International Journal of Emerging Multidisciplinary Research And Innovation (IJEMRI)

An efficient Resource Allocation framework using federated LSTM for Network Function Virtualization

[1] A. Kavitha, and [2] Dr.M. Gobi

[1] Research Scholar, Department of Computer Science,
Chikkanna Government Arts College, Tirupur, Tamil Nadu.
[2]Associate Professor, Department of Computer Science,
Chikkanna Government Arts College, Tirupur. Tamil Nadu, India.

ABSTRACT

Network Function Virtualization (NFV), a new technology emerging within the telecommunications space, get lower operating and capital costs while enabling service deployment. By co-locating various types of network appliances on standard, high-volume servers, switches and storage across the industry, NFV provides a means to resolve challenges associated with network services, by using standard information technology (IT) virtualization technologies. In addition, researchers are still continuing to solve other challenges of NFV, these research challenges include managing and orchestrating Virtual Network Functions (VNFs), chaining services through service chain (SFC), scheduling VNFs for low latency with minimum overhead, and providing efficient allocation of virtual network resources/functions within the Network Function Virtualization Infrastructure (NFVI) to create a flexible and dynamic network that supports multiple demands. Another significant challenge in the NFV space is the inability to support the current levels of inefficiency, SFC complexity, increased latency, load imbalance and poor scalability prevalent today in NFV systems. The paper proposes a new intelligent NFV resource allocation framework based on Long Short-Term Memory (LSTM) networks with Federated Deep Reinforcement Learning (FDRL) to overcome the above-mentioned challenges. The process of the proposed solution consists of multiple phases as follows: VNF placement and resource are scaling using Deep Reinforcement Learning (DRL) to facilitate dynamic and adaptive NFV management and executing (performing) the proposed intelligent NFV resource allocation solution through implementation of the above-mentioned DRL Federated Learning (FL) can be done through distributed model training in order to provide privacy-preserving model optimization without directly sharing model data and LSTM-based traffic prediction, which provides predictions of future traffic demand based upon a historic pattern of NFV resources used. By using both methods together to solve this problem, we expect to create a more efficient network, with reduced latency, and improved usage of available resources.

Keywords: NFV, VNF, Predictive Modeling, FL &DRL, LSTM.

DOI: https://doi.org/10.65180/ijemri.2026.2.2.04

INTRODUCTION

Traditionally, network services consist of various network functions that use hardware middle boxes to function. Network functions are referred to as Firewall (FW), Load balancer (LB), Intrusion Detection System (IDS), Network Address Translation (NAT). Network Function Virtualization (NFV) allows the use of software-based network functions/middle boxes wherein Virtual Network Functions (VNF) operate in the environment of servers/Virtual Machines (VM). By using this method, mobile service providers can reduce both their capital expenditures as well as their operating costs while providing more flexible network services for mobile voice and data, as well as internet access. In addition to the benefit of flexibility resulting from Network Function Virtualization; resource allocation for VNF is a major issues that creates a challenge for establishing resource allocation (i.e., NFV resource allocation). The VNFs will flexibly fit in to physical resources providing the appropriate Server Providing Corresponding VNF Function. However, due to the complexity of resource allocation for NFV spectrum resources there are many issues associated with VNF placement and Scheduling that cannot be strictly defined. The VNF Resource Allocation is broken down into three primary sub-tasks. Each sub-task contains several activities contributing to the completion of each primary task. The three primary VNF Resource Allocation components will be VNF Decomposition, VNF Placement, and VNF Scheduling. There are many studies that show effective implementation of methods to minimize end-to-end latency associated with NFV; however, there are currently very limited studies focused on VNF Resource Allocation based on Network Function virtualization to date. Despite initial development, all these attempts target to reduce the NFV latency in vertical scope.[3]Such assumptions make the VNF orchestration more manageable because of the Quality of Service(Qos) for NFV service.

In recent times, there have been some progress regarding the numerous parameters, which include the service functions chains, the number of flows and the lengths of flows etc., amount of the resources assigned to nodes, the quantity of nodes, and the Virtual Network Function (VNF). Also, there has been some progress with the resource allocation problem for Virtual Network Functions, in terms of the number of technologies that can be used to model these applications. Mixed integer linear programming will provide the best results when trying to obtain different goals ([4]). However, performance (for example, bandwidth, and link delay) and flow type, etc., should be evaluated for the mixed integer linear programming models ([4]). For optimal planning of VNF Orchestration for resource constrained networks, the proposed parallel placement orchestration will be employed ([2]). VNF parallel placement orchestration and optimization plans in this area will be implemented using a FL and DRL algorithm. Following this, delay-based methods were developed, so that orchestration can be achieved as developed in that framework ([2]). Despite avoiding the transmission of raw data, recent research has demonstrated that Federated Learning (FL) models present privacy threats. Increased latency associated with using FL models combined with the cost of increased resources can be detrimental to the performance of FL models.[3]. The problem for FL models is in regards to "catastrophic forgetting", which occurs because updated models do not retain previously learned knowledge when adapting to changes in the environment as a result of new data being introduced. Consequently, cyber attackers are able to obtain private information from updates of FL models, which means that preserving privacy in FL models will be necessary by means of various "privacy preserving techniques".

The architecture described in this paper is a combination of Federated Learning (FL), Deep Reinforcement Learning (DRL), and Long Short-Term Memory (LSTM) networks. The system is designed to deal with the dynamics of the network, the limited resources available to the system and the lag time constraints that exist. The static placement rule that is in place with LSTM based on predictive learning will provide for predictions of the future resources available to the system with more frequent use, therefore providing for better placement decisions that balance latency and efficient use of resources.

RELATED WORKS

Over the last years, several proposals have been devised to solve the NFV architecture framework to efficient VNF scheduling and VNF placement for resource allocations. Given the way VNFs are executed, these past activities are classified as illustrated below.

They were led by B. Han, V. Gopalakrishnan, L. Ji, and S. Lee who showed how Network Function Virtualization (NFV) can make faster deliveries of new services possible by introducing a flexible way of delivering network services, thus reducing time to market. By using existing virtualization technologies, by re purposing commercial off-the-shelf (COTS) hardware, including general purpose servers, storage and switches, NFV allows software implementation of network services to be decoupled from the hardware on which they run. As a new technology, NFV presents a number of challenges for network operators such as how to maintain the performance of the virtual appliances, how to dynamically migrate and instantiate them and how to effectively place them.[1]

This is the prerogative of Rashid Mijumbi, Joan Serrat, Juan Luis Gorricho, Niels Bouten, Filip De Turck and Steven Davy,illustrated, who have specifically tackled how to build virtual networks from the physical network, and then edge functions into the virtual network. In this paper the problem of online virtual function mapping and scheduling is formulated, as well as an array of potential solution techniques is presented. This research aims at furnishing fundamental algorithms to be used as a foundation for further studies in this field. It analyzed them under different network conditions, considering the various measures such as revenue, costs, total service processing times, successful service mapping etc. Based on simulation results, less than 1% difference is found between the best greedy algorithm and the tabu search based algorithm.

Long Qu, Chadi Assi, devised an algorithm to delay–aware schedule VNFs and allocate resources to a service chain. The goal of this paper was to reduce the total make span and latency of the scheduled VNFs by taking into consideration both VNF transmission and processing delay when formulating the dual problem of VNF scheduling and traffic steering as a Mixed Integer Linear Program (MILP). In doing so, Cloud operators will be able to support (and allow) more users, as well as accommodate services that have strict delay constraints by reducing scheduling latency, which will result in increased profits for operators. Considering the complexity of the dual problem, they devised a Genetic Algorithm (GA) based method to effectively solve the dual problem. Last, the authors tested the effectiveness of their heuristic algorithm numerically. The results show that the scheduled make span can possibly be reduced 15% – 20% if the bandwidths of the virtual links, which are created between virtual machines that provide the network services are adjusted dynamically. [3].

A. Suzuki et al. created an algorithm that is able to solve the problem of combined optimisation of NFV control, but this approach does not support extensibility because each time a new statistic is introduced or the combination of statistics is changed, this optimisation problem has to be re-cast. The second approach is to coordinate several different control algorithms (one for each measure), that are orchestrated by an extensible network control architecture. Until now the only known way to achieve maximum resource allocation in this way is for multiple algorithms. The aim of this research is to develop a coordinated method for application of many control algorithms and produce a flexible NFV integrated control method and also present the workable coordinated methodology which has the proof that after operation of less than 10000 steps link utilization will decrease by more than 50% after many steps of operation.

Nahida Kiran, Xuanlin Liu, Sihua Wang, Changchuan Yin, formulated a proposed a VNF Placement and Routing Algorithm (VNFPRA) problem which involves placing virtual network functions (VNFs) in a multi-access edge computing (MEC) environment with software defined networking (SDN) and network function virtualisation (NFV) functionality optimally such that deployment and resource costs are minimised. To solve this problem, the two algorithms they describes, are: (i) A mixed integer programming (MIP) optimal formulation of the problem, and (ii) A genetic algorithm-based heuristic solution for the same problem. They also formulated an algorithm and named it as GA-VNFM that is used to minimise total number of VNF migrations. The authors concluded that their solution methods comparing with four existing algorithms from the literature proved that they had higher performance in comparison to those existing algorithms. They also find out that coordination of the placement of VNFs in SDN, NFV, and MEC will be able to achieve the overall cost saving objective [8][9][10].

The online backup function in Edge computing has been studied by J. Zhang, Z. Wang, C. Peng, L. Zhang, T. Huang and Y. Liu and analysed as an optimal solution reducing the cost and maximising availability using the Volatile Data System model based upon empirical data, empirical evidence to prove the difficulty of the Problem has also been provided. For our experiments, Dr. C. Chinnaswamy tweaked the original problem to create a problem called DPP (Drift-Plus-Penalty), where he used the constraints of the Original Vehicles Backup and Ascent Problem, and additionally added some constraints related to the drift. The vast experimental simulation results show that the DPP obtained a better performance than many well-known solutions that are in use today in the area of Data Storage [11][12][13][14][15].

The subproblem of the NFV-RA problem, as proved by Zhiyuan Li, is NP-Hard and most of the available solutions are heuristic and meta-heuristic algorithms. In this paper an NFV online coordinated resource allocation framework (OCRA) is presented which brings together novel neural networks and RL training methods with parallel Multi-Agent Deep Reinforcement Learning to perform all three steps at once and in a coordinated manner. Numerous experimental results suggest that OCRA is much more time efficient than the best solutions available so far and enhances resource overhead and acceptance ratio by at least 50% and by 10.8%, respectively [16][17][18][19][20].

The coordinated strategy for NFV resource allocation optimisation of Wang et al. developed an almost optimal solution to the coordinated resource allocation management of NFV by L. Wang, Z. Lu, & R. Knopp (Wang et al.) has developed an almost optimal solution through a coordinated strategy in three phases. Wang et al. describes the developed solution to use a general cost model for both network costs and service performance. Two sub-problems of NFV resource allocation management are proposed, called CNFVRAM algorithm: 1) A multi-path greedy algorithm of VNF chain composition and VNF forwarding graph placement. 2) Optimal traffic scheduling with a single hop. Using extensive simulations, the performance of JoranNFV is thoroughly assessed and the results are empirically verified. Based on detailed analysis of these simulations, it is shown that the JoranNFV algorithm yields a solution, within 1.25 time of the one obtained by the optimal solution, within an acceptable time frame. Based on the obtained results it may be noted that the use of the JoranNFV algorithm is a helpful approach towards the management related to the allocation of NFV resources.

R. Mijumbi, S. Hasija, S.Davy, A.Davy, B.Jennings, and R.Boutaba estimated the future needs of each of these VNFCs and presented a graph neural network based approach to encapsulate the VNF forwarding graph topology information, which consists of the resource consumption history of the VNFC, and the VNFC effect prediction of its neighbouring VNFCs. The new methodology was tried with a scenario of a virtualized IP multimedia subsystem deployment and employing real VoIP traffic data, and the average(prediction) accuracy was found between 90% and 85% for a typical feed-forward neural network. Moreover, this new method decreases call setup latency by more than 29% and at least 27% of average dropped calls are avoided, as opposed to cases where the resources are allocated manually and/or statically. [22][23][24]

Stefan Schneider, Narayanan Puthenpurayil Satheeschandran, ManuelPeuster and Holger Karl explained that machine learning models trained with real VNF data consists of VNF performance and resource metrics, use to precisely predict what resources each VNF will need and how they will be placed based on traffic load. After the ML models were trained and integrated into the joint scaling and placement algorithm, the final VNF placements were assessed based on the impact MR and proper ML models had on mitigating over- or under-allocation of resources, relative to traditional fixed resource allocation methods. Based on our analysis of real-world workloads, using the proper ML models can reduce overall physical resource consumption up to 12 times when compared to traditional methods, while also enabling VNF/service improvements by reducing overall total delivery delay times of services by up to 4.5 times.[25]

Mohammad Bany Taha to improve its overall management capabilities and performance. In addition, the nature of fluctuations in the cloud service as well as peak demand of heavy workloads require appropriate techniques to handle the extensive variety of both normal and abnormal workload characteristics. This study proposes an intelligent proactive auto-scaling system based on a deep learning algorithm that will help with these types of issues in an efficient manner. Specifically, the proposed auto-scaling system is tested using a real world dataset related to VNF within service function chains (SFC) at a datacenter by evaluating the performance of five different Deep Learning (DL) approaches in the context of predicting demand for instances of VNFs within the SFC including: CPU, Memory and Bandwidth, and also creating a Hybrid model of the Multi-layered Perception (MLP) and Long Short-Term Memory (LSTM) algorithms for use in an on-line model to forecast future demand for VNF instances in SFC. Finally, by determining the autocorrelation function (ACF) for the on-line model and using a Linear Regression model with the ACF to identify abnormal demand events, the proposed system provides an alternative means of identifying abnormal demand events through the use of regression modeling techniques.[26]

Felix Hurmat Ali Shah, Lian Zhao, set a challenge to optimise the Markov decision process (MDP) of placing service function chains (SFC) under the constraints in the IoT networks. To do this, a multi agent deep reinforcement AI (DRL) algorithm was required where each agent is responsible for the execution of an SFC. Evaluation of two Q-Networks was undertaken to address the MDP problem: One, placed in the parameter space of SFCs placement, and the second, one that would modify the weight of the Q-Network in view of future policy changes, by evaluating the long term behaviour of it. Virtual agents "serve" SFCs together and share rewards, so that by sharing with each other what they learn, the agents' fight for rewards drives collective policy, its effectiveness enhanced by their shared experience. This designed system will offer solutions for SFCs placement optimisation with the adequate design of Incentives, State spaces and action spaces. The simulation results demonstrate that the proposed multiagent DRL system has better performances than the benchmark systems according to network parameters and the utility generated.The simulation results indicate that the proposed multiagent DRL system can perform better than the benchmark systems in terms of network parameters and utility generated.

Unlike many existing solutions that adopt an iterative deployment approach, MSV does not need to enclose a maximum number threshold for the number of VNF instances, but the number of VNF instances can be automatically calculated. They also chose to explore the various improvement approaches and avoid being easily trapped in local optimal solutions, and do not need to use complex anti-local-optimal solutions. Compared to the MIP approach, extensive testing demonstrated that MSV is able to provide solutions over the global range with acceptable execution time, and that a total cost ratio of less than 115% could be achieved. [28]

Liu, Shu, Chen, Zhong, and Lin shed light on the challenges of mapping, scheduling and routing VNFs (virtual network functions) with latency constraints on SFCs (service function chains) in a virtual network environment. The authors then proposed a VNF scheduling algorithm which combines the mapping, scheduling and routing of VNFs to provide for minimum SFC rejection rate.

With this objective in sight they developed a VNF scheduling model based on MDP (Markov Decision Process) that meets the resources for SFCs. This model was used by the authors to choose SFCs by employing the D3QN (Dueling Double DQN) algorithm at every scheduling interval in accordance with some common composite rules. They then applied the routing optimization method for virtual node and routes selection and minimization of rejection of SFCs. Results demonstrated the estimate of the proposed algorithm compared to the DQN single rule algorithm and the genetic algorithm, reduced SFC rejection by almost 8% more than the genetic algorithm.

Haojun Huang , Jialin Tian, Geyong Min using Federated Deep Reinforcement Learning (FDRL) model, a unique Parallel VNF Placement (PVFP) method is suggested for real-world networks. It has been developed to carry out the optimal orchestration of VNFs in resource-constrained networks. There are four finding had analyzed as loss function, local reward, average and end to end latency and resource overhead. According to simulation results in various scenarios the local training rate is to be either 0.0001 (0r) 0.001 for the small scale networks. After 1100 training epochs, Next PVFP reduced more network cost compared to the large -scale networks. The end-to-end latency of SFCs can be considerably decreased using PVFP at medium resource expenditures. VNF relationships and local DRL optimization enable it to overcome the latency issue while maintaining the resource usage, significantly outperforming ParaSFC, GSS, NCO and Gecode.[30]

However, FDRL- LSTM models utilized in the all above models have catastrophic forgetting issues. This is solved in the proposed model.

System Model and Problem description

In this section, we provide a foundational introduction referring to the system models, VNF parallel placement and problem formulation to understand our proposed FDRL-LRTM.

System models

The physical network is defined by a directed graph G = (N, L),

N = set of nodes and L = set of links. (1)

N U {node1, node2, node3, node4, etc,.} (2)

NE= end nodes, ND= distribution nodes, (3)

NS=server node (4)

Server node Resource Capacity={cpu,memory,Latency} (5)

Link in the Network = {Bandwidth} =BW (6)

Service Chain S={s1,s2,s3….sn} (7)

Set of VNFs in the service chain s is represented by F = {f 1, f 2, ..., fm}, (8)

Where m denote the number of VNFs in

The service chain. Service request1 (SFC)

Si={Fi,CPUi,RAMi,LATENCY,BWi,Fi1,CPUi1,RAMi1,LATENCYi1,BWi1} (9)

Service request 2(SFC)Sj={Fj,CPUj,RAMj,LATENCYj,BWj,Fj1 ,

The traffic flow of the SFC request I is transferred according to its VNF-FG Gi = (Ni , Li).

In a VNF-FG, the parameters f1i, f2i ∈ Li denote two VNFs and f1j f2j ∈ Li . (10).

The framework contains parallelism. Decomposition, local training and aggregation. The time scale is created based on aggregation episodes={T1,T2….}.The whole network divided into K domains each has installed an iteratively learned local model (i€[l,k]).

Problem description

The former approach was based on using Reinforcement Learning (RL) to use Service Function Chains (SFC) for optimization of the placement of Virtualized Network Functions (VNF). The means to define the SFCs has been to break them down into small elements; however, this process adds a layer of computational overhead, especially when the environments are dynamic.

To alleviate this issue, the model allows for the automatic scaling of VNFs through predicated analysis on the workload patterns and reducing wasted resources, among other approaches. With the rapid evolution of the 5G network and the adoption of cloud-native architectures, Network Function Virtualization (NFV) is becoming an important enabler for the effective, scalable, and adaptable allocation of resources. Current methods for resource management of NFV have numerous issues including high latency, inefficient utilization of resources, vulnerabilities to security threats, and a general lack of flexibility to handle changes in workloads. Traditional AI-based methods, specifically centralized deep learning models, have disadvantages such as limited capability to adapt to distributed environments, large processing requirements, and privacy issues. Furthermore, current federated learning-based methods display communication costs that are too high, they have difficulties with convergence to a common model, and they operate on non-Independent and Identically Distributed (non-IID) data types. These limitations require a comprehensive framework that will provide guarantees to scalability within a large NFV deployment while providing optimal resource allocation, enhanced security, and reduced latency.

FDRL-LRTM: Dynamic VNF Placement

The aim of this study is to create an FDRL-LSTM framework that will provide NFV resource allocation by using FDRL together with LSTM to allow for real-time, adaptive, and privacy-preserving management of NFV. The developed framework will utilize LSTM for traffic forecasting (time series), Federated Learning (FL) for decentralized training, and Deep Reinforcement Learning (DRL) to optimally place Virtual Network Functions (VNFs) within an NFV environment. The ultimate goal of the study will be to improve the efficiency of resource allocation, scalability and performance efficiency of NRV to support NFV-based networks.

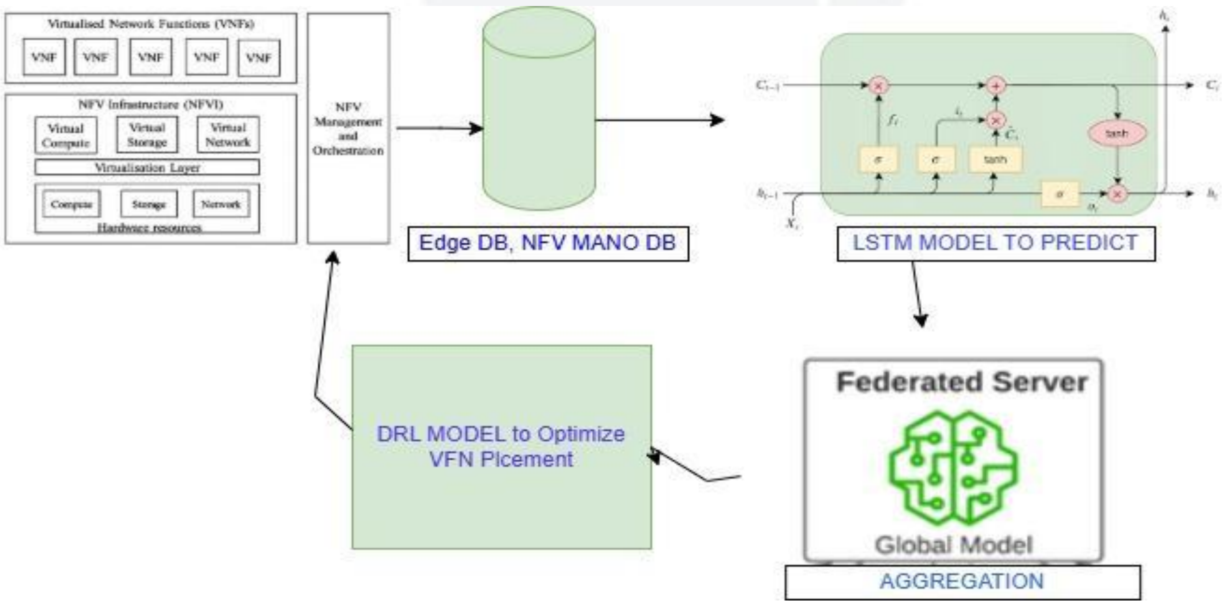

Fig 4.1 Architecture of Fdrl-Lstm Model

Framework of FDRL-LSTM

The primary purpose of the LSTM works is to collect data at the NFV Infrastructure (NFVI) level for use in decision-making. NFVI gathers real-time network statistics, such as CPU usage, bandwidth consumption, latency and packet loss through Virtual Network Functions (VNFs) and network telemetry systems. This real-time data is necessary to analyse the network activity and to predict future traffic patterns. The Data collected will be stored in a cloud-based storage repository for future historical analysis, as well as accessed through Edge Databases for real-time access to the data being collected. The resources being predicted using the historical data include Virtual CPUs (CPU Utilization), Bandwidth Usage, Memory Utilization, Latency and computing power, etc. The LSTM is to be run in a highly distributed NFV environment composed of several edge nodes, all running VNFs, where each edge node will collect real-time metrics according to the temporal dependencies of the data collected. Based on a DRL-generated focus policy map, any following storage of the data will be pre-processed and feature extracted to remove any excess noise and determine which features are most relevant and should be used as input to the LSTM model LSTM, a type of recurrent neural network (RNN), has been shown to successfully identify long-range dependencies in time-series data. LSTMs can forecast future network resource requirements based on historical traffic patterns, enabling proactive resource allocation. This can help reduce latency, prevent network congestion and improve service reliability.

Federated Learning (FL) is used in the proposed framework for secure and decentralized learning. Each NFV node will build its own local LSTM model using only its own data, rather than sending raw network data to a central server. The trained models' parameters will be sent to a global federated learning server. This server will aggregate the updates from all NFV nodes and produce a new global model. Therefore, the global model accuracy is improved and data transportation costs are reduced, while the data privacy remains secure. Once the new global model is created, it is redistributed back to all individual NFV nodes for ongoing improvements in prediction accuracy.

Federated Learning framework is offering a decentralized method of processing data as it is created from both a local server and nodes. A local model will be trained using local data. Instead of sending raw data to the Server, like with the Local Model the data will be uploaded in intervals for use in a "Local Process" where it will then be sent from "Packaged" to all nodes after receiving consent from everyone involved. This process will continue until there is a "Global Model" that meets the required criteria. The use of Federated Learning will keep Data Private & Reduce Communication Needed. Additionally, since Sensitive Data won't Leave the Local node(s), optimistic outcomes will occur. Also, using this Adapted Approach can have an impact on the Health Sector, IoT and NFV Systems due to the importance of the Confidentiality & Security of Data when it is partitioned, as well as the Diversity of Types of Information Associated. These Architecture's have the capability of Addressing the Challenges Related to Engineering difficulties such as the Efficiency of Communications, Staleness of the Model, & Changes in Power Levels Across Multiple Nodes. Finally, Once all of the "Gradients" on the Model Weights are sent to the Server, Aggregate up to create a System for VNF Placement Optimization through Star-Nodes based off of the new idea of "Fed Avg" Weighted Model Calculation. The Next Step for the System will be to Implement Deep Reinforcement Learning for VNF Placement Optimization. The agent (DRL) utilizes the predictions generated from the LSTM-FL model to determine where to optimally place the VNF for reducing latency and maximizing resource utilization in order to support the DRL agent in continually learning how to interact with the NFV environment to understand the optimal deployment, scaling, and migration of VNFs based on network demand and available resources. The DRL agent is also designed to optimize the reward function so that it supports improved network performance, while also reducing operating costs and energy consumption.

During the second part of the framework, the NFV MANO (Management and Orchestration) system plays a vital role by deploying VNFs based on the current state of the network and dynamically allocating resources according to the information provided by LSTM and DRL models. The framework is also designed to monitor the performance of VNFs continuously, allowing the system to adjust according to the current service quality, traffic patterns, and resource consumption. The feedback loop allows for LSTM and FL model improvements based on real-time performance data from the framework, ensuring that the system can continue adapting to changing network conditions.

Algorithm For Fdrl-Lstm

Input: N (number of nodes), T (global rounds),

E (local epochs),

D_i (local datasets)

Output:Optimized global policy πθglobal

1: Initialize local models πθi with random weights for each node i

2: Initialize global model πθglobal

3: For t = 1 to T do:

4: For each node i in parallel do:

5: Collect real-time resource metrics D_i

6: Train local LSTM-DRL model πθi for E epochs: 7: For each episode in D_i:

8: State S(t) ← Collect sequential metrics

9: Action A(t) ← Policy πθi(S(t))

10: Reward R(t) ← Compute based on resource utilization and latency

11: Update policy πθi using reinforcement learning 12: End For

13: Save updated weights Wi

14: End For

15: Central server aggregates weights:

16: Wglobal ← (1 / N) Σ Wi

17: Central server redistributes Wglobal to all nodes

18: Each node updates πθi with Wglobal

19: End For

20: Deploy πθglobal to all nodes for real-ti Decision-making

21: Return πθglobal

Result and Evaluation

The dataset is obtained from Kaggle and allows users to develop models to forecast where VNFs (Virtual Network Functions) will be placed in the future. As a result, there is a wealth of information about the number of variables in the dataset that can be utilized in more than one analysis. The metrics found in this dataset can also be evaluated in scenarios where hypothetical values are entered.

Performance measures

1. Latency (ms): Latency refers to the time it takes for a system to respond to a request or process a task. It can be calculated as:

L=(NTresponse−Trequest)/N.Measures response delay. Lower is better.

2. Throughput (Mbps): Throughput refers to the amount of data successfully transmitted or processed per unit of time.
Throughput=Total data transmitted\Total time taken

3. CPU Utilization (%): CPU utilization measures the percentage of CPU resources used by an algorithm.
CPU=(Ctotal/Cused)×100,Percentage of CPU resources used by VNFs. Higher is better for efficiency.

4. Memory Usage (GB): Memory usage refers to the amount of memory consumed by an algorithm.
Memory Usage=Peak memory consumption during e xecution

5. Packet Loss (%): Packet loss is the percentage of packets that are lost in transmission due to network congestion, errors, or other issues.Ploss=(Psent-Preceived)/Psent×100 Measures network reliability. Lower is better.

6. Training Time (sec): Training time refers to the time required to train an algorithm (especially for machine learning models).
Training Time=Time taken to train the model on the dataset

7. Inference Time (ms): Inference time is the time taken by a trained model to make predictions or take decisions.

Inference Time=Time taken to make a prediction or decision using the trained model

Generally, FDRL-LSTM has performed well when measured against latency, throughput, memory consumption and packet drop Rate. However, there are concerns with respect to CPU utilization and inference time which both reside very near to but just above a desired optimal level of performance.

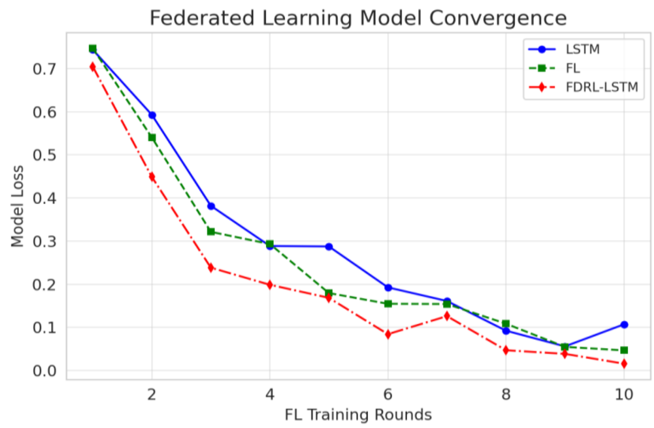

Fig 5.3.Model Loss

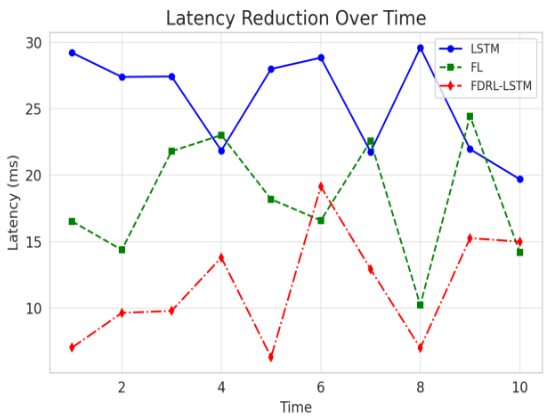

Fig 5.1.Latency Comparison

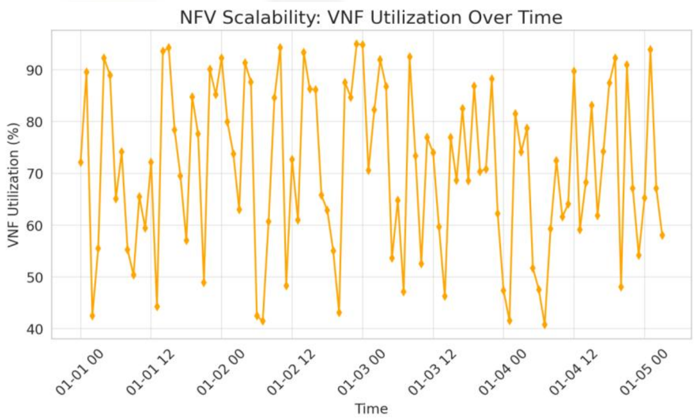

Fig 5.4.VFN Utilization

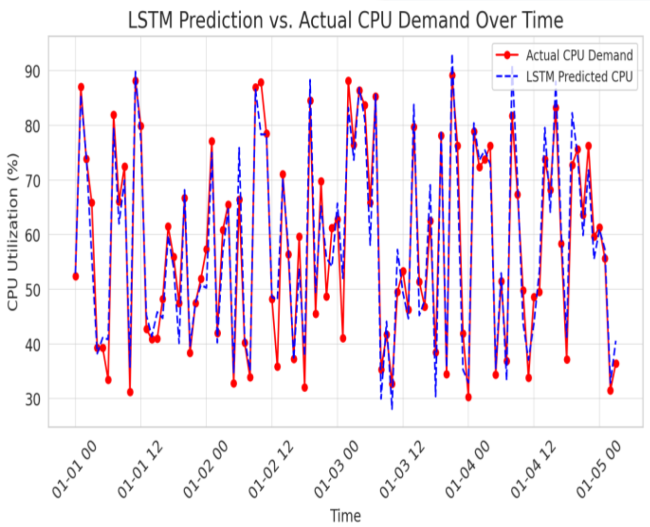

Fig 5.2.CPU Utilization

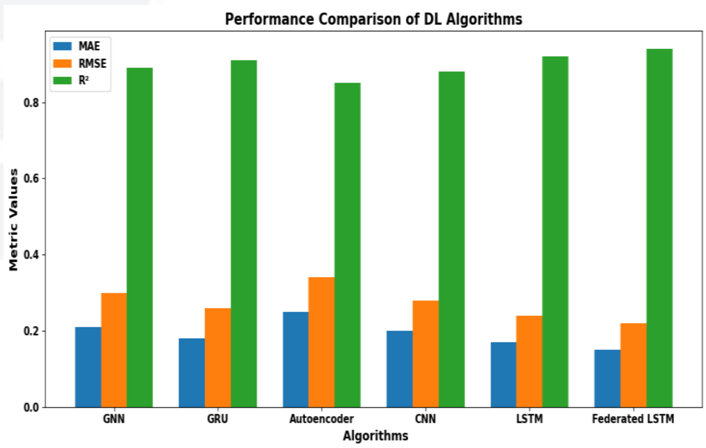

Fig 5.5.VFN Utilization

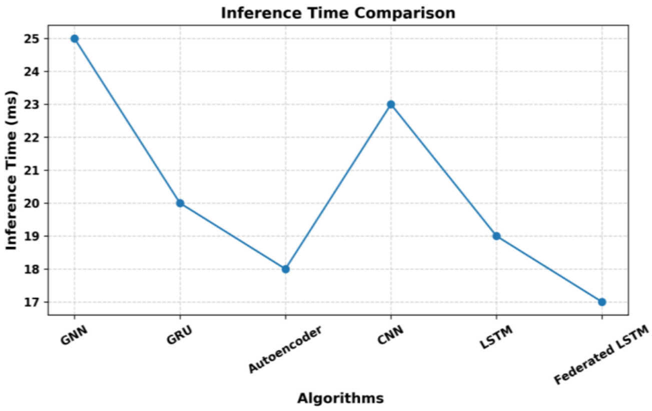

Fig 5.6.Inference Time Comparison

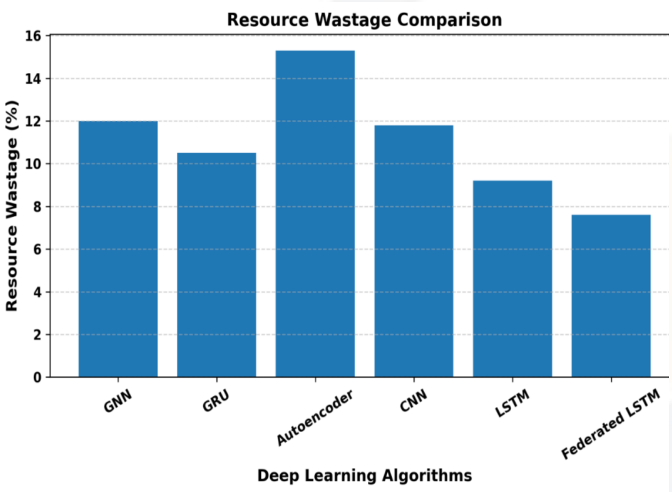

Fig 5.7.Ressource Wastage Comparison

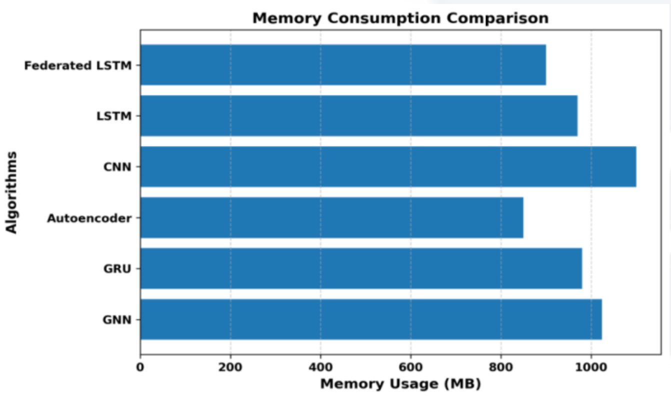

Fig 5.8.Memory Consumption Comparison

Conclusion

Utilizing decentralized learning along with predictive analytics, the Federated Deep Reinforcement Learning (FDRL) with LSTM framework can effectively optimize the allocation of resources in network function virtualization (NFV) environments. By employing Long Short Term Memory (LSTM) for traffic prediction, Federated Learning (FL) for collaborative training of the model, and Deep Reinforcement Learning (DRL) for optimizing decisions, our model achieve efficient, scalable, and adaptable resource management.

In addition, FL allows for the distributed training of the model across the NFV nodes rather than a centralized collection of data, which reduces the latency, bandwidth requirements, and security risks associated with collecting data centrally, while simultaneously protecting the privacy of the data being collected. The provision of DRL enables the allocation of resources according to changes in network demand furthermore, the enhancement of VNF scale decisions through LSTM forecasting improves the overall operational reliability of the system as a whole. The FDRL-LSTM framework provides a novel, scalable, and intelligent approach to resource allocation in NFV. By continuously evolving with cutting-edge AI-based techniques, it will provide significant improvements in the efficiency, flexibility and security of future cloud and network infrastructures.

REFERENCES

[1] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network Function Virtualization: Challenges and Opportunities for Innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[2] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and Evaluation of Algorithms for Mapping and Scheduling of Virtual Network Functions," in *Proc. IEEE NetSoft*, 2015, pp. 1–9.

[3] L. Qu, C. Assi, and K. Shaban, "Delay-Aware Scheduling and Resource Optimization with Network Function Virtualization," *IEEE*, https://doi.org/10.1109/TMC.2019.2942306.

[4] A. Suzuki *et al*., "A Method of Coordinating Multiple Control Algorithms for NFV," *IEICE Technical Report*, vol. 116, no. 485, pp. 37–42, 2017.
[5] N. Kiran, X. Liu, S. Wang, and C. Yin, "Optimising Resource Allocation for Virtual Network Functions in SDN/NFV-enabled MEC Networks," *IET Communications*, 2021.
[6] J. Zhang *et al*., "RABA: Resource-Aware Backup Allocation for a Chain of Virtual Network Functions," in *Proc. IEEE INFOCOM*, 2019, pp. 1918–1926.
[7] Z. Li *et al*., "Online Coordinated NFV Resource Allocation via Novel Machine Learning Techniques," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, Mar. 2023.
[8] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint Optimization of Service Function Chaining and Resource Allocation in Network Function Virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.
[9] R. Mijumbi *et al*., "Topology-Aware Prediction of Virtual Network Function Resource Requirements," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 106–120, 2017.
[10] S. Schneider *et al*., "Machine Learning for Dynamic Resource Allocation in Network Function Virtualization," in *Proc. IEEE NetSoft*, 2020.
[11] M. B. Taha *et al*., "Proactive Auto-Scaling for Service Function Chains in Cloud Computing Based on Deep Learning," *IEEE Access*, vol. 12, 2024.
[12] H. A. Shah and L. Zhao, "Multi-Agent Deep Reinforcement Learning Based Virtual Resource Allocation Through NFV in IoT," *IEEE Internet of Things Journal*, 2020.
[13] H. Li *et al*., "MSV: An Algorithm for Coordinated Resource Allocation in Network Function Virtualization," *IEEE Access*, vol. 6.
[14] Z. Liu *et al*., "Low-Latency Virtual Network Function Scheduling Algorithm Based on Deep Reinforcement Learning," *Computer Networks*, Elsevier, 2024.
[15] H. Huang *et al*., "Parallel Placement of Virtualized Network Functions via Federated Deep Reinforcement Learning," *IEEE/ACM Transactions on Networking*, vol. 32, no. 4, Aug. 2024.
[16] M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, "Efficient Placement of Service Function Chains in Cloud Computing Environments," *Electronics*, vol. 10, no. 3, 2021.
[17] J. Chen and R. Hu, "MOClusVNF: Leveraging Multi-Objective for Scalable NFV Orchestration," 2021.
[18] M. Moradi, M. Ahmadi, and L. PourKarimi, "Virtualized Network Functions Resource Allocation Using Mathematical Programming," *Computer Communications*, Elsevier, 2024.
[19] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near Optimal Placement of Virtual Network Functions," in *Proc. IEEE INFOCOM*, 2015, pp. 1346–1354.
[20] M. Aibin, "LSTM for Cloud Data Centers Resource Allocation in Software Defined Networks," Conference Paper, Oct. 2020.
[21] Z. Zaman *et al*., "Novel Approaches for VNF Requirement Prediction Using DNN and LSTM," https://doi.org/10.1109/GLOBECOM38437.2019.9014320.
[22] S. Selvi and S. Ganesan, "An Efficient Hybrid Cryptography Model for Cloud Data Security," *International Journal of Computer Science and Information Security*, vol. 15, no. 5, pp. 307–313, 2017.
[23] S. Selvi, M. Gobi, M. Kanchana, and S. F. Mary, "Hyper Elliptic Curve Cryptography in Multi-Cloud Security Using DNA Techniques," in *Proc. ICCMC*, 2017.
[24] S. Selvi and M. Gobi, "Hyper Elliptic Curve Based Homomorphic Encryption Scheme for Cloud Data Security," in *Proc. ICICI*, 2018.
[25] S. Selvi and R. Sridevi, "Efficient Scheduling Mechanisms for Secured Cloud Data Environment," *IJRTE*, vol. 8, no. 2S11, 2019.
[26] R. Hemalatha and S. Selvi, "Improving Security of Visual Cryptography by Contrast Sensitivity Function," *Vidyabharati International Interdisciplinary Research Journal*, 2021.
[27] S. Rekha *et al*., "A Reinforced PSO Algorithm with SJF-MMBF for Allocating Work in Cloud Computing," in *Proc. ICKECS*, 2024.
[28] V. Vishal and S. Selvi, "ZPHISHER Tool in Cyber Security," *IRJMETS*, vol. 6, no. 12, 2024.
[29] T. Lavanyan and S. Selvi, "Quantum Computing: A Threat to Cryptography," *IJRPR*, vol. 5, no. 12, 2024.
[30] S. Shaathvi and S. Selvi, "Transforming Social Media Marketing Using Artificial Intelligence and Machine Learning," *IJIRCCE*, vol. 12, no. 12, 2024.