International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

Secure Multi-Path Selection with Optimal Controller Placement Using Hybrid Software-Defined Networks with Optimization Algorithm

[1*]Sanjay Shahaji Kadam, [2]Jotiram Krishna Deshmukh, [3]Ankush Madhukar Gund, [4]Sudam Vasant Nikam, [5]Vijay Balaso Mane, [6]Dr. Dayanand Raghoba Ingle

[1]Research Scholar, Bharati Vidyapeeth College of Engineering, Navi Mumbai
Maharashtra, India
Email: sanjaykadam23@gmail.com
[2]Associate Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai
Maharashtra, India
Email: jkdeshmukh75@gmail.com
[3]Research Scholar, Bharati Vidyapeeth College of Engineering, Navi Mumbai
Maharashtra, India
Email: ankush.gund@bvcoenm.edu.in
[4]Assistant Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai
Maharashtra, India,
Email: sudam.nikam@bharatividyapeeth.edu
[5]Assistant Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai
Maharashtra, India
Email: vijay.mane@bharatividyapeeth.edu
[6]Professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai
Maharashtra, India
Email: dringleus@gmail.com

Abstract: The Internet's growth in popularity requires computer networks for both agility and resilience. Recently, unable to satisfy the computer needs for traditional networking systems. Software Defined Networking (SDN) is known as a paradigm shift in the networking industry. Many organizations are used SDN due to their efficiency of transmission. Striking the right balance between SDN and legacy switching capabilities will enable successful network scenarios in architecture networks. Therefore, this object grand scenario for a hybrid network where the external perimeter transport device is replaced with an SDN device in the service provider network. With the moving away from older networks to SDN, hybrid SDN includes both legacy and SDN switches. Existing models of SDN have limitations such as overfitting, local optimal trapping, and poor path selection efficiency. This paper proposed a Deep Kronecker Neural Network (DKNN) to improve its efficiency with a moderate optimization method for multipath selection in SDN. Dynamic resource scheduling is used for the reward function the learning performance is improved by the deep reinforcement learning (DRL) technique. The controller for centralised SDN acts as a network brain in the control plane. Among the most important duties network is selected for the best SDN controller. It is vulnerable to invasions and the controller becomes a network bottleneck. This study presents an intrusion detection system (IDS) based on the SDN model that runs as an application module within the controller. Therefore, this study suggested the feature extraction and classification of contractive auto-encoder with a triple attention-based classifier. Additionally, this study leveraged the best performing SDN controllers on which many other SDN controllers are based on OpenDayLight (ODL) provides an open northbound API and supports multiple southbound protocols. Therefore, one of the main issues in the multi-controller placement problem (CPP) that addresses needed in the setting of SDN specifically when different aspects in interruption, ability, authenticity and load distribution are being considered. Introducing the scenario concept, CPP is formulated as a robust optimization problem that considers changes in network status due to power outages, controller's capacity, load fluctuations and changes in switches demand. Therefore, to improve network performance, it is planned to improve the optimal amount of controller placements by simulated annealing using different topologies the modified Dragonfly optimization algorithm (MDOA).

Keywords: SDN, Controller Placement, ODL, Multi-Path Transmission, Security, Intrusion Detection, Dynamic Scheduling.

I. INTRODUCTION

Due to transmission efficiency, the SDN has been employed in several organizations. SDN uses machine learning techniques to progress the capability of resource scheduling [1]. The balance exploration and utilization in the multi-path selection process is applied for Gaussian distribution in the balancing module [2].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

The various kinds of facility flows in multipath routing may work out the separate necessities in terms of bandwidth, obstruct, and other quality of service (QoS) metrics, but different networks have different multipath detection, the establishment of the network efficiency and also differ in selection strategies. Resources among multiple users can also be different in Allocation and utilization [3]. This network customs SDN protocols and traditional networking, with an approach to HSDN operating in the same conditions [4]. Controller software applications are delivered over SDN instead of hardware [5]. HSDN has enabled network engineers to generate new SDN technologies to support switched fabrics across application-specific integrated circuits and multi-vendor hardware [6]. Network security is responsible for IDS have been developed to detect the occurrence of attack streams and notify the network system administrators [7]. An extrapolation model of IDS is used to identify whether network traffic is normal or aggressive. Due to classical IDS's shortcomings in sophisticated combat attacks and security enhancements, intrusion detection utilizing ML and DL is gaining more attention [8].

Investigated the research community is used for malware detection, network intrusion detection, and botnet attack detection in ML techniques [9]. It leverages the standards-based connectivity of the ODL Virtualization Platform for Southbound Protocol, which defines the control platform between the communication regulator and information plane devices (such as virtual and physical switches). In an SDN-enabled network, the location of organizers and the routing of the accompanying switches to those controllers are typically required to achieve objectives like reducing latency, boosting dependability, and maximizing energy efficiency. [11].The problem of controller placement has received a great deal of research interest because it has a significant impact on almost all aspects of SDN, the network performance has state distribution options to fault tolerance [12]. An important subfield of optimization that deals with data uncertainties in optimization problems in Robust optimization. This framework only assumes that the impartial and constraint procedures belong to a definite set in the function space [13]. Currently, the optimization community was intended by Robust optimization the methodology for dealing with the uncertainty of data [14]. This consists of accounting for data uncertainty in the procedure of additional constraints included in the model and excluding solutions that may not be viable or optimal when there is variation in the entered data [15]. Therefore, the purpose research is to analyze safe multipath selection with optimal controller placement using optimization algorithms of hybrid software-defined networks. The remainder of this research is composed as follows: Section 2 provides a literature inspection of the work, Section 3 grants the problem definition and research motivation, and Section 4 the offered research methodology is described. Section 5 offerings a conversation of experiments and results, and Section 6 presents the research termination.

II. LITERATURE SURVEY

Liu et al[16] research Established on Fountain Code – FMPST to obtain Multipath Transmission Algorithm. Investigations have revealed that FMPST can recover data safety by three times more and decrease communication latency by at least 50% compared to other multipath transmission algorithms. Rohitaksha et al [17] exhibit an efficient technique to find the lowest track among either two nodes in an HSDN by reducing a complex network topology to a basic topology while considering the figure of hop count. Mininet's emulation test environment was set up for several various nodes and anatomized the results. Firouz et al [18] introduced the controllers has been installed in the best places. The comparison shows the ascender of the suggested algorithm of controller placement. Sun et al .'s [19] investigation of the CNN-GRU deep learning model-based LDoS attack detection technique. The gated frequent unit and the convolutional neural network (CNN). Demonstrates the experimental results that the contemplated method based on the model of hybrid deep learning outperforms learning algorithms in terms of recognition efficiency and accuracy for another traditional machine. Kaul et al [20] considered hybrid optimization strategies, LION and WHALE, to analyse strategies in the cloud in offline mode, where the computer was successfully installed and programmed to meet the user's needs. The recommended changes increased latency and decreased throughput.

Yan et al [21] studied her NIDS, which uses high-fidelity unsupervised machine learning to detect both known and zero-day intrusion attacks in genuine time. Demonstrates the results that Griffin's complexity is approximately 40% lower than existing NIDS and its accuracy is up to 19% higher. Moreover, even in avoidance situations, Griffin's AUC increased by up to 9% and was compared to other solutions for good performance. Muthanna et al [22] advised an intelligent SDN-enabled hybrid framework in IoT environments that leverages Cuda Long Short Term Memory Gated Recurrent Unit (cuLSTMGRU) for efficient threat detection. At launch, this model achieved a protection accuracy of 99.23 with a short untrue positive rate. Besides validation, compared the results of the expected model with the current benchmark algorithm and two built models (i.e.,cuBLSTM and cuGRUDNN). Ravi et al[23] projected an end to a network assault model classification and network attack discovery employing the recurrent models based on deep learning. The findings of experimental research and the proffered method's performance on multiple benchmark network intrusion datasets show that it beats existing approaches and other widely utilized deep and machine learning models. is showing. Bour et al [24] studied a defence-in-depth mechanism to detect and

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

defend against three different types of flooding DDoS attacks mitigate. This representation compares well with similar schemes suggested in the literature concerning numerous metrics such as attack detection rate, detection accuracy, false positive rate, switch error rate, packet loss rate, response time, and CPU utilization will comparing the framework performs better. Bao et al [25] developed and integrated a powerful machine learning-based SDN for network intrusion detection systems (NIDS). Research outcomes express that the planned design occupies about 23% of DSP hardware resources and 16% of FFs, LUTs, and BRAMs. This research study contributes to the optimization algorithms with optimal controller placement using hybrid software-defined networks for safe multipath selection.

III. RESEARCH PROBLEM DEFINITION AND MOTIVATION

The control plane separated taken utilizing the information plane and enabling programmable management of the network, software-defined networking (SDN) has demonstrated significant benefits in many real-world aspects. Therefore, as networks grow, a single controller SDN imposes severe limitations on various functions. Multiple controllers are therefore essential for highly scalable networks. The CPP determines the ideal various controllers and the deployment location, which affects the execution of the network. Additionally, there are many limitations on the performance of an individual SDN controller. Among the most significant difficulties in SDN frameworks with several organizers is the CPP. These properties affect network functionality. The NP-hard problem is defined as the CPP. It indicates the number of controllers required to cost-effectively handle the controller traffic load and the position of the controllers in the SDN. Similarly, as the range and typologies of security risks in modern networks have expanded, the implementation of SDN, a network-based IDS, has expanded the possibilities for its application. The amount of data in networks is growing rapidly, and connected devices present their security risks.

Due to transmission efficiency, the SDN has been employed in numerous organizations. SDN uses machine learning techniques to improve the efficiency of resource scheduling. Existing models of SDN have limitations such as overfitting, local optimal trapping, and poor path selection efficiency. The SDN for real-time data processing can impact the resources and time required for reconfiguration. The network requires a dynamic resource planning scheme to support the SDN system, as network resource planning and information exchange routing strategies are also critical in emergencies. A NIDS is the process of detecting the presence of malicious or unwanted packets on a network. This process is approved by employing real-time traffic monitoring to look for any unusual network behaviour. These NIDS are important tools for detecting malicious activity and anomalous attacks to protect the network environment. Therefore, SDN does provide the ability to monitor network security issues and successfully detect them caused by the advent of customizable features. Currently, implemented the SDN based on machine learning (ML) to approach NIDS to defend the networks of computers and overcome security problems in the network.

IV. PROPOSED RESEARCH METHODOLOGY

SDN is a new architecture that separates network control and forwarding functions and allows direct programming of network control. Enables the data and control plane to be separated to simplify network management. This capability of SDN facilitates innovative applications and determines new network paradigms that NIDS can implement. Tactics Machine learning and deep learning (ML\DL) can be implemented in SDN controllers to improve security and monitoring networks. Therefore, SDN architecture for dynamic resource allocation and scheduling is a fresh paradigm in networks. Despite that, SDN's centralized architecture simplifies overall network management and meets the needs of today's data centres. Consequently, SDN architectures offer significant advantages, the risk of emerging attacks is a significant issue that may hinder widespread adoption of SDN. SDN organizers are the main elements and intruders in attractive targets.

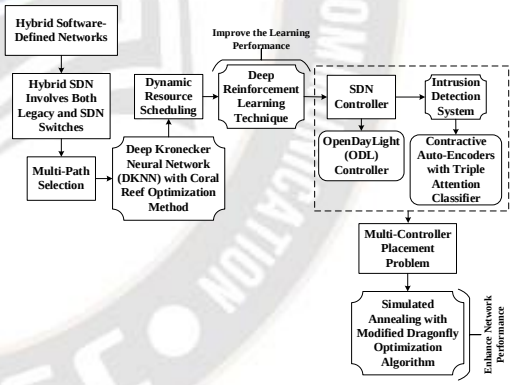

Figure 1. The proposed work's block diagram

Figure 1 research methodology represents the block diagram. The research origination accelerated in the flexible nature of SDN and conventional networks are compared in the security measures increase such as prevention and threat detection. However, SDN offers significant benefits, some security challenges need to be addressed before the updated paradigm is widely adopted. Anomaly-based detection techniques are theoretically good at identifying unknown attacks but have poor low detection rates and analytical capabilities.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

4.1 Hybrid Software-Defined Network

This study presents a cross-network situation where an SDN device replaces an external perimeter transport device in a service provider network. However, other internal transfer devices continue to function. The utilization suggested the conquest approach is based on the subnet of free IPs. All IP addresses do not know that within a subnet is used. This allows a pool of free IP addresses to be managed on the controller. However, if a packet desires to be forwarded, the controller assigns the packet one of the free IPs from the pool. The terminus address field in the packet header is then pre-modified and set by the controller. Furthermore, the controller installs mapping and remapping on the SDN router's ingress and egress in flow entries. Architecture has only ingress and egress routers, replaced by SDN routers, the rest remain legacy routers

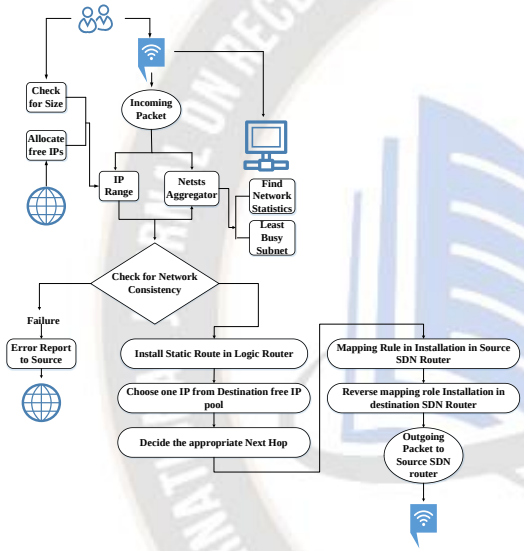

Figure 2. Architecture Diagram of Hybrid SDN

Figure 2 the suggested strategy is displayed in a flowchart. Definite assumptions are made for the intended routing method to work efficiently, which are defined as follows-

1. The network in each subnet was connected to one SDN switch in the legacy router.
2. The controller should be available in all details about the network.
3. SDN functionality is only enabled on ingress and egress routers.
4. A full mesh or subnets was formed by the topology.

There is a memory limit for current tables in SDN routers and routing tables in legacy routers. A high number of requests can cause some flow entries to overflow. Another consideration is that the system needs to keep track of the whole amount of policies. In agreement with these conditions, the system architecture can impose certain restrictions, which are defined as follows- therefore, make sure these and any policies added to the subnet are taken into account. Moreover, all subnets are frequently applied for this procedure and ultimately each subnet's count is added to make a total count. Equation 1 the specified criteria represent.

$$\sum_{i=1}^{N} PP^{i} = PP_{total} \quad \square\square\square\square$$

Where PPi denotes the amount of SDN switch policy paths, PPtotal indicates the total quantity of policy paths from each source to each destination in the system, and SDN switches are defined by the count of N that meets the criteria. The number of subnets mentioned in Assumption 1.

The size of an SDN router's flow table is determined by three factors. First, the subnet has already been defined by the flow entries for hosts. These entries are required for reverse mapping to convert the target IP to a host-specific IP. Second, add entries that are directly for the target subnets or obliquely associated with this SDN router. Third, check if there are any unused entries in the SDN router's IP mapping and mark them as unmapped entries. Finally, add all three parameters to get the final total number of flows. This number of flows should be less than the SDN router of total flow capacity. Equation (2) specifies the above criteria.

$$H_{src}^{n} + FE_{non-map}^{n} + \sum_{j=1}^{p_k} H_{dst}^{j} \leq FT_{max} \forall k \in S \quad \square\square\square\square$$

Where, H_{src}^{n} represents the number of IPs in subnet src nth SDN switch that are directly connected and $FE_{non-map}^{n}$ is the amount of flow entries that do not correspond to the nth SDN switch where H_{dst}^{j} is the numeral of IPs for the policy of j th in the destination subnet, FT_{max} is the magnitude of the SDN switch's flow table, and p_k is the quantity of multiple policies when the SDN switch acts as a source. where S represents a set of policies for an SDN switch that meets the prerequisites.

$$RP_{def}^{i} + RP_{clashes}^{i} + PP^{j} \leq RT_{max} \forall j < L \quad \square\square\square\square$$

where, RP_{def}^{i} represents the number of routing paths to perform legacy functions for legacy routers and $RP_{clashes}^{i}$ constitute the issue of routing paths issued to legacy routers due to IP conflicts. The router PP^{j} can handle it. Where j is the number of policy paths that complete the j-th legacy router, RT_{max} is the size of the legacy router's routing table, and L is the set of legacy routers\/forwarding devices for which manual policy must be defined. The various policy options an administrator can implement in a system depend on the size of the SDN router's flow table.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

4.2 Multi-Path Selection

Due to the planned cross-layer network resources and the complex planning process of network status monitoring such as RTT and bandwidth, equidistant multipath networks are very difficult to achieve. Kronecker Neural Networks (ANNs) provided the neural networks for adaptive activation capabilities in a general framework. Additionally, in this object, we reconstruct a set of completely related layers of the CNN architecture from scratch using meta-heuristics based on coral reef optimization. Multipath selection for this combination method is used to improve efficiency in SDN.

4.2.1 Deep Kronecker Neural Network

The Kronecker product structure serves as the foundation for DKNN, which implicitly enforces a latent individually smooth property of coefficients. The DKN also facilitates model interpretation by helping us identify the node sections that are most important to the result. Specifically, a fully convolutional network (FCN) and CNN are two networks that DKN mimics. This approach enables the necessary model interpretability in addition to the maximum dimension reduction.

Provide the fixed positive integer defined by K. Given the parameters of the FNN $\Theta_{FF} = \{W^l, b^l\}_{l=1}^{D}$, the n-th block weight matrix and block bias vector are indicated by,

$$1_{K\times K}\otimes W^l = \begin{bmatrix} W^l & \cdots & W^l \\ \vdots & \ddots & \vdots \\ W^l & \cdots & W^l \end{bmatrix} \in \mathbb{R}^{N_l K \times N_{l-1} K} \quad 1_{K\times 1}\otimes b^l = \begin{bmatrix} b^l \\ \vdots \\ b^l \end{bmatrix} \in \mathbb{R}^{N_l K} \qquad (4)$$

Where $\otimes$ the product was provided by Kronecker. Let's a block activation function were defined to apply per block denotes in $\vec{\varphi}$. Therefore, $z_j \in \mathbb{R}^n$ for $1 \le j \le K$, let $z = [z_1, \cdots, z_k]^T \in \mathbb{R}^{nK}$. Then build a neural network from the block weight matrix and block bias vectors as follows: Take $z^0 = z$ as input and $z^1 = (1_{K\times K}\otimes W^l)(1_{K\times 1}\otimes z^0) + 1_{K\times 1} \otimes b^1$. For $2 \le l < D$,

$$z^l = (1_{K\times K}\otimes W^l)\vec{\varphi}(z^{l-1}) + 1_{K\times 1} \otimes b^1 \qquad (5)$$

However, $z^D = (1_{1\times K}\otimes W^D)\vec{\varphi}(z^{D-1}) + b^D$. In this case $z^D$is a D-layer FNN with a large number of KN_1 neurons in the lth layer, but the amount of network parameters remains the same as u^{FF}.

Kronecker networks have K-times as many neurons in any hidden layer as in feed-forward (FF) networks. Furthermore, the total amount of parameters varies by 2K (D-1) owing to the Kronecker product. Simultaneously, Kronecker networks may be watched as an original kind of neural network that generalizes the existing class of feed-forward neural networks, especially to take advantage of functions of adaptive activation.

Any grouping of initiation functions may be utilized in ANNs, a broad framework for adaptive activation functions. Therefore, there is no obvious way to choose these activation functions. In this context, propose a rough activation function and call the corresponding ANN a rough net (a neural network with a rough activation function). In the wild network, we choose {\u03c6_1 } as the standard activation functions such as ReLU, tanh, ELU, Sine, Swish, Softplus, and the remaining $\{\varphi_k\}_{k=2}^{K}$ activation functions are selected to

$$\varphi_k(x) = n\,sin\big((k-1)nx\big) \quad n\,cos\big((k-1)nx\big) \quad \forall 2 \le k \le K \qquad (6)$$

Where $n \ge 1$ serves as a scaling factor and is a constant positive number. The word "random" means wildly fluctuating/irregular/noisy and is used to describe the activation function $\{\varphi_k\}_{k=2}^{K}$. The purpose of selecting such a variation term is to introduce a limited but highly non-monotonic and noisy effect from the outcome of any layer of the network to remove the saturated regions.

An important role of the scaling factor n plays in the convergence of the network training process. Indicating the value there is no rule of digit for scaling factor, it depends on the specific problem. Numerical experiments show that for the regression problems such as function approximation and solving partial differential equations, specifying a value of n \u2265 1 can speed up convergence, but the optimization algorithm becomes less sensitive as the value of n increases. may become. The trainable parameters are defined in scaling factors and are prepared such that the Rowdy-Net initial activation matches the corresponding default activation function. By contrasting Rowdy-Net's performance with that of fixed (default) and layer-wise locally adaptive (L-LAAF) for a variety of regression and classification test scenarios, we show the effectiveness of Rowdy-Net in this section. For simplicity, describe fixed (f) and trainable (t) parameters for all three forms of instigation functions with their respective initializations.

Fixed AF: $\alpha_1^l = 1(f); \quad \alpha_k^l = 0, \forall k \ge 2(f); \quad \omega_1^l = 1(f); \quad \omega_k^l = 0, \forall k \ge 2(f),$

L-LAAF: $\alpha_1^l = 1(f); \quad \alpha_k^l = 0, \forall k \ge 2(f); \quad \omega_1^l = 1(t); \quad \omega_k^l = 0, \forall k \ge 2(f),$

Rowdy AF: $\alpha_1^l = 1(f); \quad \alpha_k^l = 0, \forall k \ge 2(t); \quad \omega_1^l = 1(t); \quad \omega_k^l = 0, \forall k \ge 2(t),$

Note this initialization makes the initial activation functions of both L-LAAF and Rowdy-Net for each hidden layer identical to the fixed activation functions. Further initialization is performed by multiplying the scaling factor n in (6) by any trainable parameter, say, ω_1^l, the initialization is done such that $n\omega_1^l = 1, \forall n$.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

4.2.2 Coral Reefs Optimization Algorithm

Coral reef optimization-primarily based meta-heuristics can perform a deeper seek of the quest space, hence compared to assisting gradient descent-based methods to locate better solution places. SCRO metaheuristics provide a powerful tool for evolving CNN models without the need for hyperparameter modifications. First, the study expands on statistics-driven CROs. The method consists of a final hybridization process that optimizes the best solution found by the last-generation metaheuristic using a backpropagation algorithm and performs a final fine-tuning training. The mutation operator on the other hand has also been extended to perform stratified operations implementing both parametric and structural mutation.

The SCRO defined the evolutionary process as consisting of four operators: asexual reproduction, sexual reproduction, colonization, and predation. Evolutionary operators can apply various using a series of confidence intervals corresponding to population fitness values. A Gaussian distribution follows the basic assumption that these fitness values, yield a population with variance calculated as

$$S_{f_j}^2 = \frac{\sum_{i=1}^{N_j}(f_{ij}-\bar{f}_j)^2}{N_j-1}, j = 1, \cdots, M$$

f_ij is the jth generation fitness value for the ith individual, Nj is the number of individuals in this generation, while $\bar{f} = \sum_{i=1}^{N_j} f_{ij}/N_j$ is the average fitness, M is the generation is the total number of This final step aims to improve the solution through final fine-tuning.

Initialisation: First, an empty riff with $n \times n$ positions is created. The worst solutions (those with larger loss values) are eliminated from the n2 random solutions generated by the initialization function.

$$f_{i1} \notin \left[0, \bar{f}_1 + S_{f_1}\right]$$

An initial set of solutions is randomly generated to perform a comprehensive search over the entire solution space.

Asexual Reproduction: Operators of asexual reproduction carry out the process of fragmentation followed by budding. First, fragmentation computes individuals whose fitness confirms the following formula.

$$f_{ij} \in \left[0, \bar{f}_j - S_{f_j}\right]$$

Sexual Reproduction: Implemented the original SCRO contained two types of sexual reproduction: external and internal. The crossover operation is similar to performing the former replication in evolutionary algorithms, but this version of the SCRO algorithm does not perform this operation, since only mutations are considered. In the inner replica, random mutations are carried out to all people recognized via the fitness function.

$$f_{ij} \in \left[0, \bar{f}_j + S_{f_j}\right]$$

If the number of individuals meeting the previous conditions does not happen at the lowest threshold (miniprep), random corals from the remaining population are included in the set. Each iteration in this ensures that reproduces the minimum number of individuals.

Settlement: Populations obtained by the two reproductive methods mentioned above already established on the reef must compete with corals. A random reef location is then selected for each separate in the new population. If the new solution is a better fit or the position is open, the person will be assigned to this position. This process can be recurrent up to 5 times earlier the new solution is discarded. The main fitness function is categorical cross-entropy, but significantly improved accuracy (denoted by Θ) also displaces previously specified individuals with solutions. This helps evade local minima and improve search diversity. Therefore, a new individual with fitness fi satisfies one of the following conditions replace the reef with fitness fi in an individual: Each individual's fitness is determined by evaluating all training examples using an individually coded model. Similarly, during the evolutionary search is unchanged with the convolutional block, evaluate to need the presentation of the last set of entirely associated layers using the predictions made in the last flattened layer as input. is. This loss function is calculated as:

$$cate\_crossentropy = -\frac{1}{N}\sum_{i=1}^{N} log p_{model}\left[y_i \epsilon C_{y_i}\right]$$

Wherever, according to the evaluated model $p_{model}\left[y_i \epsilon C_{y_i}\right]$ is the possibility that Pattern I belong to the class C_{y_i}. The best path is determined from multiple sources in the SDN network. The parametric mutations perform the best-fit individuals selected to affect weights and biases. The parametric mutations perform the great fit individuals are decided on to affect weights and biases. The closing people undergo structural modifications along with hyperparameter mutations (simplest activation functions are mutated) and connectivity mutations (connections between nodes are activated or deactivated).

4.3 Dynamic Resource Scheduling

The fundamental function has resource planning and allocation of slicing the network in 5G networks. Dynamically adjust resource allocation from the learning-based dynamic scheduling schemes that utilize DRL.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

4.3.1 Deep Reinforcement Learning (DRL)

An extension of reinforcement gaining knowledge is the DRL using deep neural networks to represent the value capabilities. The period "agent" getting the reinforcement is used for the role of a selection-maker to interact with the surroundings. Markov selection process is applied to demonstrate this sequential decision problem. it may be defined by way of a tuple, which includes a country set S, an action set A, a country transition characteristic $P(s_{t+1}|s_t, a_t)$, a praise function $R(s_t, a_t)$, the finite making plans horizon T and a discount issue γ for calculating the anticipated discounted go back $\mathbb{E}[\sum_{i=0}^{T-1} \gamma^t r_t]$.

A program can manage subsequent slots by intervening appropriately in any state, such as:

$$x_\tau = \{\epsilon_\tau, \psi_\tau, q_\tau, l_\tau\}$$

As ϵ_τ collective system cost at time τ, and $\psi_\tau = \{ck_{k,1}^\tau, ck_{k,2}^\tau, \dots, ck_{k,N}^\tau\}$ vector of length N indicates the computing power of a compute node as follows:

$$ck_k^\tau = K - \sum_{n=1}^{N} \alpha_w^a k_n^o$$

Where, ck_k^τ is the remaining capacity of each compute node k with slot τ. Somewhere, $q_\tau = \{q_{k,1}^\tau, q_{k,2}^\tau, \dots q_{k,N}^\tau\}$. Furthermore, $l_\tau = \{d_{a,k,1}^\tau, d_{a,k,2}^\tau, \dots, d_{a,k,N}^\tau\}$ define $d_{a,k}^\tau = [l_a - l_k]$ measuring distance among users' devices and nodes.

Reward Function

Equivalent to the coverage gradient set of rules, the simplest manner to update the intrinsic reward characteristic is to consume$\nabla_{\theta^{in}} j^{fitness}(\mu)$, the gradient of policy overall performance with appreciation to the fitness feature $j^{fitness}(\mu)$ to calculate. Considering that a change in the inherent recompense function parameter θ^{in} it affects the guideline $\mu(s|\theta^\mu)$ and may affect $j^{fitness}(\mu)$, the gradient $\nabla_{\theta^{in}} j^{fitness}(\mu)$ may be unglued into two gradients.

The agent contains two modules. One is a policy module based on the DDPG algorithm and the other is a unique reward module built according to our approach.

Through our system, the rate state awards for specific measures as follows:

$$\max_{v_\tau} E[\sum_{\tau=0}^{r} r_\tau(s_\tau, v_\tau)]$$

It represents also tries to obtain a new maximum discounted reward R_τ in the future.

$$R_\tau = r_\tau + \gamma R_\tau + 1$$

Whereas, γ is a discounted amount $0 \leq \gamma \leq 1$.

The essential reward engine consists of a fitness network $Q^{fitness}(s_t, a_t|\theta^{fitness})$ which estimates the value of an action centred on fitness, and an intrinsic reward network $R_{\theta^{in}}^{in}(s_t, a_t)$ issue rewards state by state. For strategy modules, the critical network$Q(s_t, a_t|\theta^Q)$ estimates the value of actions based on total rewards, and the actor-network $\mu(s_t|\theta^\mu)$ maps states to deterministic actions. Interaction with the environment, the agent achieves activities a_t from the actor-network according to the state s_t, and the environment returns to fitness $r_t^{fitness}$ and external rewards r_t^{ex} to the agent. Fitness $r_t^{fitness}$ works within a fitness network. Specific reward network parameters can be updated as follows:

$$\nabla_{\theta^{in}} L_{intinsic} = \nabla_{\theta^{in}} j^{fitness}(\mu) = \nabla_{\theta^\mu} j^{fitness}(\mu) \nabla_{\theta^{in}} \theta^\mu$$

The precise praise community besides the goal network, which is a duplicate of another community, is used for goal action value estimation and goal policy calculation. The target network of the parameters is updated by soft parameter replacement, which improves the stability of learning. Noise is additional to the action to facilitate exploration using a Gaussian process. After interacting with the environment, agents are trained on specific procedures.

4.4 SDN Controller and IDS

The SDN organizer is accountable for centralized communication with all programmable elements of the network, providing a holistic view of the network. ODL is an exposed foundation modular platform that adapts and automates networks of any size. The key layers of the ODL architecture are the Controller Platform Layer and the Service Abstraction Layer (SAL). These are Southbound (SB) plugins that communicate with network devices and Northbound (NB) plugins that communicate with valid applications using the controller. The essential function can provide a controller platform layer containing modules. Modules in this layer, therefore, define the behaviour of the network and are selected by the controller. DoS attack mitigation application is at this layer. When the controller generates proactive rules are populates the flow table with rules before a packet arrives. The functionality of the switch is respected for these proactive rules prevent flooding. However, it is conceivable to install reactive rules that allow data forwarding without flooding and send data only to the appropriate egress port.

The controller will become the community bottleneck and is susceptible to intruders. The IDS advanced inside the network detects threats via monitoring the transmitted packets. IDS detects unusual and malicious activity by internal and external

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

intruders. Consequently, this study offered a contractile auto-encoder with a triple attention base for the intrusion detection process is classified.

4.4.1 Contractive Auto-Encoders

The regularization period of the equation yielding a goal feature become obtained by the Contractive car-encoder (CAE)

$$J_{CAE}(\theta) = \sum_{x \in D_n} \left(L\left(x, g\left(f(x)\right)\right) + \lambda \left\| J_f(x) \right\|_F^2 \right)$$

Relationship with Weight Decay

It is straightforward to look that the squared Frobenius standard of the Jacobian corresponds to an L2 weight deterioration in the event of a linear encoding. In this unique case $J_{CAE}$and J_{AE+wd} are same. observe that in the linear case, preserving weights small is the most effective way to have a contraction. However, the hidden units of their saturated regime with a sigmoid non-linearity, contraction and robustness also can be completed by way of equitation.

Relationship with Sparse Auto-Encoders

Variants of autoencoders that stimulate sparse representations aim for the majority of the components of the representation to approach zero in each instance. To bring these features close to zero, they should be computed with the left-saturation part of the sigmoidal nonlinearity. It is nearly flat and has a very small first derivative. This gives us a small entry corresponding to the Jacobian $j_f(x)$. Therefore, a sparse autoencoder that emits many near-zero features is likely to accommodate strong shrinkage mappings, even if shrinkage and robustness are not explicitly the learning criteria that were recommended.

Relationship with Denoising Auto-Encoders

Vincent et al, have found that robustness to input interference is also one of the motivations behind noise-cancelling autoencoders. (2010). However, CAE and DAE fluctuate within the following approaches: CAE explicitly promotes robustness of the representation f(x), while DAE promotes robustness of the reconstruction (g ◦ f) (x) (that is simplest partially and indirectly sturdy illustration, the z-invariant requirement is shared between the two elements of the autoencoder). This property makes CAE a better preference than DAE for gaining knowledge of beneficial feature extractors.

4.4.2 Triple Attention Classifier

A triple-attention-based model is contemplated for detecting anomalies in heterogeneous graphs. The edge-level interest, view-degree interest, and node-type attention are indicated by the Triple interest. A given heterogeneous diagram has two types of nodes, C and S, where a type C node has three views and a type S node has two views. Each association of opinions is entered into the HGT separately and the overall embedding is generated through view-level attention-based aggregation. Anomalies may be perceived by ranking the anomalies after decoding by structure, attribute and node type decoders.

Multi-View Heterogeneous Graph Encoder

Heterogeneous graphs distinguish between vertices and edges. Of course, if aggregating the attributes of adjacent nodes are to get the node embeddings, the node embeddings can aggregate by allowing the network to be aware of the edge diversity and absorb the reputation of each edge. The embedding of nodes can be able to get it better based on importance. Networks like GCN failed to detect edge heterogeneity. In HGT, the network uses attention mechanisms to automatically learn the meaning of each edge and provide tools. Therefore, he uses HGT as an encoder to encode the given heterogeneous graph. To get the embedding of goal node t into supply node s of (s, e, t), heterogeneous multi-head attention $Attention_{edge}(s,e,t)$ denoted utilizing $Att_{edge}(s,e,t)$ and message $Message_{edge}(s,e,t)$ denoted via$Mes_{edge}(s,e,t)$.

$$\widetilde{H}^{(l)}[t] = \sum_{\forall s \in N(t)} Att_{edge}(s,e,t) \times Mes\,(s,e,t)$$

$$Att_{edge}(s,e,t) = \underset{\forall s \epsilon N(t)}{\text{softmax}} \left(\underset{i \in [1,h]}{\text{Concat}}\, Att - head^{i}_{(s,e,t)} \right)$$

$$Mes_{edge}(s,e,t) = \underset{i \in [1,h]}{\text{Concat}}\, Mes - head^{i}_{(s,e,t)}$$

Where the $i-th$ attention head is calculated by

$$Att - head^{i}_{(s,e,t)} = k^{i}(s) W^{Att}_{\phi(e)} Q^{i}(t) \frac{\mu < \tau(s), \phi(e), \tau(t) >}{\sqrt{d}}$$

$$k^{i}(s) = K - Linear^{i}_{\tau(s)}\left(H^{(l-1)}[s]\right)$$

Linear projection K-Linear:$\mathbb{R}^{d} \rightarrow \mathbb{R}^{\frac{d}{h}}$ projects τ (s)- type node s into Key vector $K^{i}(s)$ where h is the number of attention heads, d is the overall vector dimension and $\frac{d}{h}$

$$Q^{i}(t) = Q - Linear^{i}_{\tau(t)}(H^{(l-1)}[t]$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

Similarly $Q - Linear^{i}_{\tau(t)}$ is a linear projection projecting the target node t into the $i - th$ Query vector $Q^{i}(t)$. $W^{Att}_{\phi(e)} \epsilon R^{\frac{d}{h}\times\frac{d}{h}}$ is an edge-based weight matrix for edge type $\phi(e)$

$$Mes - head^{i}_{(s,e,t)} = M - Linear^{i}_{\tau(t)}(H^{(l-1)}[s])\, W^{Mes}_{\phi(e)}$$

refers to the $i - th$ message head. $M - linear: \mathbb{R}^{d} \rightarrow \mathbb{R}^{\frac{d}{h}}$ is a linear projection. $W^{Mse}_{\phi(e)} \epsilon R^{\frac{d}{h}\times\frac{d}{h}}$ is a matrix incorporating edge dependency. $\mu \epsilon \mathbb{R}^{|A|\times|R|\times|A|}$ is an adaptive scale tensor to attention. Then the embedding of target node t can be obtained by

$$H^{(l)}[t] = \sigma\left(Linear_{\tau(t)}\tilde{H}^{(l)}[t]\right) + H^{(l-1)}[t] \quad \square\square\square\square\square\square\square\square\square\square\square\square\square\square$$

If σ is the real function, then $Linear_{\tau(t)}$ is a linear projection that projects the vector of target nodes t onto the attribute dimension of that particular type. AHEAD enters a view for each node of type HGT that is encoded separately. This means that there are a total of K Q HGT models for target node t. where $K = \prod_{\forall s \epsilon N(t)\cup\{t\}} K_{T(s)}$ Of course, the HGT has K outputs, each an embedding of the view of the target node t. Map each view's attributes to the same space before passing them to the HGT. For Edge (s,e,t), the process of the i-th opinion of t and the j-th view of s feeding into the HGT can be described as follows.

$$\square\square\square H^{(0)}(t_i) = H - Linear^{i}_{\tau(t)}\left(X^{i}_{\tau(t)}\right) \quad \square\square\square\square\square\square\square\square\square\square\square\square\square\square$$

$$\square\square\square H^{(0)}\left(S_j\right) = H - Linear^{i}_{\tau(t)}\left(X^{j}_{\tau(s)}\right) \quad \square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square$$

Where, $H - Linear^{i}_{\tau(v)}: \mathbb{R}^{D^{i}_{\tau(v)}}$ indexed by the form of nodes v's. This means that each node type has a unique linear projection to compress the size of every view's attributes. T_i denotes the i-th view of t and S_j denotes the j-th view of s.

Structure and Attribute Decoder

A structural decoder aims to compute a reconstructed adjacency matrix based on the output of the encoder. As mentioned in the introduction, each edge type has its adjacency matrix. Certainly, each edge type has its own reconstructed adjacency matrix. The attribute decoder aims to reconstruct the attribute information for each view. Attribute information for nodes of $\tau\ (v)$ -type through an associated layer is reconstructed.

Node-Type Decoder

Therefore, node heterogeneity is the main distinguishing feature between heterogeneous and homogeneous diagrams, the node type decoder makes the model more suitable for heterogeneous diagrams.

Anomaly Detection

The convergence of L allows AHEAD to gradually grasp potential anomaly information. After the training process, we use the anomaly score to indicate how anomalous node v is. This is expressed as:

$$s(v) = \lambda_1 \left|\left|A_{\phi(e)}[v] - \tilde{A}_{\phi(e)}[v]\right|\right|_2 + \lambda_2 \left|\left|X_{\tau(v)}[v] - \tilde{X}_{\tau(v)}[v]\right|\right|_2 + (1 - \lambda_1 - \lambda_2)\left|\left|T[v] - \tilde{T}[v]\right|\right|_2 \quad \square\square\square\square\square\square\square\square\square\square$$

Where λ_1 and λ_2 are hyperparameters to balance the weights between the three terms. It can rank anomalies based on anomaly scores. Or, in terms of probabilities, the following expression is the prospect that node v is anomalous.

$$p(v) = \frac{s(v)}{\max\limits_{\forall v \epsilon V}(v)} \quad \square\square\square\square\square\square\square\square\square\square\square\square$$

To capture network structure heterogeneity, node properties, and network structure information, in this study, attention to three levels, edge level, Level node type, and namely attribute level the specially designed encoders designed in a single node. The decoder obtains the anomaly groove for any node using the reconstruction terms for structure, attributes, and node types.

4.5 Multi-CPP

SDN's centralized controller-based totally architecture simplifies control aircraft design. but, an individual controller has restrained potential to deal with network events, and may effortlessly become a bottleneck in phrases of reliability and scalability. One solution to this difficulty is to custom multiple controllers in the control plane. The problem of controller placement in large networks remains a difficult challenge due to its high complexity and difficult performance trade-offs. Intended a new approach called simulated His annealing using an MDOA.

4.5.1 Simulated Annealing with MDOA

Suggested an adapted dragonfly optimization and established before solving the design model for the DER mix. Obtained the results were competitive, indicating that the planned improvements to the standard DA increased the solution-finding potential. The contemplated approach uses MDA to optimize the gain parameters. In SDN surroundings, the greatest controller placement reduces latency in communique between controllers and switches. consequently, adopted a simulated glow algorithm. The intention is to transition the device from a random preliminary state to a nation with minimum energy. SA may be utilized for very hard computational optimization issues wherein precise algorithms fail.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
Article Received: 01 May 2023 Revised: 28 June 2023 Accepted: 27 July 2023

Because SA is based on probabilistic models, it can solve nonlinear global problems. The objective function (F(x)) is provided below in maximized and minimized form. In the maximized form, defining an objective function is possible as

$$F(\overrightarrow{x_{OPT}}) = \max_{\vec{x}_i \epsilon \vec{Y}} F(\vec{x})$$

Equation (34), represents the impartial function as the optimal vector that solves the problem of placing the controller at the right place in the network for the maximal function of $x_{i,j}$ represented by $F(\overrightarrow{x_{OPT}})$. In underestimated form, the objective function may be expressed as

$$F(\overrightarrow{x_{OPT}}) = \min_{\vec{x}_i \epsilon \vec{Y}} F(\vec{x})$$

The objective function used in equation (35) is the optimal vector that solves the problem of properly positioning the controller for the smallest function of $x_{i,j}$ represented by $F(\overrightarrow{x_{OPT}})$ is represented.

For the phases of SA, a new optimal point is selected. If $\Delta F < 0$, $(x'_{i,j})_{New}$ and $(x_{i,j})$ Current is the new optimal points and $F(x'_{i,j})_{New} = F(x_{i,j})_{Current}$ is used for the next process. Otherwise, the possibility of the density function $(Pr(\Delta F))$ is computed, which can be formulated as

$$Pr(\Delta F) = exp^{(-\Delta F / T_{L(itr)})}$$

Calculating the opportunity of the density characteristic Pr(ΔF) is used for Equation (36), that's a fashionable equation for all parameters. Then, a random range (Θ) is generated to attain the optimum role. If Θ is less than $Pr(\Delta F)$, then $(x'_{i,j})_{New}$ Now is the greatest point of (x_(i,j))_Current and the next phase proceeds. The system is expressed as

$$Pr = \begin{cases} (x'_{i,j})_{New}, & if\ \Theta \epsilon [0,1] < Pr(\Delta F) \\ \Delta F = F(x'_{i,j})_{New} - F(x_{i,j})_{Current}, & Else \end{cases}$$

The probability in equation (37) consists of two parts: the controller's position in terms of distance and time, and the density function(ΔF).

In this research, Based on the SA discovery process described, multiple controllers are optimally placed in his SDN environment. This process is repeated until the optimal slider position is reached.

4.5.2 Modified Dragonfly Optimization Algorithm

The metaheuristic algorithm is a dragonfly algorithm founded on swarm intelligence based on the fixed and dynamic nature of dragonfly behaviour.

The suggested optimization starts by initializing a population of "n" dragonflies (i.e. controllers "$C = \varphi_1, \varphi_2, \ldots, \varphi_n$") in the exploration space (i.e. SDN).

$$C = \varphi_1, \varphi_2, \ldots, \varphi_n$$

After the initialization process, the fitness of each dragonfly in the current swarm population is calculated. Fitness is calculated based on a multi-objective function. First, estimate the propagation delay between the switch and controller. Data transmission time from the controller to the switch in a network is known as propagation delay. Mathematically, it is estimated as follows.

$$\alpha_{lat} = time[TD]$$

where, α_{lat}suggests latency and TD show data switch from the controller to replace. the load balancing capability (α_{load}) between the transfer and the controller is measured primarily based on the load aspect calculation.

Fitness measurements based on four herd behaviours of dragonflies are estimated within the search space. The four behaviours are segregation, alignment, cohesion, and attraction to food sources that help populations find optimal solutions globally. First, run the separation process to find out the current bubble position and the neighbouring bubble positions.

$$\beta_1 = -\sum_{j=1}^{n} (x_{it} - x_{jt})$$

Where $\beta 1$ indicates a separation of the dragonflies, x_{it} denotes a current position of a dragonfly, x_{jt} symbolizes the position of the neighbouring dragonflies, 'n' is some neighbouring dragonflies in search of Space. The second process is alignment. This shows the speed of the dragonfly and the speed of neighbouring dragonflies.

$$\beta_2 = \frac{1}{n} \sum_{j=1}^{n} \mathrm{T}_j(t)$$

Where, β_2 represents the direction, $\mathrm{T}_j(t)$ represents the velocity of the neighbouring dragonfly, and 'n' indicates the neighbouring dragonfly. Third, the agglomeration processes determine the tendency of the dragonflies to steer toward nearby centres of mass.

$$\beta_3 = \frac{1}{n} \sum_{j=1}^{n} [x_{jt} - x_{it}]$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

From (42), indicates that β_3 represents the dragonfly aggregation process, x_{jt} indicates the situation of the neighbouring dragonfly, x_{it} is the current dragonfly position, and n is the number of neighbours. Finally, we estimate the attraction process to the food source based on the current position of the dragonfly and the site of the food source.

$$\beta_4 = |x_f - x_{it}| \qquad \square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square$$

From (43), perceive that β_4 represents the attraction to the food source, x_f represents the spot of the food source, and x_{it} represents the current position of the dragonfly.

The current registration mark position is updated with its neighbours,

$$\chi_{i(t+1)} = \chi_{i(t)} + \nabla\chi_{i(t+1)} \qquad \square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square$$

From (44), $\chi_{i(t+1)}$ represents the updated position of the registration mark, $\chi_{i(t)}$ is the current position of the registration mark, and, $\nabla\chi_{i(t+1)}$is the step vector used to indicate the direction of movement of the registration mark? indicates A dragonfly to identify a dragonfly.

Based on updated results, the world's best solution is determined. Based on the analysis results, the optimal quantity of controllers is selected and placed in the network to improve overall network performance. Furthermore, it's far pretty ideal to lay out a green community to enhance network throughput and transport rate and decrease latency.

V. EXPERIMENTATION AND RESULTS DISCUSSION

Evaluated the simulation results based on various parameters such as packet delivery rate, packet loss rate, throughput and average delay. Simulation results of existing technologies are discussed in terms of numerous performance indicators such as average delay, throughput, packet loss rate and packet delivery rate. This section describes the experimental analyzes performed to evaluate the production of the submitted method. The unsurpassable regulator is designated from among many SDN controllers constructed on many package characteristics. The unsurpassable controller is cloned to 7 controllers and a multi-controller placement is performed. Position and distance are taken into account for efficient multi-controller placement and mitigating controller placement issues. Optimal selection and placement of multiple controllers improve communication efficiency among switches and controllers.

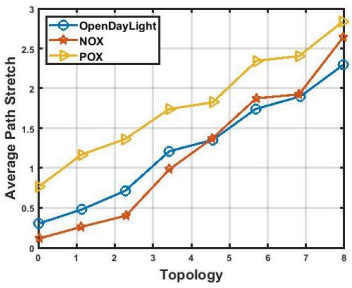

Figure 3. Average Network Path Stretch for Varying Sizes of Topology

Figure 3 shows a graphic representation of the typical network path coverage for different controller platform architectures. Network path length on average is minimal when using the ODL controller indicating the results. Therefore, this approach also works well for NOX controllers where the minimum path distance is calculated, but ODL outperforms his two other controllers for large networks.

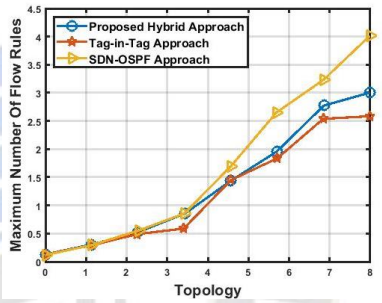

Figure 4. Average Flow Table Entry Consumption for Varying Sizes of Topology

Figure 4 graphically shows the consumption of flow entries by different topologies on different controller platforms. Figure 5 graphically illustrates flow entry usage by different topologies. From this observation, it is clear that the planned HSDN policy-based routing approach saves flow table space in the range of 87.512% to 81.052%. The largest savings occurs for the ODL controller, consuming an average of 12488.75 flow entries in different topology spaces. A small variation of 0.029% in flow entry consumption for NOX and ODL controllers was observed

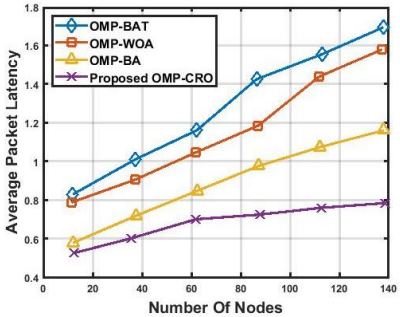

Figure 5. Average Packet Latency Exploration Based On the Number of Nodes

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

Figure 5 shows the standard packet delay for different network sizes of the suggested system. With 100 nodes, the typical packet delays for BAT, WOA, BA, and CRO are 0.65, 0.6, 0.46, and 0.41. The same technique yielded packet delays on average of 0.9, 0.8, 0.66, and 0.53 when the node reached the 120th level. However, existing technologies BAT and WOA have longer packet delays. With 150 H. nodes, the BA reaches 0.9 and the typical packet delay CRO reaches 0.6, which is roughly between 1.2 and 1.3. The intended OMP-CRO is speedy. Complicated as compared to all other processes. A node chooses the first-rate route with fewer hops and less node sharing. The consequence approaches in shorter delays and better utilization of bandwidth.

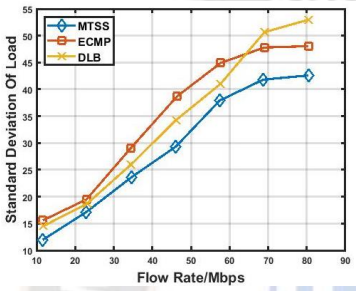

Figure 6. Comparison Performance of Load Balancing

Figure 6 shows the network load balancing performance. It might be understood from the figure that the standard nonconformity of MTSS is the smallest, and MTSS is always below the ECMP and DLB load under different flow rates. This shows that the flow distribution is relatively uniform and the network load is relatively balanced.

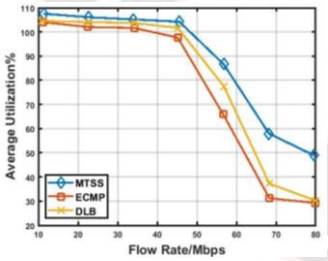

Figure 7. Comparison Analysis with Average Utilization of Network

Figure 7 demonstrates the regular bandwidth usage at various flow rates. Average utilization decreases as flow increases for all three algorithms. When the flow is low, the network has little data and the connection bandwidth is sufficient, so the load of the three algorithms is relatively high. While the current rate is high, the utilization of the three algorithms all decline. However, the overall utilization rate of MTSS is higher than that of the other two algorithms. ECMP and DLB are more likely to cause network congestion, so the average bandwidth utilization is lower.

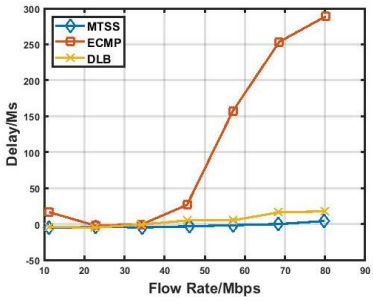

Figure 8. Transmission Delay Comparison Analysis

Figure 8 indicates the three algorithms behave similarly for transmission rates below 30 Mbps, which shows that the lower the transmission rate is, the lighter the network load is, and there is no congestion. However, with the increase in the rate, the delay of ECMP increases gradually, which indicates that the congestion is serious and the network performance is declining. Because ECMP itself determines it cannot distinguish which link is with a light load, and which link load is heavy, the data cannot be transferred to the lower load link. The result shows the count of packets waiting in the queue increases, greatly increasing latency. However, in the case of different traffic rates, MTSS is more stable in delay, because the algorithm can get the load of each link in the whole network topology, and then switch the packet to the optimal one.

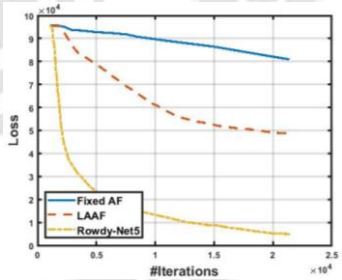

Figure 9. Damage Function for the Rowdy and Fixed Activation Functions

Figure 9 presented the loss capabilities for the constant activation characteristic and the layer-by-way-of-layer neighbourhood version (L-LAAF) difficult activation feature. In all instances, the take look used a gaining knowledge of fee of 9e-5, the activation feature become a hyperbolic tangent, the count of residual points changed into 10,000, and more than one side information points were four hundred. The FNN consists of three hidden layers containing 60 neurons in every layer. For this problem, Rowdy-Net5 converges faster neither Fixed nor L-LAAF converge after 20,000 iterations.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

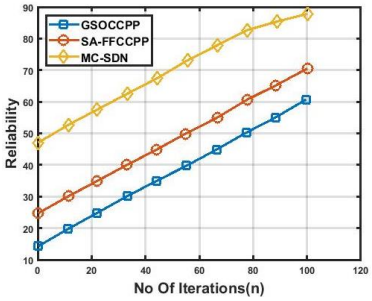

Figure 10. Reliability Analysis with Existing Techniques

Figure 10 suggests an assessment of the planned MC-SDN process and some previous research on generation count reliability. expanded reliability in increasing the number of iterations. Controller placement became done with unwell-adjusted optimisation techniques in existing methods and inefficient inter-controller verbal exchange in the SDN community, main to the unreliability of parent nine. From the determination, it's far clear that the offered MC-SDN technique achieves high reliability in comparison with other previous research. The suggested MC-SDN approach accomplishes a dependability of roughly 0.95 for 100 iterations. This is 0.07 higher than the SA-FFCCPP method and 0.11 higher than the same number of iterations at the GSOCCPP method.

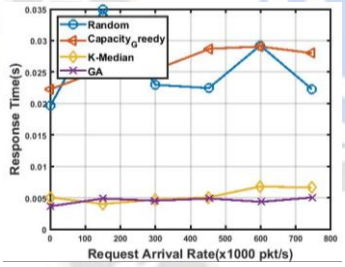

Figure 11. Average RTT Analysis with Response Time

Figure 11 shows the average RTT analysis and response times. In the considered study, the optimal controller is designated using an FA algorithm that considers a large number of the controller features that reduce the RTT compared to prior art studies. Comparative data demonstrates that the suggested MC-SDN method, when compared to other approaches already in utilise, achieves low RTT. It achieves the suggested MC-SDN scheme with an average RTT of about 11ms. This is 4ms shortened compared to the SA-FFCCPP method and 8ms shorter than the GSOCCPP method.

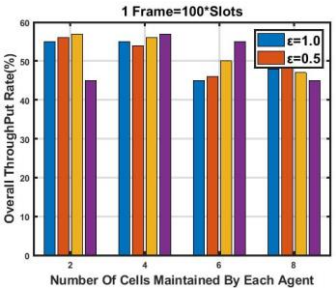

Figure 12. Training Performance Under Various Parameters

In this article, we will step-by-step check some important parameters during training, as shown in Figure 12. Service frequency has a significant impact on exercise performance. So, set 100 slots in each frame. System performance improves as the number of cells managed by each node increases. However, since nodes manage cells that require an additional resource cost, we set the number of cells managed by each node to 6. As one of the key characteristics of SLA priority, the dynamic resource percentage determines the cell's delay guarantee. The set of $\varepsilon = 1.5$ because the heavy weight of the edge asset portion means that the impact of the core network resource is negligible.

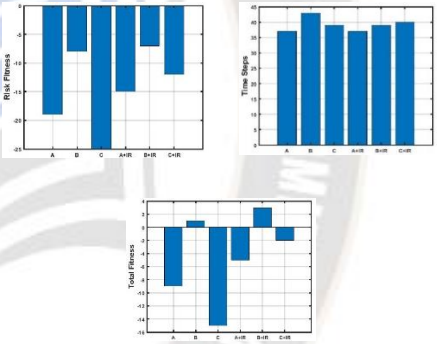

Figure 13. Average Performances of Different Agents

To assess the effect of introducing the inherent prize function on the extrinsic reward function, the average presentation of different agents on each criterion across 1000 test scenarios was calculated as shown in Fig. 13. Figure 9 illustrates, the regular recital of the IRDDPG agent is superior to that of the DDPG agent for both risk suitability and overall suitability criteria, but in conditions of time steps the two algorithms show similar performance. showed. Additionally, Figure 8, observed the difference in performance of different agents on each criterion. The average performances of agents A, B, and C show that different weight contributions to the external reward function strongly influence policy performance. However, the formation of a unique praise characteristic reduces the performance difference among unique guidelines because of the range of weight contributions.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

VI. RESEARCH CONCLUSION

SDN is a classic centralized network architecture for managing and operating networks. Logically centralized control with one controller simplifies network management and facilitates network problem resolution. The main reason for this migration is to boom the number of programmable visitors and boom the ability of community control operations such as community tracking, load balancing and strength control. Security resources provide many security advantages to an SDN network, but they may not be deployed in physical locations that can best accommodate the diverse and growing security needs of different users. SDN offers the potential for flexible use of security resources within the network. Therefore, this article, investigated safe m-multipath selection using DKNN and reef optimization techniques. Furthermore, DRL is applied to dynamic resource scheduling in SDN. Choosing the best SDN controller for the network is one of the most important tasks. For this purpose, the study used his open-source SDN controller, ODL. Additionally, controller placement is a critical process in large-scale SDN. The SDN performance is maximized and controller placement reduces average SDN latency. It uses meta-heuristics-based optimization techniques to improve throughput and minimize latency. The optimal solution for placing the controller under the multi-objective function is achieved by the optimization algorithm Simulated Annealing with Modified Dragonfly. Simulation findings demonstrate that the suggested strategy outperformed other cutting-edge research in multiple controller combinations. Networking is done to improve connectivity and scalability among controllers and switches. It then makes use of an optimization set of rules to choose the satisfactory controller primarily based on controller abilities to enhance community performance in SDN environments with the best reliability of 0.96 for every.

REFERENCES

[1] H. D. Praveena, V. Srilakshmi, S. Rajini, R. Kolluri, and M. Manohar, “Balancing module in evolutionary optimization and Deep Reinforcement Learning for multi-path selection in Software Defined Networks”, Phys. Commun., vol. 56, no. 101956, p. 101956, 2023.

[2] H. Qi et al., “SDN-based dynamic multi-path routing strategy for satellite networks”, Future Gener. Comput. Syst., vol. 133, pp. 254–265, 2022.

[3] B. Mohankumar and K. Karuppasamy, “Honesty aware congestion concerned secured edge disjoint multi-path routing with fuzzy rule descriptors”, Journal of Intelligent & Fuzzy Systems, vol. 43, no. 3, pp. 2219–2229, 2022.

[4] D. Jeyaraj, J. Yesudhasan, and A. A. S. Aliar, “Developing multi-path routing protocol in MANET using hybrid SM-CSBO based on novel multi-objective function”, Int. J. Commun. Syst., 2022.

[5] H.-J. Hong, S.-Y. Chang, and X. Zhou, “Auto-tune: An efficient autonomous multi-path payment routing algorithm for Payment Channel Networks”, Comput. Netw., vol. 225, no. 109659, p. 109659, 2023.

[6] L. Q. Chen, J. Q. Chen, Q. Y. Chen, and Y. L. Zhao, “A quantum key distribution routing scheme for hybrid-trusted QKD network system”, Quantum Information Processing, vol. 22, 2023.

[7] A. O. Alzahrani and M. J. F. Alenazi, “ML-IDSDN: Machine learning based intrusion detection system for software-defined network”, Concurr. Comput., vol. 35, no. 1, 2023.

[8] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, “Securing IoT and SDN systems using deep-learning based automatic intrusion detection”, Ain Shams Engineering Journal, 2023.

[9] M. Kumar Pulligilla and C. Vanmathi, “An authentication approach in SDN-VANET architecture with Rider-Sea Lion optimized neural network for intrusion detection”, Internet of Things, 2023.

[10] P. Krishnan, K. Jain, A. Aldweesh, P. Prabu, and R. Buyya, “OpenStackDP: a scalable network security framework for SDN-based OpenStack cloud infrastructure”, Journal of Cloud Computing, vol. 12, no. 1, 2023.

[11] N. S. Radam, S. T. F. Al-Janabi, and K. S. Jasim, “Using Metaheuristics (SA-MCSDN) Optimized for Multi-Controller Placement in Software-Defined Networking”, Future Internet, vol. 15, no. 1, 2023.

[12] A. A. Qaffas, S. Kamal, F. Sayeed, P. Dutta, S. Joshi, and I. Alhassan, “Adaptive population-based multi-objective optimization in SDN controllers for cost optimization”, Phys. Commun., no. 102006, p. 102006, 2023.

[13] D. Angelo and G. Palmieri, “A Co-evolutionary genetic algorithm for robust and balanced controller placement in software-defined networks”, Journal of Network and Computer Applications, 2023.

[14] S. R. Deepu, B. S. Shylaja, and R. Bhaskar, “Convergence Time Aware Network Comprehensive Switch Migration Algorithm Using Machine Learning for SDN Cloud Datacenter”, Big Data, Cloud Computing and IoT: Tools and Applications. 2023.

[15] J. Singh, P. Singh, M. Hedabou, and N. Kumar, “An Efficient Machine Learning-based Resource Allocation Scheme for SDN-enabled Fog Computing Environment”, IEEE Transactions on Vehicular Technology, 2023.

[16] J. Liu, J. Xu, S. Li, X. Cui, and Y. Zhang, “A secure multi-path transmission algorithm based on fountain codes”, Trans. Emerg. Telecommun. Technol., vol. 33, no. 5, 2022.

[17] Rohitaksha, “CSE/JSS Academy of Technical Education/Bengaluru, India,560060, A. B. Rajendra, and J. Mohan, Routing in hybrid software defined networking using hop-count approach”, Int. J. Wirel. Microw. Technol., vol. 12, no. 3, pp. 54–60, 2022.

[18] N. Firouz, M. Masdari, A. B. Sangar, and K. Majidzadeh, “A hybrid multi-objective algorithm for imbalanced controller placement in software-defined networks”, J. Netw. Syst. Manag., vol. 30, no. 3, 2022.

[19] W. Sun, S. Guan, P. Wang, and Q. Wu, “A hybrid deep learning model based low-rate DoS attack detection method for software defined network”, Trans. Emerg. Telecommun. Technol., vol. 33, no. 5, 2022.

[20] J. Kaur and D. Singh, “Optimizing Packets of Software Defined Network Using Hybrid Optimization Techniques”, in 2022 2nd

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7932
***Article Received**: 01 May 2023 **Revised**: 28 June 2023 **Accepted**: 27 July 2023*

International Conference on Innovative Practices in Technology and Management (ICIPTM), vol. 2, IEEE, 2022, pp. 344–349.

[21] L. Yang, Y. Song, S. Gao, A. Hu, and B. Xiao, "Griffin: Real-time network intrusion detection system via ensemble of autoencoder in SDN", IEEE Trans. Netw. Serv. Manag., vol. 19, no. 3, pp. 2269–2281, 2022.

[22] M. S. A. Muthanna, R. Alkanhel, A. Muthanna, A. Rafiq, and W. A. M. Abdullah, "Towards SDN-enabled, intelligent intrusion detection system for internet of things (IoT)", IEEE Access, vol. 10, pp. 22756–22768, 2022.

[23] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system", Comput. Electr. Eng., vol. 102, no. 108156, p. 108156, 2022.

[24] H. Bour, M. Abolhasan, S. Jafarizadeh, J. Lipman, and I. Makhdoom, "A multi-layered intrusion detection system for software defined networking", Comput. Electr. Eng., vol. 101, no. 108042, p. 108042, 2022.

[25] T. H. Q. Bao, L. T. Le, T. N. Thinh, and C.-K. Pham, "A high-performance FPGA-based feature engineering architecture for intrusion detection system in SDN networks, in Intelligence of Things: Technologies and Applications", Cham: Springer International Publishing, 2022, pp. 259–268.