used to states when there isn't a verifiable event model.

IV. RESEARCH CHALLENGES

Even while ML algorithms offer fantastic prospects for proving the offloading problem, there are still several evident obstacles to be solved. The complications that arise when ML is used to perform offloading problems are described in this section.

BIG DATA

The pain of dealing with large volumes can be lessened by using parallel programming techniques. By distributing the learning process across several computers or processors, distributed ML techniques can address problems with memory constraints and large-scale ML algorithm complexity. When both the dataset and the frame work are too vast to fix into a one memory, hybrid techniques that guarantee model and data parallelism by concurrently dividing both the data and the model variables allow ML applications to execute effectively in distributed clusters.

VARIETY OF BIG DATA

Next issue is data diversity.The variety, semantic interpretation, and sources of what the dataset represents are all described, along with its structural diversity and the different sorts of data it contains [129]. Learning from diverse data provided by numerous network sources is one factor that increases complexity. It is effective to add features that were discovered at different stages into the model by looking at the data representations from each data source [130]. Big data is frequently referred to as noisy since it combines data from numerous unidentified sources.

Veracity of big data

Given the wide variety of data sources and the inability to fully verify data quality, the correctness and dependability of the source data represent another significant problem for the big data idea. Learning from such speculative, sometimes erroneous data might lead to data misunderstanding and reduce the precision of predictions. The noise brought on by the provision of data from diverse sources could potentially provide data in an unsuitable manner, which would further damage the performance of ML.

STORAGE CAPACITY PROBLEM

ML methods require additional memory in order to store the data and use it to train the models. Content caching has been developed as a solution for the storage capacity problem in HetNets, especially to get over the restriction of limited memory in vehicle networks where behaviour patterns influenced by earlier experiences are frequently utilized. The outcome has been a decrease in latency and quick adaptation to radio link conditions, improving the experience of end users.

1) VELOCITY OF BIG DATA

In order to handle the constant influx of changing data, an ML frame needs to interact with its environment quickly. In a typical ML frame, new data supplied into the system accomplishes the job that has been learned after the system has already been trained using the current training set. Sometimes, ML is unable to learn on its own when new data is introduced. The model could become stale and cease to faithfully reflect how the system is currently operating. A potential solution to the issue of real-time or almost real-time data processing is provided by online schooling.

V. FUTURE DIRECTIONS

NETWORK FUNCTION VIRTUALIZATION (NFV)

SDN technology gives network systems a global perspective and centralised control, increasing their flexibility, dynamic functionality, and programmability. The network separates the control and data planes [137]. The SDN controller monitor the load levels of network nodes because of its central control function. In the event that any flow line becomes congested, the traffic can be split into numerous flow paths using sophisticated offloading techniques [20]. Network resources are efficiently allocated by the SDN technology to enhance network performance and user QoS.

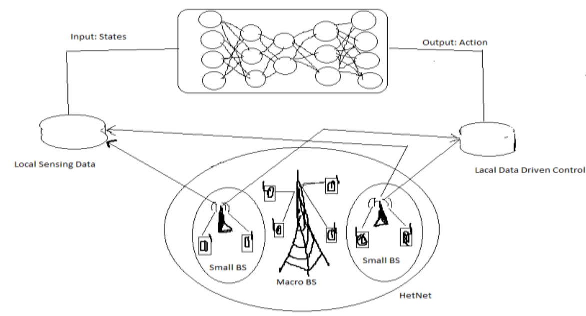
DEEP LEARNING

By "depth", or complexity, to traditional ML models, DL techniques enhance the models. A hierarchy of layers of abstraction is used to represent data. By adding more neurons and hidden layers, NNs can become deeper. They are essential for handling higher dimensional data and mastering ever-more sophisticated models. To take action or generate predictions against specified targets, data is hierarchically retrieved from original data by numerous layers of nonlinear operating units.

The primary benefit of DL is that internal correlations and high-level features can be automatically extracted from complexly structured data without the requirement for a specially created learning procedure [139].

DL manages massive volumes of data and offers additional advantages because big data training prevents overfitting of the model. Multiple DL objectives can be met with a single model. As a result, there is no longer a need to continuously train the frame for various jobs. DL has certain distinctive advantages, but it also has some limiting drawbacks.

Along with recent technology developments including more potent processors, quicker networking, and best software architecture, new methods for training massive data and deeper networks have offered a significant potential for DL research. DL techniques have shown to be efficient at incorporating data into frames created from various sources. This suggests that HetNets can use DL approaches very effectively.



Centralized load balancing methodology.

VI. CONCLUSION

Readers will receive a thorough analysis of ML-oriented offloading techniques in this work. A thorough verification of offloading is done, covering the idea, algorithms, addressing, KPI, issues, and how to regulate them. It gives a thorough account of how HetNets' ML-based load balancing models have evolved over time. First, a fundamental explanation of load balancing and its goals is given in this work. The next section introduces a fundamental load balancing concept and describes each subsequent step.

The definitions of the KPIs used to judge how well offloading models work are supplied. A mathematical model of the user interaction optimization problem is also presented, where the goal is to optimize the utility function while still respecting resource constraints. Additionally, some pointers are provided to help users and BSs associate in the best way feasible. The study also provides a detailed explanation of the CCO and MRO operating principles and their connections with the self-organized networks' activities of joint optimization and load balancing. The control schemes for the load balancing algorithms are also included.

In particular, the paper offers insights on the use of ML techniques to address the load balancing issue in HetNets. ML algorithms and its various categories are presented first. Then, we present a thorough analysis of technical challenges surrounding the use of offloading models based on

Machine Learning methods, evaluations of performances of the models that have been published in the literature

REFERENCES

[1] K. Addali, ‘‘Mobility and resource management for 5G small cell networks,’’ Ph.D. dissertation, École de technologie supèrieure, 2020.

[2] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, ‘‘Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service,’’ IEEE Veh. Technol. Mag., vol. 15, no. 4, pp. 122–134, Dec. 2020.

[3] E. Gures, I. Shayea, A. Alhammadi, M. Ergen, and H. Mohamad, ‘‘A comprehensive survey on mobility management in 5G heterogeneous networks: Architectures, challenges and solutions,’’ IEEE Access, vol. 8, pp. 195883–195913, 2020.

[4] I. Shayea, M. Ergen, M. H. Azmi, S. A. Colak, R. Nordin, and Y. I. Daradkeh, ‘‘Key challenges, drivers and solutions for mobility management in 5G networks: A survey,’’ IEEE Access, vol. 8, pp. 172534–172552, 2020.

[5] I. Siomina and D. Yuan, ‘‘Load balancing in heterogeneous LTE: Range optimization via cell offset and load-coupling characterization,’’ in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2012, pp. 1357–1361.

[6] M. Tayyab, X. Gelabert, and R. Jäntti, ‘‘A survey on handover management: From LTE to NR,’’ IEEE Access, vol. 7, pp. 118907–118930, 2019.

[7] V. V. Nimmalapudi, A. K. Mengani, R. Vuppula, and R. Jashvantbhai Pandya, ‘‘Deep learning based load balancing for improved QoS towards 6G,’’ 2020, arXiv:2006.16733.

[8] H. Fourati, R. Maaloul, and L. Chaari, ‘‘A survey of 5G network systems: Challenges and machine learning approaches,’’ Int. J. Mach. Learn. Cybern., vol. 12, no. 2, pp. 385–431, Feb. 2021.

[9] J. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, ‘‘An overview of load balancing in HetNets: Old myths and open problems,’’ IEEE Wireless. Commun., vol. 21, no. 2, pp. 18–25, Apr. 2014.

[10] S. Mishra and N. Mathur, ‘‘Load balancing optimization in LTE/LTE—A cellular networks: A review,’’ 2014, arXiv:1412.7273.

[11] M. A. Gadam, M. A. Ahmed, C. K. Ng, N. K. Nordin, A. Sali, and F. Hashim, ‘‘Review of adaptive cell selection techniques in LTE-advanced heterogeneous networks,’’ J. Comput. Netw. Commun., vol. 2016, pp. 1–12, 2016.

[12] A. S. Milani and N. J. Navimipour, ‘‘Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends,’’ J. Netw. Comput. Appl., vol. 71, pp. 86–98, Aug. 2016.

[13] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, ‘‘Load-balancing algorithms in cloud computing: A survey,’’ J. Netw. Comput. Appl., vol. 88, pp. 50–71, Jun. 2017.

[14] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, ‘‘Load balancing mechanisms in the software defined networks: A systematic and comprehensive review of the literature,’’ IEEE Access, vol. 6, pp. 14159–14178, 2018.

[15] B. Pourghebleh and V. Hayyolalam, ‘‘A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things,’’ Cluster Comput., pp. 1–21, 2019.

[16] A. Chandak and N. K. Ray, ‘‘A review of load balancing in fog computing,’’ in Proc. Int. Conf. Inf. Technol. (ICIT), Dec. 2019, pp. 460–465.

[17] P. Kumar and R. Kumar, ‘‘Issues and challenges of load balancing techniques in cloud computing: A

survey,'' ACM Comput. Surveys, vol. 51, no. 6, pp. 1–35, Nov. 2019.

[18] M. Mehra, S. Maurya, and N. K. Tiwari, ''Network load balancing in software defined network: A survey,'' Int. J. Appl. Eng. Res., vol. 14, no. 2, pp. 245–253, 2019.

[19] M. R. Belgaum, S. Musa, M. M. Alam, and M. M. Su'ud, ''A systematic review of load balancing techniques in software-defined networking,'' IEEE Access, vol. 8, pp. 98612–98636, 2020.

[20] T. Semong, T. Maupong, S. Anokye, K. Kehulakae, S. Dimakatso, G. Boipelo, and S. Sarefo, ''Intelligent load balancing techniques in software defined networks: A survey,'' Electronics, vol. 9, no. 7, p. 1091, Jul. 2020.

[21] M. Hamdan, E. Hassan, A. Abdelaziz, A. Elhigazi, B. Mohammed, S. Khan, A. V. Vasilakos, and M. N. Marsono, ''A comprehensive survey of load balancing techniques in software-defined network,'' J. Netw. Comput. Appl., vol. 174, Jan. 2021, Art. no. 102856.

[22] J. Li, L. Ma, Y. Fu, D. Ma, and A. Xiao, ''Load balancing in heterogeneous network with SDN: A survey,'' in Proc. China Conf. Wireless Sensor Netw. Springer, 2021, pp. 250–261. [23] A. Kumar and D. Anand, ''Study and analysis of various load balancing techniques for software-defined network (a systematic survey),'' in Proc. Int. Conf. Big Data, Mach. Learn. their Appl. Springer, 2021, pp. 325–349

[24] V. Mohammadian, N. J. Navimipour, M. Hosseinzadeh, and A. Darwesh, ''Fault-tolerant load balancing in cloud computing: A systematic literature review,'' IEEE Access, vol. 10, pp. 12714–12731, 2022.

[25] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, ''Load balancing techniques in cloud computing environment: A review,'' J. King Saud Univ. Comput. Inf. Sci., Mar. 2021.

[26] A. A. Alkhatib, A. Alsabbagh, R. Maraqa, and S. Alzubi, ''Load balancing techniques in cloud computing: Extensive review,'' Adv. Sci., Technol. Eng. Syst. J., vol. 6, no. 2, pp. 860–870, Apr. 2021.

[27] Z. Ning, Q. Song, L. Guo, M. Dai, and M. Yue, ''Dynamic cell range expansion-based interference coordination scheme in next generation wireless networks,'' China Commun., vol. 11, no. 5, pp. 98–104, May 2014.

[28] M. G. Calabuig, ''Self-organizing networks,'' M.S. thesis, Universitat Politècnica de Catalunya, 2018.

Authors Biography

Duraimurugan J*[1],

Department of IST, CEG Campus, Anna University, Chennai, India, dmurugan02@gmail.com

Dr. Indra Gandhi K[2], Associate Professor,

Department of IST, CEG Campus, Anna University, Chennai, India,

jeevaindra@gmail.com