International Journal of Communication Networks and Information Security
ISSN: 2073-607X, 2076-0930
Volume 14 Issue 03 Year 2022 Page 187:198

Energy Efficiency Based Load Balancing Optimization Routing Protocol In 5G Wireless Communication Networks

[1] Divya Paikaray, [2] Divyanshi Chhabra, [3] Sachin Sharma, [4] Sachin Goswami, [5] Shashikala H K, [6] Prof. Gordhan Jethava

[1] Assistant Professor, Department of Computer Science, ARKA JAIN University, Jamshedpur, Jharkhand, India

[2] Assistant Professor, Department of Computer Science Engineering, Chandigarh Engineering College, Jhanjeri, India

[3] Assistant Professor, Department of Computer Science and Engineering, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India

[4] Assistant HR Manager, Department of Management, Sanskriti University, Mathura, Uttar Pradesh, India

[5] Assistant Professor, Department of Computer Science and Engineering, Jain (Deemed-to-be University), Bangalore, India

[6] Assistant Professor, Department of Computer Science and Engineering, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.

[1] divya.p@arkajainuniversity.ac.in, [2]divyanshi.j1983@cgc.ac.in

[3]socse.sachin@dbuu.ac.in, [4]hr@sanskriti.edu.in

[5]hk.shashikala@jainuniversity.ac.in, [6]gordhan.jethava@paruluniversity.ac.in

Article History

Received: 13 August 2022
Revised: 24 October 2022
Accepted: 12 November 2022

Abstract

A significant study area in cloud computing that still requires attention is how to distribute the workload among virtual machines and resources. Main goal of this research is to develop an efficient cloud load balancing approach, improve response time, decrease readiness time, maximise source utilisation, and decrease activity rejection time. This research propose novel technique in load balancing based network optimization using routing protocol for 5G wireless communication networks. the network load balancing has been carried out using cloud based software defined multi-objective optimization routing protocol. then the network security has been enhanced by data classification utilizing deep belief Boltzmann NN. Experimental analysis has been carried out based on load balancing and security data classification in terms of throughput, packet delivery ratio, energy efficiency, latency, accuracy, precision, recall.

Keywords- load balancing, network optimization, routing protocol, 5G wireless communication networks, network security

1. Introduction

Cloud computing includes renting hardware and software, paying only for the services that are actually used, and more. Depending on the needs of the customer or client, it refers to delivery of computer hardware, software, and other data technology services to them via a network [1].These services are frequently offered by third-party CC service providers who operate their own data

centres or network infrastructure. The information technology sector has seen the rapid expansion of the cloud computing sector [2].However, security issues with this developing technology continue to be a major source of worry. Attacks on cloud infrastructure that cause economic denial of service (EDoS) are quickly hardening into security issues [3,4]. Hoff and Cohen first used the phrase "economic denial of sustainability" in 2008 [5,6].It was further defined by Cohen in 2011 [7], and the scientific community now generally accepts this definition. Cloud computing infrastructures, which are playing a bigger role in emerging communication technologies, are frequently the target of EDoS assaults.Singh et al[8] .'s definition of DDoS attacks as "threats that strive to render the pricing model unsustainable and, as a consequence, make it difficult for a corporation to financially use or pay for its cloud-based infrastructure" [9,10] reflects this stated definition.EDoS threats are also known as reduction of quality (RoQ) threats and FRC attacks, according to studies. Attackers use computational intelligence techniques to exploit "pay-as-you-go" accounting method and auto-scaling features supplied by the majority of cloud computing companies.

Deep learning and machine learning approaches are widely used in a variety of real-world applications across numerous application domains. Machine learning has recently been used to secure cloud services and applications by safeguarding cloud networks.To guarantee the ongoing availability and accessibility of cloud resources and services to cloud users, cloud security is a crucial necessity. Network assaults may potentially have an impact on cloud service providers and cloud consumers if cloud networks are not safeguarded.Building and installing effective cloud intrusion detection systems (CIDS) in cloud networks at strategic network points is one method of securing them. Cloud load balancing can also be accomplished by the deployment of Cloud IDS in cloud networks by limiting excessively large amounts of network attack traffic and collecting malicious network traffic.For instance, distributed DDoS assaults are the most difficult network attacks in cloud environments because they have an impact not only on cloud services but also on users and providers of cloud services from a technical and financial standpoint.As a result, by safeguarding cloud networks and limiting DDoS network attacks by detection of these network attacks through the implementation of Cloud IDS, cloud resources and services are made accessible for their rightful users.

2. Related works

The authentication and access control policies provide the foundation for a number of techniques that have been introduced. However, in the current environment, where cloud computing systems are widely accessed globally as a result of which the cloud computing systems become more vulnerable, these strategies are not appropriate. [6] focused on using a fuzzy neural network and genetic algorithm to create an intelligent intrusion detection system (IDS). Cluster creation, initial fuzzy rule generation, fuzzy rule base optimization, and system parameter refinement are the four stages that this approach goes through.[7] adopted a semi-supervised machine learning method for robotic cloud systems' intrusion detection. This method employs a 3-layer neural network and a classification and regression tree (CART) to create a hybrid of fuzzy logic and ensemble learning.[8] used the multilayer perceptron artificial bee colony and fuzzy clustering approaches to create a unique IDS solution. Accompanying factors were discussed throughout this research: total allocation ratios of VMs and resources, total time period, time spent co-placing VMs.Researchers [9] also indicated that somehow the side operation element of future studies is the identification of variants of jobs for requirements that can sometimes affect the planned price. The author’s most up-to-date objective hypotheses throughout the analysis paper [10], which will then be conditioned to analyzing the deployment rules for different virtual machines (VMs) in cloud computing conditions.The outcome of the simulation between proposed and current methods shows the overall enchantment with regard to system performance and computational optimization [11]. In the cloud development paper [12] participant predominantly recommended a conceptual mathematical formalism-conscious VM placement process [13].The authors of [14,15] have presented a method that utilises a three-layer cloud computing network and load balancing idea. Authors in [16] present enhanced load balanced Min-Min (ELBMM) method as a resource management method.Another technique suggested in [17] is a Resource-based Load balanced Min-Min (RBLMM). The technique is also designed to balance

workload on virtual machines and take Makespan reduction into account.The authors of [18] suggested effective load balancing and scheduling techniques that would minimise execution time for the advantage of both cloud users and service providers. The suggested technique takes network delay in Data Centers into account and aims to choose the VM with the lowest cost.

3. System model

This section discuss novel method in load balancing based network optimization using routing protocol for 5G wireless communication networks. the network load balancing has been carried out using cloud based software defined multi-objective optimization routing protocol. then the network security has been enhanced by data classification using deep belief Boltzmann neural network.Only the receiving circuit is activated when data is being received, hence the energy required to receive one bit of data is erx = eamp. Since all nodes generate data at the same rate and the energy used for sensing has been evenly distributed across all nodes, we did not take into account energy usage for data sensing in this study.This model states that etx(d) = eelec + eamp d k, where eelec is the energy consumed by the transmitter electronics, eamp is the energy used by the transmitting amplifier, and k(k2) is the propagation loss exponent, gives the energy needed to transmit a single data bit across a distance d.The energy lost during data aggregation is significantly less than that lost during data communication, so it is not taken into account.

Cloud based software defined multi-objective optimization routing protocol:
The SDN structure that offers application-oriented flexibility, such as load balancing, is depicted in Fig. 1. The SDN framework aids in application modification, which results in meeting any heterogeneous service requirement.One of the main requirements of heterogeneous applications is intelligent load balancing due to the increased volume of data.

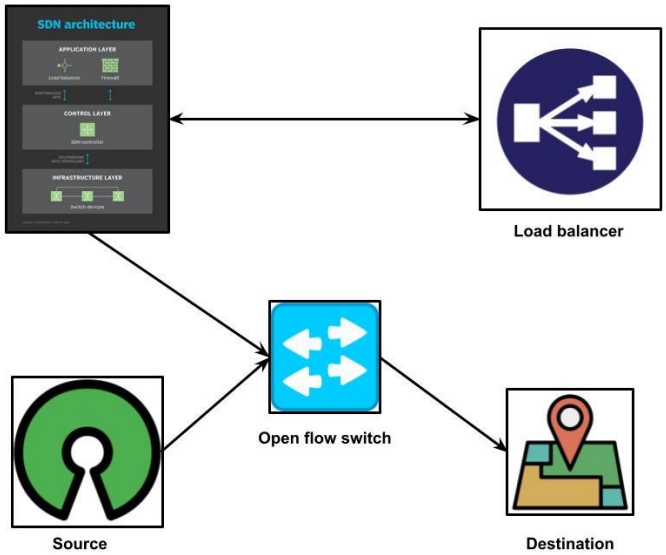

Figure-1 *Proposed SDN load-balancing architecture*

We write M = {β1, β2, · · · , βM} to represent the set of numerous behaviour policies to adhere to, where M is the total number of behaviour policies. Objective under many behaviour policies could be expressed as eq based on objective under a single behaviour policy provided in (12).

$$J(\pi_\theta) = \sum_{m\in\mathcal{M}} J_{\beta_m}(\pi_\theta) \quad (1)$$

$$J_{\beta_m}(\pi_\theta) = \mathbb{E}_{s\sim\kappa^{3m}}[\sum_{k=0}^{\infty}\gamma^k r(s,\pi_\theta(s))] \quad (2)$$

with κ βm(s) := R S P∞ t=1 γ t−1p1(s)p(s → s ′ , t, β)ds. New objective J(πθ) is thought of as target policy's value function averaged over states produced by adhering to various behaviour policies. The MLB issue within the cluster is now caused by eq (3)

$$\mathcal{P}_1: \quad \max_\theta J(\pi_\theta) = \sum_{m\in\mathcal{M}} J_{\beta_m}(\pi_\theta)$$
$$\text{s.t.} \quad C_1: \mathcal{X}_{u,i}^t \in \{0,1\}, \sum_{i\in\mathcal{I}} \mathcal{X}_{u,i}^t \le 1, \forall u \in \mathcal{U} \quad C_2: O_{ij} \in [O_{\min}, O_{\max}], \forall i,j \in \mathcal{I}. \tag{3}$$

As a result, eq (4) is used to represent policy gradient for learning under many behaviour policies

$$\nabla_\theta J(\pi_\theta) = \sum_{m\in\mathcal{M}} \nabla_\theta J_{\beta_m}(\pi_\theta)$$
$$\approx \sum_{m\in\mathcal{M}} \int_S \rho^{\beta_m}(s) \nabla_\theta \pi_\theta(s) \nabla_a Q^\pi(s,a)\big|_{a=\pi_\theta(s)} \mathrm{d}s$$
$$= \sum_{m\in\mathcal{M}} \mathbb{E}_{s\sim\rho^{im}}\left[\nabla_\theta \pi_\theta(s) \nabla_a Q^\pi(s,a)|_{a=\pi_\theta(s)}\right] \tag{4}$$

Second, based on Q-learning, each local critic assesses Q-function of the target policy [24]. Gradient m w for the Q-function updates are expressed specifically as eq (5)

$$\delta_t^m = r_t^m + \eta Q^w\left(s_{t+1}^m, \pi_\theta(s_{t+1}^m)\right) - Q^w(s_t^m, a_t^m) \tag{5}$$

The enhanced OPDPG theorem is used by every local actor to compute policy gradient in step three (26). Target policy updates' gradient, Δm θ is expressed mathematically as eq. (6)

$$\Delta_w^m = \delta_t^{mi} \nabla_w Q^w(s_t^m, a_t^m)$$
$$\Delta_\theta^m = \nabla_\theta \pi_\theta(s_t^m) \nabla_a Q^w(s_t^m, a)|_{a=\pi\theta(sv_t^m)} \tag{6}$$

Gradient of MSE based on w should satisfy equation if w can reduce the MSE (7)

$$\nabla_w \text{MSE}\ (\theta, w) = 2\mathbb{E}_{s\sim p^s}[\nabla_w \epsilon(s;\theta,w)\epsilon(s;\theta,w)]$$
$$= 2\mathbb{E}_{s\sim p^{\theta m}}[\nabla_\theta \mu_\theta(s)\epsilon(s;\theta,w)] = 0. \tag{7}$$

According to definition of ϙ(s; θ, w), we have by eq. (8)

$$\mathbb{E}_{s\sim\rho^*\ldots}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^w(s,a)|_{a=\mu_\theta(s)}\right]$$
$$= \mathbb{Z}_{s\sim\rho^s m}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right]$$
$$= \nabla_\theta J_\beta m(\mu_\theta), \tag{8}$$

The user association (UA) problem is expressed as eq. (9) taking into account all of the aforementioned variables

$$\min_\rho \sum_{j\in B} w_j(\rho_j) L(\rho_j)$$
$$\text{subjectto:} \quad 0 \le \rho_j \le 1 - \epsilon\left(\alpha_j(x) - \alpha^*(x)\right)\eta_j(x) \le 0$$
$$\forall x \in \mathcal{A}, j \in \mathcal{B}. \tag{9}$$

Uplink pathloss from user at position x to jth BS and user's uplink pathloss threshold are represented here by the variables αj (x) and α ∗ (x), respectively. To guarantee that j 1, the real number 0 1 is used. ρj< 1. ρ = (ρ1, ρ2, · · · ,ρ|B|), and by eq (10)

$$w_j(\rho_j) = e^{\kappa(\rho_j - T_j)}$$
$$= e^{\kappa(\rho_j - (1-\theta_j)\rho_j - \theta_j\hat{\beta}_j)}$$
$$= e^{n\theta_j(\rho_j - \hat{\beta}_j)} \tag{10}$$

$$\frac{dL(\rho_j)}{d\rho_j} = \frac{v_j}{(1-\rho_j)^2} > 0 \tag{11}$$

In order to conserve on-grid electricity in jth BS, a smaller delay indicator indicates lower traffic loads in jth BS. The user association for traffic delivery latency and green energy awareness is thus made possible by the introduction of weights for BSs' latency indication in objective function.Virtual

users choose vBSs based on user side method after obtaining updates on the operation state of vBSs. Then, jthvBS's coverage area for kth time slot is updated as equal. (12)

$$\bar{\mathcal{A}}_j(k) = \{x \mid j = b^k(x), \forall x \in \mathcal{A}\} \quad (12)$$

$$M_j(\rho(k), \theta, \hat{\rho}) = \min\left(\int_{x \in \hat{\mathcal{A}},(k)} e_j(x)dx, 1 - \epsilon\right) \quad (13)$$

Assuming that and do not change during the course of a user association procedure, Mj (ρ(k), θ, ρˆ) evolves solely dependent on (k). For simplicity's sake, we will therefore use Mj (k)) rather than Mj (ρ(k), θ, ρˆ) in analysis that follows. jthvBS updates its traffic loads as equal after deriving the perceived traffic loads (14)

$$\rho_j(k+1) = \delta(k)\rho_j(k) + \big(1 - \delta(k)\big)M_j\big(\rho(k)\big) \quad (14)$$

Here, $0 \leq \delta(k) < 1$ is a system parameter evaluated by RANC to enable by eq. (15)

$$\psi(\rho(k+1)) \leq \psi\big(\rho(k)\big) + \varsigma\big(1 - \delta(k)\big)\sum_{j \in B} \phi_j\left(\rho_j(k)\right)\left(M_j\big(\rho(k)\big) - \rho_j(k)\right) \quad (15)$$

Here, $0 < \varsigma < 0.5$ is a constant.

The paper's Fog network architecture for the suggested system is shown in Fig. 1. In terms of computing capabilities and task demand distributions, this study discusses a dynamic load balancing strategy utilizing distributed fog computing where nodes can offload their calculation duties to a neighbouring node with available queue spaces.A load index, which offers a measurement of the workload at a node relative to some global average, is typically foundation of load balancing algorithms. When load index at one node is much higher or significantly lower than load index at other nodes, a load imbalance state is present, and this state is detected by the load index.

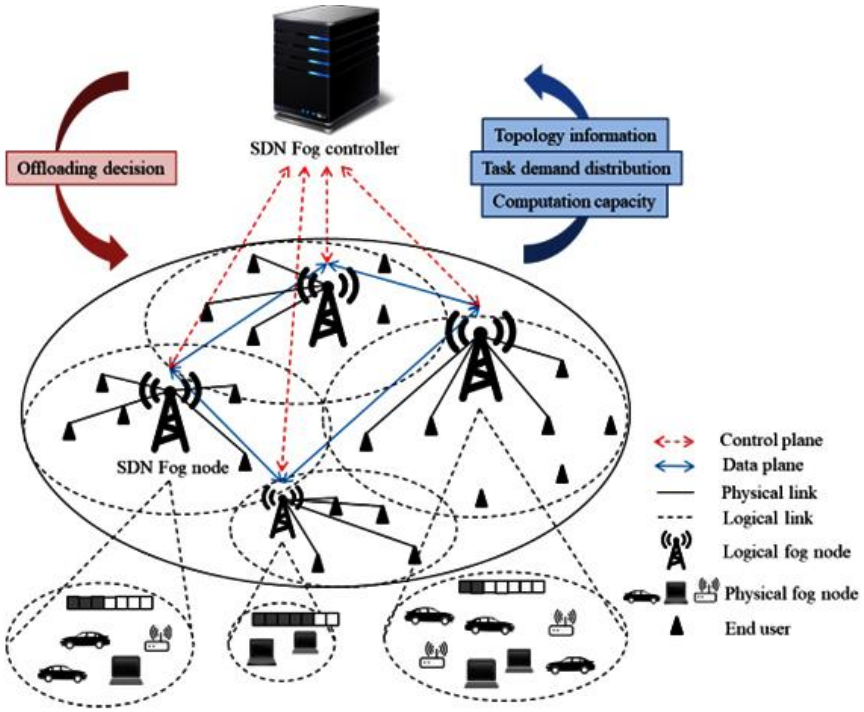

Figure- 2 *software defined multi-objective optimization architecture*

When the target performance metric is the average response time, it has been demonstrated that length of CPU queue provides a useful load index for the remaining resources on time-shared workstations.When target performance metric is average response time, it has been demonstrated

that length of CPU queue provides a useful load index for the remaining resources on time-shared workstations.While the nodes work directly with end users to provide services and transmit data based on gathered traffic data and queue status to controllers, controller is in charge of information aggregation and decision-making. This is how the system functions.The controller first creates a network map using the queue data sent by the nodes. The controller then employs algorithms to decide whether or not the current node should offload its requested duties. Main goal is to select best offloading strategies while minimising task processing time as well as node overload risk.

Deep belief Boltzmann neural network based data security analysis:

DBN can capture a hierarchical representation of incoming data due to its deep structure. The joint distribution is defined as eq(16) given a visible unit of x and l hidden layers.

$$\min_{\theta_L,\theta_{DBN}} \mathbb{E}_{y,x}[\mathcal{L}(\theta_L; \mathbf{y}, h(\mathbf{x}))] + \rho \mathbb{E}_{\mathbf{x}}[-\log\, p(\mathbf{x}; \theta_{DBN})] \quad (16)$$

Training a DBN's layers is equivalent to training an RBM because each layer is built as an RBM. Through DBN training, a network is initialised to do classification. Pre-training step solves following optimization problem at every layer k by eq given training dataset D = $\{(x^{(1)^\circ}, y^{(1)}), \ldots, (x^{(|D|)^\circ}, y^{(|D|)})\}$ with input x and label y. (17)

$$\min_{\theta_L,\theta_{DBN}} \frac{1}{|D|}\sum_{i=1}^{|D|} \left[\mathcal{L}\left(\theta_L; y^{(i)}, h\left(\theta_{DBN}; x^{(i)}\right)\right)\right] \quad (17)$$

Keep in mind that layer-wise updating requires us to fix all of issues from bottom hidden layer to top visible layer. We use eq(18) to solve the following optimization problem at the fine-tuning stage

$$\min_{\phi} \frac{1}{|D|}\sum_{i=1}^{|D|} \left[\mathcal{L}\left(\phi; y^{(i)}, h\left(x^{(i)}\right)\right)\right] \quad (18)$$

where h stands for the final hidden features at layer l, L() is a loss function, l and φ are the classifier's parameters. Here, we'll write $h(\bar{x}^{(i)}) = h\left(x_\ell^{(i)}\right)$ for simplicity. To begin, we use a simplistic model of aggregating training and fine-tuning goals. The definition of model (DBN+loss) is eq (19)

$$\min_{\theta_L,\theta_{DBN}} \mathbb{E}_{y,x}[\mathcal{L}(\theta_L; \mathbf{y}, h(\mathbf{x}))] + \rho \mathbb{E}_{\mathbf{x}}[-\log\, p(\mathbf{x}; \theta_{DBN})] \quad (19)$$

and empirically based on training samples D by eq. (20),

$$\min_{\theta_L,\theta_{DBN}} \frac{1}{|D|}\sum_{i=1}^{|D|} \left[\sum_h p\left(h \mid x^{(i)}\right)\mathcal{L}\left(\theta_L; y^{(i)}, h\left(\theta_{DBN}; x^{(i)}\right)\right)\right] \quad (20)$$

where the underlying parameters are θ_L, θ_{DBN}. Take note that (1)'s $\theta_L = \phi$ from (1) and $\theta_{DBN} = (\bar{\theta}_k)_{k=1}$.On the basis of the conditional distribution p(h|x) derived by DBN, we first create an anticipated loss model. Hidden space is classified utilizing this paradigm. It should be more reliable and, as a result, produce better accuracy on data that hasn't been observed because it minimises the projected loss. Equation (eq) designates the mathematical method that minimises anticipated loss function (21)

$$\min_{L,\theta_{DBN}} \mathbb{E}_{\mathbf{y},\mathbf{h}|\mathbf{x}}\left[\mathcal{L}\left(\theta_L; \mathbf{y}, h(\theta_{DBN}; \mathbf{x})\right)\right. \quad (21)$$

and empirically based on training samples D eq. (22),

$$\min_{\theta_L,\theta_{DBN}} \frac{1}{|D|}\sum_{i=1}^{|D|} \left[\sum_h p\left(h \mid x^{(i)}\right)\mathcal{L}\left(\theta_L; y^{(i)}, h\left(\theta_{DBN}; x^{(i)}\right)\right)\right] \quad (22)$$

By using the notation $h\left(\theta_{DBN}; x^{(i)}\right) = h\left(x^{(i)}\right)$, we clearly demonstrate how dependent h is on θDBN. By adding a constraint that places constraints on DBN-related specifications with regard to

their ideal values, we modify the predicted loss model. This model offers two advantages. In the beginning, the model limits the parameters that were fitted in an unsupervised way to maintain a good representation of the input. The restriction also regularises the model parameters by avoiding their explosion during updating. Mathematical form of model readsbyeq(23) given training samples D.

$$\min_{\theta_L,\theta_{DBN}} \frac{1}{|D|}\sum_{i=1}^{|D|}\left[\sum_h p\left(h \mid x^{(i)}\right)\mathcal{L}\left(\theta_L; y^{(i)}, h\left(\theta_{DBN}; x^{(i)}\right)\right)\right] \quad (23)$$

$$\text{s.t.}|\theta_{DBN} - \theta^*_{DBN}| \le \delta$$

where δ is a hyperparameter, θ ∗ DBN are the ideal DBN parameters. For the model to produce the DBN fitted parameters, a pre-training phase is required. With parameters that are fitted using both supervised and unsupervised methods, this model regularises them. Therefore, even if we require an additional training in addition to two-phase trainings, it can still attain improved accuracy. The model reads by eq(24) given training samples D.

$$\min_{\theta_{L,\theta_{DBN}}} \frac{1}{|D|}\sum_{i=1}^{|D|}\left[\sum_h p\left(h \mid x^{(i)}\right)\mathcal{L}\left(\theta_L; y^{(i)}, h\left(\theta_{DBN}; x^{(i)}\right)\right)\right] \quad (24)$$
$$\text{s.t.}|\theta_{DBN} - \theta^*_{DBN-OPT}| \le \delta$$

where δ is a hyperparameter and $\theta^*_{DBN-OPT}$ are the optimal DBN parameter values following two phases of training. Equation provides mathematical model FFNDBN based on training samples D. (25)

$$\min_{\theta_L,\theta_{DBN}} \quad \frac{1}{|D|}\sum_{i=1}^{|D|}\left[\mathcal{L}\left(\theta_L; y^{(i)}, h\left(\theta_{DBN}; x^{(i)}\right)\right)\right] \quad (25)$$
$$\text{s.t.} \quad |\theta_{DBN} - \theta^*_{DBN}| \le \delta.$$

With respect to training samples D, this model, which combines (26), reads

$$\min_{\theta_L,\theta_{DBN}} \frac{1}{|D|}\sum_{i=1}^{|D|}\left[\mathcal{L}\left(\theta_L; y^{(i)}, h\left(\theta_{DBN}; x^{(i)}\right)\right)\right] \quad (26)$$

$$\text{s.t.}|\theta_{DBN} - \theta^*_{DBN-OPT}| \le \delta.$$

The concealed layer's jth output layer is defined as eq. (27)

$$h_j = \sum_{i=1}^{n_H} w_{ji}x_i \quad j = 1,2,\dots,n_2 \quad (27)$$

where wji is the hidden weight, n2 is number of nodes in hidden layer, and n1 is number of nodes in the input layer. NN model's input vector is $x = [u(k), u(k-1), u(k-2), \dots]^T$.
where u(k) is output of NN controller. Eq. (28) that follows gives the neural network model's output:

$$yr(k+1) = \lambda_s\left(\sum_{j=1}^{n_2} w_{1j}s\left(h_j\right)\right) \quad (28)$$

where λ is a scaling coefficient and w1j is output weight. Following equation (29) gives the output's compact form:

$$yr(k+1) = \lambda s(h_1) = \lambda s[w_1^T S(Wx)] \quad (29)$$

With eq. (30)

$$x = [x_i]^T, i = 1, \dots, n_1$$

$$W = [w_{jl}], i = 1, \dots, n_1, j = 1, \dots, n_2 \quad (30)$$

$$S(Wx) = [s(h_j)]^T, j = 1, \dots, n_2$$

$$w_1 = [w_1]^T, j = 1, \dots, n_2$$

Identification error e(k) is given by eq. (31)

$$e(k) = y(k) - yr(k) \quad (31)$$

Function cost is given by following eq. (32):

$$E = \frac{1}{2}(e(k))^2 \quad (32)$$

N is the total number of observations. The following equation (33) updates the output weights:

$$w_{11}(k+1) = w_{1j}(k) + \Delta w_{1y}(k) \quad (33)$$

where $\Delta w_{ji}(k), j = 1, \dots . n$ is given by minimizing cost function given as eq. (34):

$$\Delta w_{1j} = -\eta(k)\frac{\partial E(k)}{\partial w_{1j}} = -\eta(k)\frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial h_1}\frac{\partial h_1}{\partial w_{1j}} = \lambda\eta(k)e(k)S(h_1)S(W_x) \quad (34)$$

$\eta(k)$ is variable learning rate for weights of NN model, $0 \leq \eta(k) \leq 1$, given by eq. (35)

$$\eta(k) = \frac{1}{\lambda^2 y^2(k_1)[S^f(W_s)S(W_x) + w_1^r, S(W_x)S(W_x)m_1 e^r x x]} \quad (35)$$

$s'(h_1)$ is derivative of $s'(h_1)$ defined as eq. (36):

$$s'(h_1) = s(h_1)(1 - s(h_1)) = \frac{e^{-h_1}}{(1+e^{-h_i})^2} \approx \frac{1}{4} + \frac{1}{2}h_1 + O(h_1^3) \quad (36)$$

The following equation (37) updates the hidden weights:

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k) \quad (37)$$

where Δwji is given by following eq. (38):

$$\Delta w_{ji} = -\eta(k)\frac{\partial E(k)}{\partial w_{ji}}$$

$$= -\eta(k)\frac{\partial E(k)}{\partial e(k)}\frac{\partial e(k)}{\partial h_1}\frac{\partial h_1}{\partial h_j}\frac{\partial h_j}{\partial w_{jk}} \quad (38)$$

with $S'(Wx) = \text{diag}[s'(h_j)]^T, j = 1, \dots, n_2$.

At the time of algorithm execution, sensors, diverse application data, and network topology are all inputs. Following that, the application is requested that is mapped in accordance with the minimal task fulfilment by each sensor.The load balancer and resource selector are the two components of the balancer. The sensors' load for the required data is provided by the SDN controller. We choose the sensor with the highest load. Then, the data collection activity that requires the least amount of time to complete is chosen as a migrating task. The focus then shifts to the sensors with a load that is less than (total load / number of sensors) and that have the least time necessary for finishing crucial application data.The cycle continues until the data is migrated if the SDN controller is unable to locate any machines, at which point the end minimum completion time sensor is chosen. After the migration, the chosen sensor no longer receives the necessary data request. We eliminate specific

sensor data from the load balancing operation in this way.This process keeps going till all the machines are distributed with the data after which the resource usage ratio is determined and compared. The primary cycle must be repeated if, at its conclusion, the network is still in an unbalanced state.The simulation of the method described [6] reveals unexpected oscillations in the output of the balancing process. A mobile station is unconnected from one AP, linked to another, and then reconnected to the first AP during the balancing process.Such oscillations are particularly common when attempting load-balancing in situations with few nodes and when there are nodes that are utilising more bandwidth than the other mobile stations.In order to prevent this issue, we programmed the algorithm to terminate after four rounds of its main loop. By doing so, the Controller was given the ability to recognise these fluctuations and take them out of the balancing output.

Performance analysis:

In the simulated SON scenario, there are 200 users walking at random speeds of 1 to 10 m/s, creating a CBR traffic demand for each of 12 SBSs randomly dispersed throughout a 300 m by 300 m area. Each SBS is programmed with a transmit power of 46 dBm. Each user's path loss to a specific SBS is calculated as 128.1+37.6 log, where d is distance in kilometres. Modeling additional log-normal shadowing results in a zero mean and an 8 dB standard deviation.3 dB handover hysteresis is default value. Each of the two hidden layers that make up actor-net and critic-net has 400 and 300 neurons, respectively. A rectified non-linearity layer comes after every concealed layer. Actor and critic networks have fixed learning rates of 104 and 103, respectively. For awards, the discount factor is set at 0.99.Typically, it takes 30 minutes to train a single DRL model with 10,000 time steps. Due to the wide variety of task lengths taken into account in the experiment, the graph demonstrates that Makespan in our proposed algorithm increases in case of 25–40 tasks.Suggested LB algorithm can manage requests for jobs with a length of 1,000,000 MI, whereas Dynamic LBA can only handle tasks with a length of 400,000 MI. Because Makespan is based on the load on VMs, lengthening tasks will also lengthen Makespan.

When the client's request is received, a job is started. The system executes the binary code in response to this request and then communicates the outcome to the client over the network. There are distinct computer devices on each cloud host. In contrast to host server B, which has Xeon 3.0 GHz dual core CPUs and 2.0 GB of RAM, host server A has Intel Xeon 2.4 GHz dual core CPUs.Last but not least, host C contains Xeon 2.8 GHz dual core CPUs with 1.0 GB of RAM. The computers are running Linux, with CPU throttling enabled and on-demand governor, which dynamically modifies cores' frequency in response to load. Additionally, each server handles jobs in accordance with FCFS scheduling principle. The parameters utilized in simulation are displayed in table.1 below.

Table.1 *Simulation Parameters*

Parameters	**Values**
Simulation time	1000 seconds
Allotted area	40m*2500m
Number of devices	500
Packet size	1024 bits
PDR	I packet/sec
Initial energy	100J
Channel capacity	2MBPS

Table-2 shows comparative analysis between proposed and existing technique based on network load balancing and security analysis. here the network analysis has been carried out in terms of throughput, packet delivery ratio, energy efficiency, latency, accuracy, precision, recall.

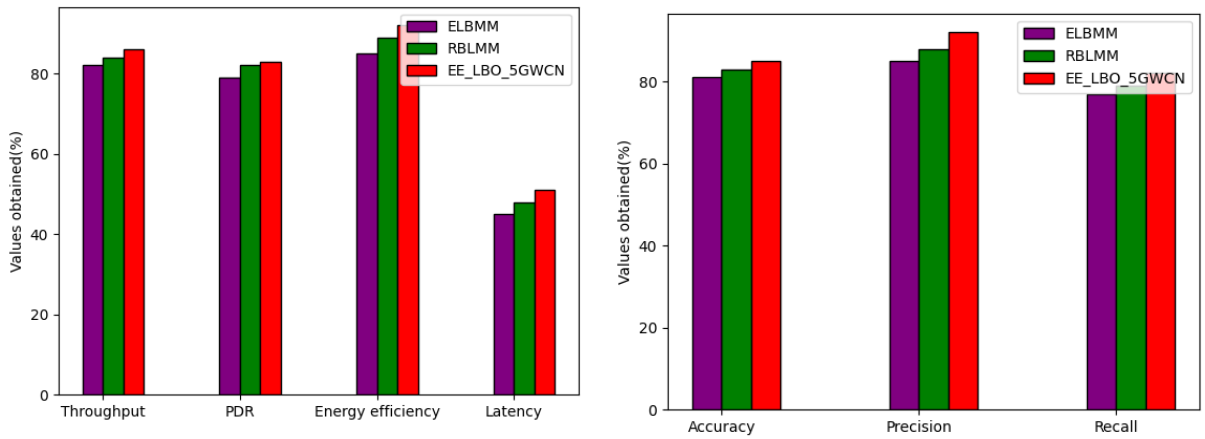

Figure- 3 *Analysis based on network load balancing in terms of throughput, packet delivery ratio, energy efficiency, latency, accuracy, precision, recall*

Figure-3 gives analysis based on network load balancing. Proposed technique attained throughput of 86%, packet delivery ratio 83%, energy efficiency of 92%, latency of 51%, accuracy of 85%, precision of 92%, recall of 82%; existing ELBMM attained throughput of 82%, packet delivery ratio 79%, energy efficiency of 85%, latency of 45%, accuracy of 81%, precision of 85%, recall of 77%, RBLMM attained throughput of 84%, packet delivery ratio 82%, energy efficiency of 89%, latency of 48%, accuracy of 83%, precision of 88%, recall of 79%

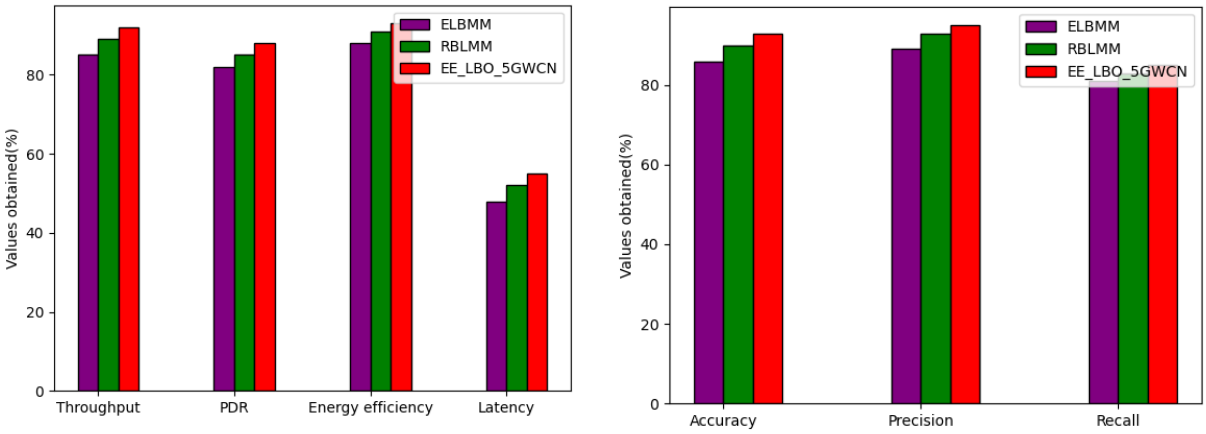

Figure- 3 *Analysis based on Security data classification in terms of throughput, packet delivery ratio, energy efficiency, latency, accuracy, precision, recall*

Figure-4 analysis based on Security data classification is shown. the proposed technique attained throughput of 92%, packet delivery ratio 88%, energy efficiency of 93%, latency of 55%, accuracy of 93%, precision of 95%, recall of 85%; existing ELBMM attained throughput of 85%, packet delivery ratio 82%, energy efficiency of 88%, latency of 48%, accuracy of 86%, precision of 89%, recall of 81%, RBLMM attained throughput of 89%, packet delivery ratio 85%, energy efficiency of 91%, latency of 52%, accuracy of 90%, precision of 93%, recall of 83%.

4. Conclusion:

This research propose novel technique in load balancing based network optimization using routing protocol for 5G wireless communication networks. We think that there should be a balance between energy use and area. In 5G networks, the IoT devices are enmeshed with a variety of MIMO transmission interfaces.An efficient clustering strategy for quickly growing IoT systems is both absent and urgently required to handle a variety of user situations now that MIMO is more frequently available on IoT devices.Network load balancing has been carried out using cloud based software defined multi-objective optimization routing protocol where security is enhanced by data

classification using deep belief Boltzmann neural network. the proposed technique attained throughput of 92%, packet delivery ratio 88%, energy efficiency of 93%, latency of 55%, accuracy of 93%, precision of 95%, recall of 85%.

Reference

[1] N. N. Dao, Q. V. Pham, N. H. Tu et al., "Survey on aerial radio access networks: toward a comprehensive 6G access infrastructure," IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1193–1225, 2021.

[2] B. Shang, Y. Yi, and L. Liu, "Computing over space-air-ground integrated networks: challenges and opportunities," IEEE Network, vol. 35, no. 4, pp. 302–309, 2021.

[3] Q. Wang, X. Li, Y. Liu, L. T. Alex, S. A. Khowaja, and V. G. Menon, "UAV-enabled non-orthogonal multiple access networks for ground-air-ground communications," IEEE Transactions on Green Communications and Networking, p. 1, 2022.

[4] X. Deng, J. Shao, L. Chang, and J. Liang, "A blockchain-based authentication protocol using cryptocurrency technology in LEO satellite networks," Electronics, vol. 10, no. 24, p. 3151, 2021.

[5] X. Li, Y. Zheng, W. U. Khan et al., "Physical layer security of cognitive ambient backscatter communications for green Internet-of-Things," IEEE Transactions on Green Communications and Networking, vol. 5, no. 3, pp. 1066–1076, 2021

[6] Z. Xie, Z. Chu, V. G. Menon, S. Mumtaz, and J. Zhang, "Exploiting benefits of IRS in wireless powered NOMA networks," IEEE Transactions on Green Communications and Networking, vol. 6, no. 1, pp. 175–186, 2022.

[7] G. Li, H. Liu, G. Huang, X. Li, B. Raj, and F. Kara, "Effective capacity analysis of reconfigurable intelligent surfaces aided NOMA network," EURASIP Journal on Wireless Communications and Networking, vol. 2021, 16 pages, 2021.

[8] X. Xiang, Y. Tian, X. Zhang, J. Xiao, and Y. Jin, "A pairwise proximity learning-based ant colony algorithm for dynamic vehicle routing problems," IEEE Transactions on Intelligent Transportation Systems, vol. PP(99), pp. 1–12, 2021.

[9] M. Rathee, S. Kumar, A. H. Gandomi, K. Dilip, B. Balusamy, and R. Patan, "Ant colony optimization based quality of service aware energy balancing secure routing algorithm for wireless sensor networks," IEEE Transactions on Engineering Management, vol. 68, no. 1, pp. 170–182, 2021.

[10] W. Ni, Z. Xu, J. Zou, Z. Wan, and X. Zhao, "Neural network optimal routing algorithm based on genetic ant colony in IPv6 environment," Computational Intelligence and Neuroscience, vol. 2021, 3115713 pages, 2021.

[11] Y. Zhao, K. Liu, X. Xu, H. Yang, and L. Huang, "Distributed dynamic cluster-head selection and clustering for massive IoT access in 5G networks," Applied Sciences, vol. 9, no. 1, article 132, 2019.

[12] C. Jothikumar, K. Ramana, V. D. Chakravarthy, S. Singh, and I. H. Ra, "An efficient routing approach to maximize the lifetime of IoT-based wireless sensor networks in 5G and beyond," Mobile Information Systems, vol. 2021, 11 pages, 2021.

[13] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "Residual energybased cluster-head selection in WSNs for IoT application," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5132–5139, 2019.

[14] A. Seyyedabbasi and F. Kiani, "MAP-ACO: an efficient protocol for multi-agent pathfinding in real-time WSN and decentralized IoT systems," Microprocessors and Microsystems, vol. 79, article 103325, 2020.

[15] D. Sharma, S. Singhal, A. Rai, and A. Singh, "Analysis of power consumption in standalone 5G network and enhancement in energy efficiency using a novel routing protocol," Sustainable Energy, Grids and Networks, vol. 26, article 100427, 2021.

[16] V. Nivedhitha, A. G. Saminathan, and P. Thirumurugan, "DMEERP: a dynamic multi-hop energy efficient routing protocol for WSN," Microprocessors and Microsystems, vol. 79, article 103291, 2020.

[17]] K. M. Kumaran and M. Chinnadurai, "A competent ad-hoc sensor routing protocol for energy efficiency in mobile wireless sensor networks," Wireless Personal Communications, vol. 116, no. 1, pp. 829–844, 2021.

[18] Deng, X., Zeng, S., Chang, L., Wang, Y., Wu, X., Liang, J., ... & Fan, C. (2022). An Ant Colony Optimization-Based Routing Algorithm for Load Balancing in LEO Satellite Networks. *Wireless Communications and Mobile Computing*, *2022*.