Optimized Network Resource Management Using SDN Architecture by Hybrid Fuzzy Neural-Grey Wolf Algorithms in Cloud Environments

by

SeyedehGaliya REZADOUST

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS IN INFORMATION TECHNOLOGY ENGINEERING
M.A.Sc.

MONTREAL, JANUARY 09, 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Michel Kadoch, Thesis supervisor
Department of Electrical Engineering, École de technologie supérieure

Mr. Amin Chaabane, Chair, Board of Examiners
Department of System Engineering, École de technologie supérieure

Mr. Kim Khoa Nguyen, Member of the Jury
Department of Electrical Engineering, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON DECEMBER 20, 2024

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I want to express my gratitude to Professor Kadoch, who oversaw my thesis, for all of his help, advice, criticism, and encouragement during the investigation. We appreciate all of your helpful suggestions and unwavering assistance during this project.

In addition, I want to sincerely thank my parents for their love, support, and understanding during this trying period when I am away from home pursuing my academic objectives. I have the guts and drive to get through challenging times thanks to their support.

Lastly, I want to express my gratitude to my friends Zeinab and Hamidreza for their unselfish sharing of their time, expertise, and experiences as well as for their assistance and support during my academic career.

Gestion optimisée des ressources réseau à l'aide de l'architecture SDN par des algorithmes hybrides Fuzzy Neural-Grey Wolf dans les environnements cloud

SeyedehGaliya REZADOUST

RÉSUMÉ

Cette thèse examine la manière de mettre en œuvre une gestion dynamique des ressources dans le cloud grâce à l'architecture de réseau défini par logiciel (SDN). La complexité et la demande de services cloud s'accroissent, tandis que la gestion des ressources dans les réseaux traditionnels se heurte à des problèmes d'efficacité et d'adaptabilité.

Pour faire face à ces difficultés, nous proposons dans ce travail de recherche une solution novatrice en intégrant conjointement un algorithme hybride de réseau neuronal flou et l'optimiseur Grey Wolf (GWO) au sein du cadre SDN. Cet algorithme hybride améliore l'équilibrage de charge et la répartition des ressources dans les clouds en fonction des changements contextuels du réseau. Ainsi, nous pouvons comparer les résultats obtenus par la méthode proposée aux techniques conventionnelles, afin de démontrer les avantages potentiels en termes de meilleure satisfaction, de réduction de la congestion du réseau, d'optimisation des ressources et de diminution du temps de résolution des erreurs.

Cette recherche s'inscrit dans le domaine du SDN en abordant la problématique de la gestion des ressources et de l'équilibrage de charge pour des améliorations et des optimisations en temps réel du réseau, applicables notamment au cloud computing, à l'IoT et à la 5G.

Mots-clés: Gestion des ressources réseau, Informatique en nuage, SDN (réseaux définis par logiciel), Optimisation des ressources, Optimiseur du Loup Gris (GWO), Répartition de charge, Réseau neuronal flou hybride

Optimized Network Resource Management Using SDN Architecture by Hybrid Fuzzy Neural-Grey Wolf Algorithms in Cloud Environments

SeyedehGaliya REZADOUST

ABSTRACT

This thesis looks at how Dynamic Resource Management in the Cloud using Software Defined Network Architecture can be achieved. The complexity and demand for cloud services are increasing, and traditional network sources' management is confronting problems with efficiency and adaptability. To address these difficulties, we present an original solution in this research by implementing both hybrid Fuzzy Neural Network and Grey Wolf Optimizer (GWO) algorithms in the SDN framework. The hybrid algorithm enhances load-balancing and resource distribution in the clouds based on the network's contextual changes. We can thus compare the results achieved by the proposed method with the conventional techniques to show the potential benefits in terms of greater satisfaction, less congestion on the network, higher resource efficiency, and minimizing error resolution time.
This research fits into the line of work in SDN by addressing the challenge of resource management and load balancing to real-time network enhancements and optimization that concerns issues such as cloud computing, IoT, and 5G.

Keywords: Network Resource Management, Cloud Computing, SDN (Software-Defined Networking), Resource Optimization, Grey Wolf Optimizer (GWO), Load Balancing, Hybrid Fuzzy Neural Network (FNN)

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
AI	Artificial intelligence
ATM	Asynchronous Transfer Mode
DI	Degree of Imbalance
FEPSO	Fuzzy Evolutionary PSO
FN-GW	Fuzzy Neural-Grey Wolf Algorithm
FCFS	First Come First Served
FNN	Fuzzy Neural Network
FPSO	Fuzzy Particle Swarm Optimization
GWO	Grey Wolf Optimizer
GA	Genetic Algorithm
HFA	Honey Bee Foraging Algorithm
IoT	Internet of Things
IaaS	Infrastructure as a Service
MPLS	Multiprotocol Label Switching
NaaS	Network-as-a-Service
NN	Neural Network
ONF	Open Networking Foundation
OPU	Optimal Power Utilization
OSPF	Open Shortest Path

PaaS	Platform as a Service
PBA	Population-Based Algorithm
PSO	Particle Swarm Optimization Algorithm
QoS	Quality of Service
SaaS	Software as a Service
SDN	Software Defined Network
SSA	Single Solution Algorithm
S-ICM	SDN-Based Improved Cloud Management
TE	Traffic Engineering

INTRODUCTION

Due to the nature of cloud computing's ability to ensure scalability, flexibility, and cost-cutting, it has revolutionized how businesses deploy services. Consumers today expect the availability of service and information always on, seven days a week due to the deployment of cloud environments. Ensuring that cloud service providers deliver reliable quality service has become necessary as their propositions expand worldwide. While cost reduction and adaptability of the cloud infrastructure were the major concerns of the initial cloud computing research studies, availability, and network resource utilization became critical concerns, particularly in executing real-time services (Sahu & Tiwari, 2012).

At the same time, SDN has attracted a great deal of attention as a promising architecture to improve resource orchestration and network adaptability in cloud settings. SDN increases the ability to centralize control and optimize resource usage by separating the network into the control and data planes (Neghabi, Jafari Navimipour, Hosseinzadeh & Rezaee, 2018). However, traditional load balance systems have a lot of struggles in adapting the strategies to the need and availability of the new network and new resources as the cloud network grows more in terms of size and structure. The mentioned issues contribute to service quality, efficiency, and cost issues which need to be solved (Belgaum, Musa, Alam & Su'ud, 2020).

Problem statements

1. Internet Traffic and Cloud Service Requirement:

   - Growing Demand: Pressure on the fact that an increased number of companies and individuals require cloud services, leaning on cloud infrastructure. This increased demand is a challenge to conventional systems as these usually are not elastic in that they can quickly scale resources for real-time demands.

- Dynamic Traffic Patterns: Cloud services involve creating traffic that constantly fluctuates and is therefore not easily predictable, meaning the traffic needs to be managed. More over, traditional networks may encounter some problems, such as potential bottleneck and the performance decreases when such fluctuation appears.

2. Resource Allocation Inefficiencies:

- Limitations of Traditional Load Balancing: Conventional approaches to load balancing fail to implement well to variability in demand or network characteristics. While the structures may lead to the strategic allocation of resources in that some resources are used more frequently than others, the allocation can cause imbalance in resource utilisation.
- Energy and Cost Concerns: Hire new data centers for cloud services and the energy used, as well as other costs, rises. These problems are aggravated by inefficient resource management, and one must have a technique that will solve the problem without compromising on energy or performance.

3. SDN as a Solution and Its Challenges:

- Dealing with dynamic change in network topology: SDN has centralized control, via an architecture that differentiates between the control plane and the data plane, thus can better utilize resources. Still, the dynamic management of resources in the context of the SDN framework is challenging since a majority of existing solutions do not support real-time change. The existing standard of SDN frameworks requires improvement when it comes to dealing with dynamic change in network topology.

Summing up, cloud service providers' experience includes the following: network congestion and resource allocation, utilization of resources, and load balancing. These challenges may lead to performance bottlenecks, reduced Quality of Service (QoS), and higher operating costs as demand for cloud services escalates. It is often the case that traditional network management approaches do not meet the dynamic adaptability requirements for resource allocation in situations where workload distribution varies significantly in real-time. Additionally, energy consumption is growing as a significant issue due to the high power demand of vast data center establishments (Abu Sharkh, Jammal, Shami & Ouda, 2013). It is relevant to improve and optimize the routing, energy consumption, and load distribution in the cloud networks to increase service availability and reduce costs.

Objectives

Our goal in this study is to develop a framework that optimizes network resource management dynamically in a cloud environment by using SDN. Through efficient load-balancing and resource allocation, this framework will assist us in minimizing response time delays.

Our proposed method will be able to dynamically respond to real-time changes in the load on the network and available resources respectively, something which conventional approaches cannot manage perfectly.

We will increase the load balancing and minimize the error resolution time through efficient resource allocation. The goal is to easily meet the growing demand for cloud services.

By providing an adaptable method of resource management that can be used with new technologies like IoT and 5G, this research advances the domains of SDN and cloud computing.

More accurately and quickly adjust to demand changes than with traditional SDN management techniques.

Methodology

This study developed a hybrid method for dynamically managing cloud resources in SDN infrastructures that combines the Grey Wolf Optimizer (GWO) algorithm with the Fuzzy Neural Network (FNN).

The GWO algorithm and FNN were combined because they address the difficulties of dynamic resource management in SDN-based cloud systems by utilizing the advantages of both approaches.

While GWO effectively looks for the best answers by following the social hierarchy and hunting habits of wolves, the FNN algorithm offers flexibility and manages decision-making uncertainty.

Optimizing resource management improves load balancing and minimizes error resolution time, effectively addressing the limitations of conventional optimization techniques. This advancement facilitates real-time network management.

The methodology includes:

- Simulating network conditions in a cloud environment with SDN by using Matlab.
- Implementing and testing a hybrid algorithm to determine the best route for traffic flow.
- Evaluating the algorithm's performance in terms of load balancing, resource utilization, and problem resolution time. The hybrid method seeks to improve dynamic resource allocation and reduce congestion, resulting in more efficient network performance.

Research Suggestion and Innovation

The novelty of this research is in the design of a new hybrid technique of FNN combined with Grey Wolf Optimizer for load balancing and efficient resource utilization in SDN-based cloud computing systems. While traditional approaches can be fixed or only respond to the current state, this algorithm lets its resources adapt according to the network environment, making the optimization of resources in real-time (Xingjun, Zhiwei, Hongping & Mohammed, 2020) Therefore, the findings of this research can be useful for enhancing knowledge within cloud computing, SDN, and intelligent network management domains and can provide customizable solutions for current cloud infrastructures (Spindler, Reissmann & Rieger, 2014).

This thesis is organized into five chapters:

Chapter One provides the background concepts essential for understanding the research, including foundational topics in network resource management, cloud computing, and Software-Defined Networking (SDN).

Chapter Two presents a comprehensive literature review, examining existing studies related to SDN, resource management, and optimization algorithms.

Chapter Three details the methodology employed, explaining the design and implementation of the proposed hybrid fuzzy neural and Grey Wolf Optimization algorithms within the SDN framework. This chapter elaborates on the simulation processes and the criteria used to evaluate the performance of the proposed solutions.

Chapter Four presents the results of the simulations and their analysis, comparing the performance of the proposed methods against traditional approaches in terms of load balancing, resource utilization.

Finally, Chapter Five concludes the thesis by summarizing the key findings, discussing their implications for future research, and suggesting avenues for further exploration in the fields of cloud computing and intelligent network management.

Through this structure, this thesis aims to contribute to the ongoing discourse on enhancing resource management in cloud environments, addressing challenges related to efficiency in modern network infrastructures.

CHAPTER 1

BACKGROUND

This chapter covers the fundamentals of Network Traffic, the cloud computing concept, SDN, Fuzzy Neural and Neuro-Fuzzy Networks, and Grey Wolf Optimizer Algorithms. Following a broad discussion of artificial intelligence and its algorithms, as well as the algorithm utilized in the study's cloud-based SDN load balancing, this section also discusses the difficulties associated with cloud SDN.

1.1 Network Traffic

Traffic Engineering (TE) is an important mechanism for optimizing the performance of data networks by dynamically analyzing, predicting, and adjusting the behavior of transmitted data. This method has been widely used in past and present data networks, such as ATM and IP/MPLS networks. However, these past and present network models and their related TE solutions are not suitable for next-generation network models and their management for two main reasons (Akyildiz, Lee, Wang, Luo & Chou, 2016).

First, today's internet applications require a network infrastructure that can react in real-time and scale to handle high traffic volumes. This infrastructure must be able to classify different types of traffic from various applications and provide appropriate and specific services for each type of traffic within a very short time frame (Gorlatch, Humernbrum & Glinka, 2014).

Second, with the rapidly growing trend in cloud computing and the consequent demand for large-scale data centers, appropriate network management must be able to improve resource utilization for better system performance (Khoshbakht, Tajiki & Akbari, 2016). Therefore, new network architectures and smarter and more efficient TE tools are quickly needed. Traffic Engineering is a crucial mechanism for optimizing the efficiency of data networks through dynamic analysis, prediction, and regulation of data transfer (Quttoum, 2018).

The traffic engineering patterns used in old networks cannot be used in today's new networks for two reasons:

1. Due to the scalability of its architecture, it must be able to classify traffic generated from various applications and provide services in a short time.
2. As data centers and cloud computing grow in popularity, network management has to be able to boost performance and resource efficiency. A new architecture and effective, intelligent traffic engineering technologies are required for the previous reasons.

In software-based networks, the control plane is separated from the data plane, which may include one or more controllers based on the network sites. In both cases, the control layer is responsible for policy-making and enforcement. Additionally, SDN applications are located in the application layer, which includes a set of programming interfaces that connect this layer to the control layer. This setup enables the activation of network services such as routing, traffic engineering, multicasting, security, access control, bandwidth management, QoS assurance, energy consumption, and network management (Shah, Giaccone, Rawat, Rayes & Zhao, 2019)

1.2 Software-Defined Networks (SDN)

SDN is an innovative approach to computer networks that focuses on the structure of the networks. In the current Network, the control and data plane reside in certain physical equipment such as switches and routers where these devices are currently loaded with control responsibilities. Moreover, configuring current networks as well is very consuming and to some extent challenging (Bakshi, 2013).

In SDN, the control plane sits on top of this architecture in a logically centralized device known as a controller where control decisions are taken. The data plane is at the bottom of the network; it consists of physical devices, such as switches but its sole purpose is to forward data. Thus, this centralization of control of the network means that network administrators can control and manage their networks conveniently. Furthermore, SDN introduces network and system management in a single, centralized controller without necessarily being tied to specific hardware with any single company (Kim & Feamster, 2013).

Software-defined networks (SDN) have four special capabilities:

- **Programmable Networks:** The split between the control plane and forwarding plane allows control functionality to be written directly into the network without any interaction with forwarding algorithms. This enables administrators to configure the network in a specific and even make major changes to the network depending on the flow and trend (Li *et al.*, 2017).
- **Centralized Control and Flexible Development:** SDN architecture creates the major control point that provides the general view of the network state and allows making fast changes in network flows and functions. Such centralization makes network management easier since it is possible to make real-time changes in the behavior of the network and coordinate the resources available within the network (He, Varasteh & Kellerer, 2019).
- A**utomated Network Configuration:** SDN provides network administrators the ability to manage network inputs and output via programmable means. SDN, by divorcing the network from specific hardware or software tie-ins, steps up the rate of network configuration which enhances the general competencies of the networks in terms of management, optimization, and resource allocation (Nunes, Mendonca, Nguyen, Obraczka & Turletti, 2014).
- **Based on Open Standards:** The SDN architecture is based on open standards, thus allowing for cohesive network management, as well as for the use of devices made by different vendors (Janz, Ong, Sethuraman & Shukla, 2016). This standardization makes it possible for gear as well as provides convenient integration and a network that runs efficiently across different networks.
- **Dynamic Scalability and Resource Sharing:** The capability that makes SDN unique when compared to traditional networks is that SDN design entails a fine-grained and extensible scale, where nodes can be added or removed depending on the current relative demand, load, or traffic. The provision of this feature provides for improved resource utilization and guarantees efficient utilization of the networks (Haji *et al.*, 2021).
- **Load Balancing:** Since SDN has centralized control of all the hardware components like switches and routers, all these aspects allow load balancing, avoiding cases of congestion of the network. This means better network function, durability, and effectiveness (Neghabi *et al.*, 2018).

Among the benefits of SDN are:

- Equipment concentration in the network
- Configuration and minimization of equipment in the network.
- Allowing overall network programmability to meet specific goals;
- hiding physical elements of network infrastructure from management
- Applying different forms of addressing and packet routing to address the issue of virtualization
- Any application that is developed for the platform by a third party other than Google or Facebook is referred to as a third-party application or third-party software or application for short. These applications in SDN mean the applications that are not developed by SDN itself but developed by other businesses like Cisco, Intel, and others or even on an individual basis (Zhong *et al.*, 2013).

1.2.1 SDN Architecture

The changes in the network are driven by the development of new technology, the emergence of new and diverse service requirements, the growth of traffic demand, and the pressing need to minimize costs. These are the conditions for extensive changes in networks, and the new SDN networks are a specific outcome of the changed network structure standardly formed to meet the above-mentioned needs (Doshi, Nagarajan, Prasanna & Qureshi, 2001).

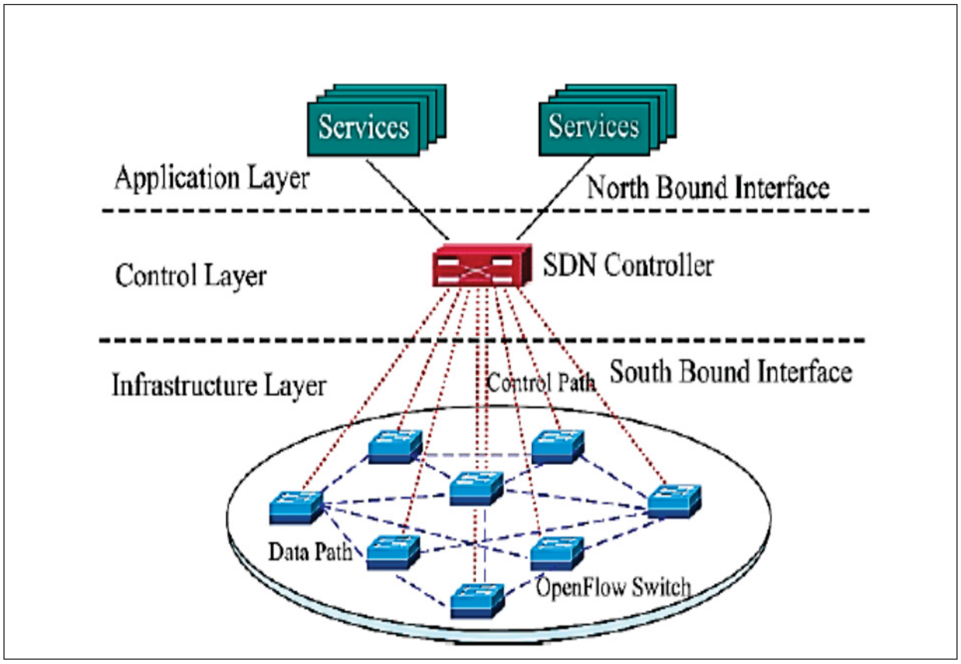

Figure 1.1 Software-based network architecture
adapted from Das *et al.* (2020)

Software-defined networking, or SDN, is a novel networking design that essentially divides the network's control and data planes from one another. Figure 1.1 displays the SDN architecture as described by the Open Networking Foundation (ONF). Three separate layers make up the SDN architecture, as shown in Figure 1.1, and are reachable via an application interface (Hoang, 2015):

1. Application Layer: From the perspective of this layer, the new network appears as a series of logical and unified switches. Commercial applications residing in this layer can include security applications, network virtualization applications, network monitoring, access control management, and others. It also includes commercial applications for end-users and consumers of SDN communication services (Xia, Wen, Foh, Niyato & Xie, 2014).
2. Control Layer: This layer belongs to control functions that regulate the behavior of the network. It is also known as the control plane and consists of a number of programmed software controllers that offer coherent and centralized control utilizing an open Interface to control network behavior. In this layer, there are two API interfaces:

- Southbound Interface: The primary function of this API interface is to link the control layer with the lower layer, the infrastructure layer. Protocols such as OpenFlow and For CES are part of this interface, and their role is to enable software-based network programming. These protocols will be explained in subsequent sections (Lang & Gui, 2021).
- Northbound Interface: This interface is therefore used to provide a connection between the application layer and the control layer. In other words, Northbound enables the application to communicate with the control unit. This layer is a system with a software hierarchy and follows a standard API whose importance is central to the future of SDN. The SDN controller is also located in this layer and is the separated control part of the switch, it provides the user's flexibility in programming the control and data forwarding capabilities. Furthermore, as the traffic flows through the forward and data plane, the controller further inserts modifies, or removes entries in the routing table of the switch (Du, Lee & Kim, 2018).

3. Infrastructure Layer: This layer is the bottom layer of the SDN architecture and is named the data layer. There are physical and Virtual switches located in this layer. Virtual switches have been identified to have arisen from the application of server virtualization technologies used by controllers (Xia *et al.*, 2014). Virtual switches are also important in linking virtual servers to virtual network adapters and in controlling the traffic density and passing them out of the physical network through controllers. In general, this layer is for handling and routing packets through a path decided by the controllers in the control layer.

1.2.2 OpenFlow

OpenFlow is an essential form of exchange between various layers in the Software-Defined Networking (SDN). It enables the SDN controller to forward instructions to the flow tables of network switches via a TLS connection and uses port 6633 as its communication port with the switches. This protocol has remarkable importance in the provision of centralized control of the forwarding table of the network and flow and traffic control (Agborubere & Sanchez-Velazquez, 2017).

The design of OpenFlow is composed of controllers that manage to provide flow instructions to OpenFlow switches to police traffic flows into the desired network path (Figure1.2). This dynamic

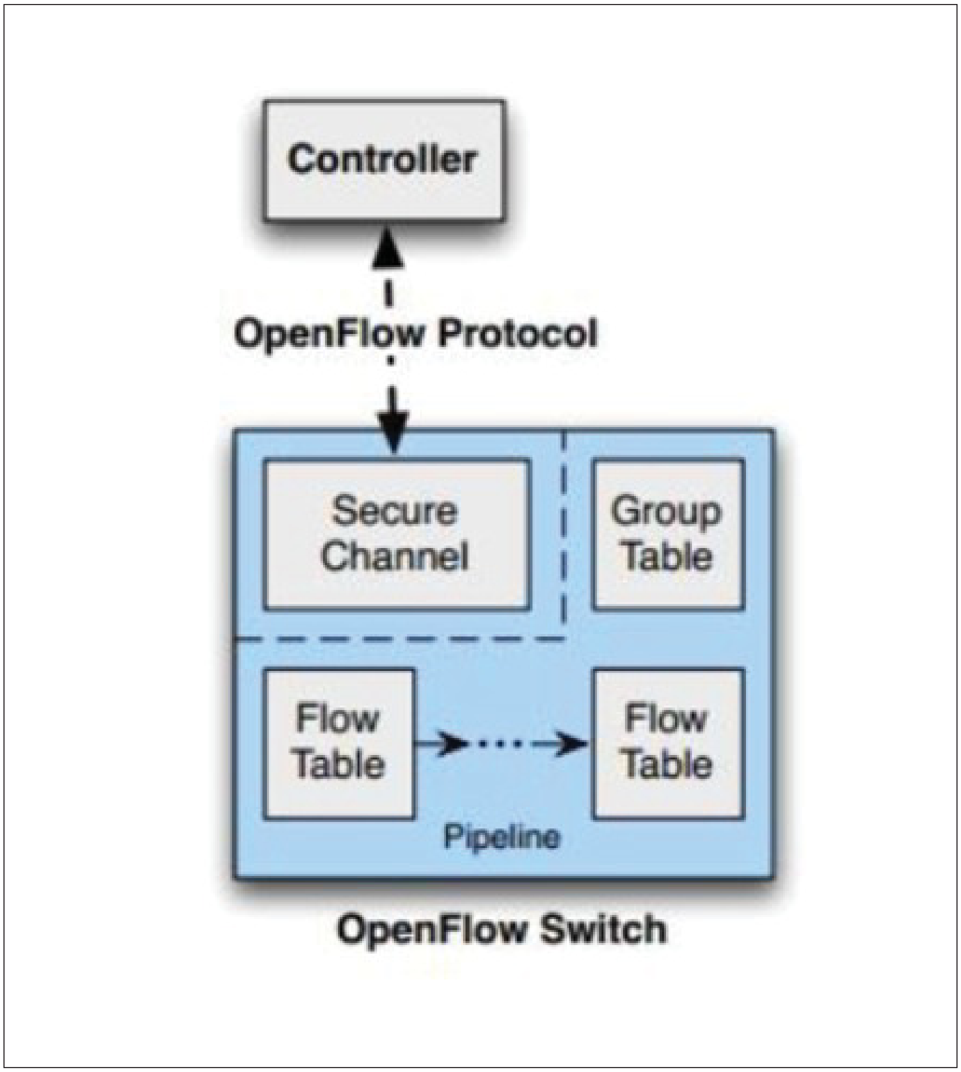

Figure 1.2 Software-based network architecture
adapted from Chen *et al.* (2015)

capability is important for increasing the network performance since it enables the controller to change the network flows according to the traffic load, available resources, and network utilization conditions (Lara, Kolasani & Ramamurthy, 2013).

As such, OpenFlow is critical to the methods used in this study because it provides the technological basis for the emulation of network resource management. The specific protocol is used to show how other entities in an SDN framework can reduce data flow complexity and enable the use of resource controllers to improve load distribution across the network and improve resource utilization. This perfectly corresponds to the goal of effective management of the network resources in cloud platforms by utilizing SDN, which is described in this thesis (Mulla, Raikar, Meghana, Shetti & Madhu, 2019).

1.3 Cloud Computing

Cloud computing is a mechanism for storing and accessing data and programs over the internet. The cloud is just a word used for the internet. Cloud Computing is the use of hardware and software to deliver a service over the internet. An instance of a Cloud Computing provider is Google's Gmail. It's the environment that provides the on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be quickly provisioned and released with minimal management effort or service provider interaction.

The cloud is composed of five compulsory characteristics, three service models, and four deployment models. The well-known cloud service providers are Google, Microsoft, and Amazon) who provides on-demand services to its client's business model. Cloud services are majorly provided in different areas such as business and education however information technology is one of the emerging areas in which cloud computing is utilized fast. The outstanding potential of the cloud is its ability to provide resources such as hardware and software over the Internet. Generally, we can divide the cloud into private, public, community, and hybrid clouds (Mell, 2011).

- **Private Cloud:** The private cloud provides services to an organization.
- **Public Cloud:** The public cloud provides the infrastructure and services to organizations and the public and provides resource sharing to multiple people.
- **Community Cloud:** Community cloud provides services to organizations and the public with similar interests.
- **Hybrid Cloud:** It's the mixture of the private and public cloud.

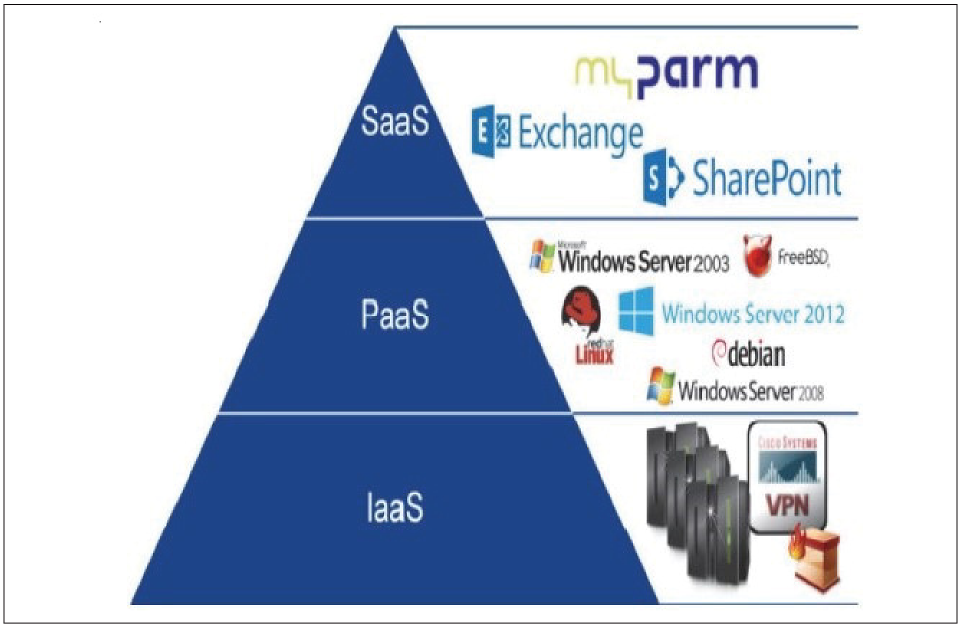

Figure 1.3 Main types of cloud (SaaS Vs PaaS Vs IaaS) adapted from Alsufyani *et al.* (2015)

Cloud computing provides services to the users these are the main models of cloud computing Software as service, plate for as service, and infrastructure as service Linthicum (2016). These service models will be presented here shortly (Figure1.3).

SaaS; Software as a Service. It is a software delivery model that delivers via an interface, such as web browsers. Users have no concerns about cloud infrastructure, operating systems, service platforms, and storage, among other things. In this instance, there is no need to install software on the computers. The SaaS model is currently a robust delivery platform for corporate applications, billing, and management.

PaaS; Platform As A Service, provides high-level integrated environments to make and test deployments of the applications. The platform as services model mainly deals with assistance provided for the development of the application.

IaaS; Infrastructure as services, facilitate the context of processing, storing, networking, and other basic computing resources to users. IaaS users can deploy applications randomly, and applications and operating systems on the infrastructure can scale up and down dynamically. Its information is virtualized flexible and manageable to meet users' requirements.

1.4 Internet of Things (IoT)

A network of physical objects with sensors, actuators, and networking capabilities that allow them to exchange data and communicate online is known as the Internet of Things (IoT). IoT has significantly improved modern life, influencing both personal and business activities. For instance, IoT technology is used in smart homes to provide residents with a better and more convenient living environment (Fereidouni, Fadeitcheva & Zalai, 2023).

1.4.1 Advantages of Using SDN in IoT:

SDN is advantageous for IoT networks by enabling service chaining, improving traffic and bandwidth control, and simplifying operational management. It supports dynamic resource allocation and programmability, which help in managing the increasing number of IoT devices (Manguri & Omer, 2022). SDN also enhances automation and managerial efficiency in data centers for IoT (Tayyaba, Shah, Khan & Ahmed, 2017).

1.4.2 Challenges of Using SDN in IoT:

SDN's unfamiliar architecture, security concerns, scalability issues, and lack of expertise pose challenges for IoT adoption. Organizations reliant on traditional tools struggle with SDN adaptation. Specific security requirements for IoT devices complicate implementation (Tselios, Politis & Kotsopoulos, 2017). The separation of control and data planes can create scalability issues, such as controller bottlenecks, affecting network performance. Additionally, high training costs hinder adoption (Hu, Wang, Gong, Que & Cheng, 2014).

1.4.3 Integration of SDN and IoT:

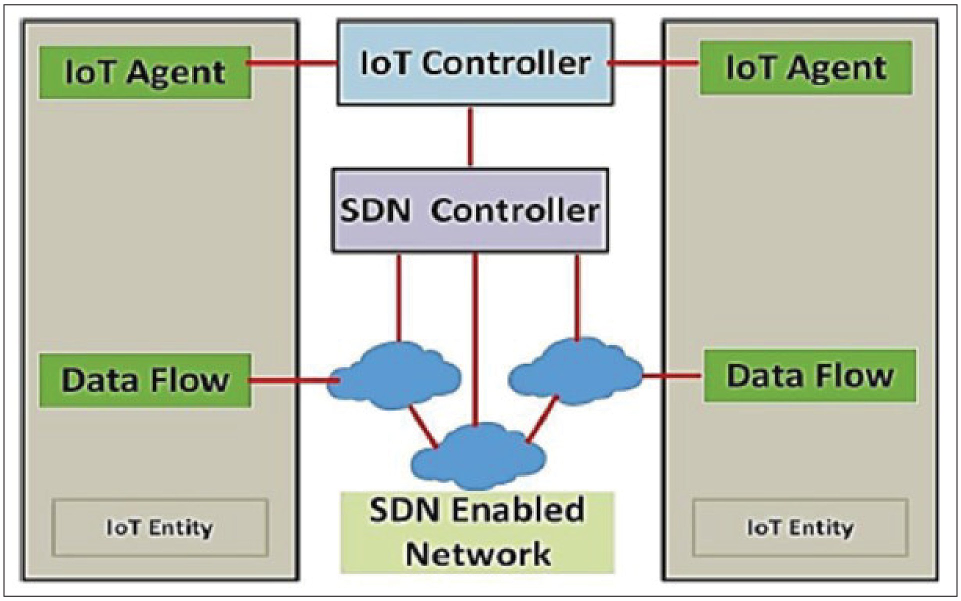

Figure 1.4 Overview of SDN and IoT integration)
adapted from Islam *et al.* (2019)

Integrating SDN with IoT requires assessing IoT elements to determine SDN control needs. The control layer connects to the SDN controller, managing IoT requests and setting up communication paths (Figure1.4). IoT controllers manage connection requests and routing, adjusting forwarding rules to ensure compatibility between protocols. This approach, though initially slower, is efficient, especially with SDN-optimized hardware (Bedhief, Kassar & Aguili, 2018).

1.5 Fuzzy Neural and Neuro-Fuzzy Networks:

Due to the relationship between fuzzy logic and neural networks, there are diverse types of systems. Some complain that its usage in all of these combinations is wrong because some of them have complementary associations with each other and other systems including decision trees, evolutionary facilities, etc could substitute for any of these. That is, neuro-fuzzy means the integration of the technique of artificial neural network and the fuzzy inference system whereby the neural network selects the parameters of the fuzzy system. This means that at the time the neural network defines the parameters of the fuzzy system, it is also a method of setting fuzzy parameters (fuzzy rules or membership functions of fuzzy sets) (Mitiku & Manshahia, 2018). On the other hand, fuzzy neural networks are the networks in which the utilization of fuzzy logic can enhance the functioning of the neural network. In this network, fuzzy logic is supplementary and is employed solely to enhance the circumstances of the neural network or to incorporate the principle of uncertainty into it. The following classification explains the interaction between fuzzy logic and neural networks according to this perspective:

- Fuzzy Neural Networks: Fuzzy logic is used to improve the performance or increase the learning capability of the neural network. In these networks, fuzzy rules are added to change the learning rate or to modify the output from a non-fuzzy to a fuzzy state (Yu & Li, 2004).
- Synchronous Neuro-Fuzzy Models: Neural networks and fuzzy systems work together on a single task without influencing each other. Neither is used to determine the parameters of the other. In this model, the neural network is typically used for preprocessing the input or output of the fuzzy system (Yu & Li, 2004).
- Shared Neuro-Fuzzy Models: The neural network is used to determine the parameters of the fuzzy system. These parameters include fuzzy rules, rule weights, and fuzzy sets (Nauck & Kruse, 2020).
- Fuzzy Inference System Based on Neural Networks: Some consider these systems to be part of the shared models. These models are used to expand fuzzy rules (Nauck & Kruse, 2020).
- Hybrid Neuro-Fuzzy Models: In this model, the neural network and fuzzy system are combined within a coordinated structure. This model can be considered as either a neural network with fuzzy parameters or a fuzzy system with distributed learning (Rutkowska, 2001).

1.6 Metaheuristic Algorithms

It is possible to divide metaheuristic algorithms into those which focus on a single solution and those which work with a population of solutions. Single Solution Algorithms (SSA) change a single solution during the search, while Population-Based Algorithms (PBA) deal with a population of solutions during the search.

- Nature-inspired and non-nature-inspired: It should be noted that while a large number of metaheuristic algorithms are based on naturally existing systems, others are not.

- Memory-based and memory-less: Some metaheuristic algorithms do not rely on the information derived throughout the search and in this sense are memoryless. On the other hand, some algorithms such as Tabu Search employ memory which is information acquired in the search process.
- Deterministic and probabilistic: Deterministic metaheuristic algorithms, such as Tabu Search, make exact, or deterministic decisions to solve the problem. Whereas, there exist probabilistic metaheuristic algorithms such as Simulated Annealing in which the probabilistic rules are employed during the search stage (Rajabi Moshtaghi, Toloie Eshlaghy & Motadel, 2021).

1.6.1 Grey Wolf Optimizer Algorithm

The Grey Wolf Optimizer (GWO) algorithm was presented by Seyed Ali Mirjalili in 2014 in his article with the same title in the journal Advances in Engineering Software published by Elsevier. The Grey Wolf Optimizer is a metaheuristic algorithm based on the grey wolves' structure and hunting technique. GWO is a population-based, nature-inspired, memory-less, and probabilistic metaheuristic algorithm. In the optimization process, alpha, beta, and delta wolves are used. It is assumed that there should be one Alpha wolf that leads this algorithm and coordinates with Beta and Delta wolves, while the rest of the wolves are perceived as obedient ones (Mirjalili, Mirjalili & Lewis, 2014). To design the optimization algorithm, Mirjalili has used the essence of the search and hunting process of the grey wolves. In the mathematical model of this algorithm the best solution is represented by the alpha term (α), the second-best solution by the beta term(β), and the worst solution by the δ term. All the other candidate solutions are identified as omega (ω) and three of the grey wolves are used throughout the search, optimize, and hunt. In the Grey Wolf Optimizer algorithm, the iteration is initiated each time prey is discovered, and the alpha, beta, and delta wolves marshal the omega wolves and encircle the prey. In this algorithm, the position vector of the grey wolves depends on the position vectors of the alpha, beta, and delta wolves. The control parameter also varies; it is a linear parameter that initially has the value '2' and goes down to '0' at a later iteration (Mirjalili *et al.*, 2014).

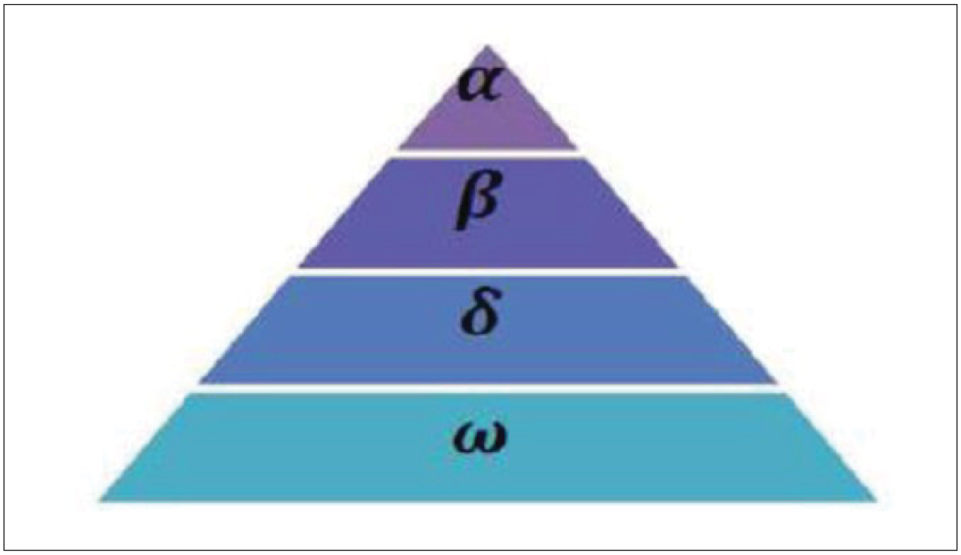

Figure 1.5 The social hierarchy of gray wolves
adapted from Sharmila & Indra Gandhi (2019)

1.6.2 Grey Wolf Optimizer Algorithm

In the social hierarchy diagram highlighted in Figure1.5, the alpha wolf is understood to be the leader of the given group and is the first to make decisions concerning the hunting timing, the place to sleep, wake time, etc. The alpha's decisions are imposed on the rest of the group members; however, the level of democracy is present too. The second step in the pack's hierarchy is given to the beta wolves which helps the alpha in the decision-making and any other activity within the group. These wolves are the best bets to replace the alpha whenever the alphas are too old or when the alphas die. The most subordinate position in the Pack is occupied by the omega wolves who are scapegoats for the entire Pack. They have to bow before all the other wolves and are the last to be given food. The omegas which are deemed to be of little significance within the pack, it was noted that the removal of the same creates inner struggles and problems in the pack. Wolves not included in this hierarchy are known as the delta wolves. Although delta wolves are next to alpha and beta they enjoy a higher ranking than omega (Mirjalili, Hashim & Sardroudi, 2012).

Most engineering optimization problems are often very difficult and many applications have to grapple with these complications. In these problems, the size of the search space increases exponentially with the size of the problem, so classical optimization methods are not effective in solving these problems. Hence, over the years, a considerable number of metaheuristic algorithms have been proposed to handle such problems. Scientists have shown the efficiency of applying metaheuristic algorithms to a large number of difficult tasks, including scheduling problems, data clustering, image and video processing, tuning the parameters of neural networks, and pattern recognition.

When developing the mathematical model for the Grey Wolf Optimizer algorithm that is based on the social behavior of the wolves, the best solution is called the alpha. After that, it is anticlockwise, then the clockwise solution is called a beta solution, and the rest of the solution is considered as an omega solution in conjunction with the delta solution. In the Grey Wolf Optimizer algorithm, the hunting, or optimization is done through the help of alpha, beta, and delta and the omega wolves perform these three (Hansen, Müller & Koumoutsakos, 2003).

1.6.3 Analysis of Grey Wolves' Hunting Behavior

Grey wolves can detect the position of prey and surround it, typically under the leadership and guidance of the alpha wolf. Occasionally, delta wolves also participate in the hunt. However, in an abstract search space, we have no idea of the optimal prey location. Therefore, the mathematical simulation of grey wolves' hunting behavior is based on selecting the alpha as the best candidate solution, with beta and delta as the second and third best solutions, respectively. In addition to the social hierarchy of wolves, group hunting is another notable social behavior of grey wolves.

The main stages of grey wolves' hunting are as follows (Figure1.6):

- Tracking, chasing, and approaching the prey (Figure A).
- Encircling and exhausting the prey until it stops (Figures B and C).
- Attacking the prey (Figure E).

Updating the Positions of Grey Wolves:

At this stage, after the analysis of the hunting behavior which is equivalent to optimization in this case, we save the first solution, which is the best solution obtained so far, and compel the other search agents including omega, to force their positions to be updated depending on the position of

Figure 1.6 Hunting behavior of gray wolves
adapted from (Dai & Zhao, 2021)

the other best search agents (Mousavi, Mosavi, Varkonyi-Koczy & Fazekas, 2017).

1.6.4 Mathematical Equations for Modeling and Optimization

In hunting, grey wolves surround and pen their prey and this is as discussed earlier. The following equations are used to model the encircling behavior:

Equations 1- 1 are part of the Grey Wolf Optimization algorithm's mechanism for simulating the social behavior of grey wolves as they hunt for prey (the optimal solution). The algorithm mimics the leadership structure of wolf packs, where the alpha, beta, and delta wolves guide the search process while the omega wolves explore the search space. The coefficients A and C play crucial roles in balancing the exploration of new areas in the solution space and the exploitation of known good solutions. As the iterations progress, the algorithm increasingly focuses on refining solutions rather than exploring new ones. represents the current iteration, D Represents the distance between the position of the prey (optimal solution) and the position of a wolf at time t. A and C are coefficient vectors, XP is the position vector of the prey, and X is the position vector of the grey wolf.

Component a decreases linearly from 2 to 0 over the iterations, and r1 and r2 are random vectors between 0 and 1 (Kalra & Singh, 2015).

$$\begin{aligned}
\vec{D} &= \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \\
\vec{X}(t+1) &= \vec{X}_p(t) - \vec{A} \cdot \vec{D} \\
\vec{A} &= 2\vec{a} \cdot \vec{r}_1 - \vec{a} \\
\vec{C} &= 2 \cdot \vec{r}_2
\end{aligned} \tag{1.1}$$

Equation 1.1: Grey Wolf Optimizer (GWO) algorithm equations (Kaveh & Talatahari, 2010).

Equation 1.1 is part of the Grey Wolf Optimization algorithm's mechanism for simulating the social behavior of Grey wolves as they hunt for prey (the optimal solution). The algorithm mimics the leadership structure of wolf packs, where the alpha, beta, and delta wolves guide the search process while the omega wolves explore the search space. The coefficients A and C play crucial roles in balancing the exploration of new areas in the solution space and the exploitation of known good solutions. As the iterations progress, the algorithm increasingly focuses on refining solutions rather than exploring new ones. represents the current iteration, D Represents the distance between the position of the prey (optimal solution) and the position of a wolf at time t. A and C are coefficient vectors, XP is the position vector of the prey, and X is the position vector of the Grey wolf. Component a decreases linearly from 2 to 0 over the iterations, and r1 and r2 are random vectors between 0 and 1 (Kaveh & Talatahari, 2010).

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a literature review of Software Defined Networking (SDN), AI algorithms, Internet of Things (IoT), and load-balancing methodologies is provided. As a result, this review aims to explain such aspects as the theoretical background of SDN, benefits that can be derived from it in dynamic networks, how it works with IoT, as well as the most critical and crucial challenges connected with the integration such as scalability, security, and control. As a preparation for the subsequent discussion of the algorithms and frameworks, the present literature review aims to provide a historical overview of SDN and its application to cope with the rising IoT and cloud challenges. It also presents diverse techniques in load balancing in cloud computing, static and dynamic, and the way that SDN can assist in cloud performance in operational dynamic systems.

2.2 SDN for Cloud-Based Network Optimization and IoT Integration

(Kang & Choo, 2018) depicted a study with the title "An SDN-Enhanced Load Balancing Technique in a Cloud System" which was published in the year 2018. Most of the web services and sites are hosted on different types of clouds while for achieving certain QoS levels in such systems, effective policies for load balancing, selected from multiple clouds are needed. Among the latest solutions furthering research regarding load balancing in cloud data centers, Software-Defined Networks (SDNs) have stood out. SDN is characterized by its two distinct features: introducing the idea of separating the control plane from the data plane and also providing a programmable interface for developing network applications. For this reason, SDN and cloud computing can enhance dependability, provision, and mastery while increasing scalability and being more manageable and controllable. SDN-based cloud is a next-generation purpose-built for the cloud with control in the network using the technology of SDN and delivers NaaS to the cloud computing system. In this paper, they propose an SDN-based Improved Cloud Management (S-ICM) that assigns the network flows in a cloud environment. S-ICM consists of two main components:

coordination, control, kind of disclosure, and data application when it comes to monitoring and decision-making. For monitoring, S-ICM relies on the SDN control messages to sample the state of the network, and for decision-making, the measured end-to-end delay of the packets is used. In order to put S-ICM to par with task allocation across clouds whereby workload is fairly distributed and employing the Honeybee Foraging Algorithm (HFA), measurements were taken. The student's experimental result showed that S-ICM outperformed HFA and RR when system saturation was tested using heavy load scheduling combined with RR task scheduling. Measurements also established if system performance under multiple clouds could be predicted using a plain queuing formula when under an RR scheduling policy to justify the theoretical approach.

(Mohiuddin, Khan & Engelbrecht, 2016) published a paper known as Fuzzy Particle Swarm Optimization Algorithms for the First Open Shortest Path First (OSPF) Weight Setting Problem. There is nothing better than the Open Shortest Path First (OSPF) method when it comes to routing packets from a source node to a destination node. In the protocol, priority (or cost) is given to the links that are connected to the network. These weights are used in order to find the shortest paths between all sources and all the destination nodes. This activity of assigning weights to the links belongs to the class of NP-hard problems. The problem that OSPF has with weight setting is an issue that ought to be solved to arrive at the most efficient routes in a network. In this paper, the above problem is defined as a multi-objective optimization issue. These include the maximum utilization or the highest possible usage of links, the number of links experiencing congestion and finally the number of inactive links. These criteria possess conflicting characteristics, which explain why they are processed iteratively using fuzzy logic and optimized for the development of a scalar cost function. This paper presented a fuzzy Particle Swarm Optimization (FPSO) algorithm and a variant of the proposed PSO that is known as the Fuzzy Evolutionary PSO (FEPSO). Simulated evolution with FPSO is a combination of exploratory and FPSO elements in the human engineering and simulation process.

Experiments were conducted using 12 test cases reported in the literature. These test cases included networks with 50 and 100 nodes, with the number of arcs ranging from 148 to 503. The experimental results for various FPSO parameters were obtained and analyzed. The results also showed that

FEPSO outperformed FPSO in terms of solution quality, achieving improvements between 7% and 31%. Additionally, comparisons between FEPSO and various other algorithms, such as Pareto Dominant PSO, Weighted Aggregation PSO, NSGA-II, Simulated Annealing, and Simulated Annealing Algorithms, showed that FEPSO outperformed all of them by achieving the best results for two or all three objectives.

(Xingjun *et al.*, 2020) wrote a paper with the title: 'A Novel Fuzzy-Based Approach for Load Balancing in Cloud-Based IoT Using the Grey Wolf Optimizer Algorithm'. Cloud computing offers greater availability, modularity, and versatility in the computing period for most real-life uses. The Internet of Things (IoT) is a new generation of technology that makes use of devices and objects to deliver the required user service. Because of the growing amount of data and information in the Internet of Things, the data is often managed through cloud computing which is popularly referred to as cloud-based IoT. Considering the number of requirements and versatility of services, one of the essential difficulties, that cloud-based IoT does have, is the load distribution. Since load balancing is known as the NP-hard problem in the context of the heterogeneous environment, this work proposes a new framework. The formulated response time reduction method builds on the concept of a typical Grey Wolf Optimizer algorithm. For the scenario where the CloudSim simulation environment has been used, the response time was higher than that of the HBB-LB and EBCA-LB algorithms but lower than that of the proposed CloudSim algorithms. Also, the load imbalance was enhanced compared to TSLBACO and HJSA.

(Pan, Ren & Tang, 2015) conducted a study titled "Research on Heuristic-Based Load Balancing Algorithms in Cloud Computing": Smart Data Analysis and Smart Applications Analysis have been described in detail in the following section. The Bee Algorithm came to life because of the load balancing problem. With the help of this method, load balancing will be used with the example of the bee foraging intensity while trying to get through virtual machines most intelligently and use power Carefully. Like the Bee Algorithm, the tasks taken away from a loaded virtual machine are considered as bees while the virtual machines are analogous to the food sources. If a virtual machine is overloaded, the process resembles honey collection from the food source and the task can be returned to a less loaded machine – the latter is the indication of a new food source discovery. Although it somewhat increases the overall wait time of the tasks, this algorithm greatly reduces the average wait time and in turn, optimizes the time needed to execute all the tasks by estimating task importance.

(Kalra & Singh, 2015) published an article with the title “A Review of Metaheuristic Scheduling Techniques in Cloud Computing” in 2015. The powerful optimization of load balancing problems in cloud-based IoT resulted in the development of the Firefly Algorithm and Weighted Round Robin. All in all, the present paper aims to establish a dependable and very efficient system in the best use of available resources. Therefore, load balancing in cloud computing for data loading onto nodes uses a combination of two types of algorithms: dynamic (LOAD balancing with adaptive layer) and static (ROUND ROBIN). The results demonstrate that the proposed method consistently outperforms other methods in achieving higher resource utilization, faster processing times, and overall lower response time.

(Katyal & Mishra, 2014) In the paper “A Comparative Review of Load Balancing Algorithms in Cloud Computing Environments" presented a new approach derived from ACO to design a new load-balancing method. Globally, speaking random search method was used in the Ant Colony Optimization approach which was similar to the behavior of actual ants to look for food. In the proposed method, artificial ants are dispersed to search out a major loaded virtual machine and allocate tasks to it to control load balance. The results have shown that the reduction in the response time was achieved with the use of the proposed method in comparison with the Genetic Algorithm (GA) and the First Come First Served (FCFS) algorithm. But fault tolerance is not incorporated, task transfer time, throughput, and energy consumption are not taken into account, and the priority of all tasks is assumed to be equal.

(Javanmardi *et al.*, 2014) presented a "Hybrid Task Scheduling Algorithm for Cloud Computing Environments." With the help of a Genetic Algorithm and Fuzzy Theory, they proposed a hybrid task scheduling approach that considers system load balancing to reduce overall execution time and cost. The main objective of this research is to allocate jobs to resources by considering the MIPS of the VM and job length. The new algorithm assigns tasks to resources by considering task length and resource capacity. The experimental results demonstrated the efficiency of the proposed method in terms of execution time, execution cost, and the average degree of imbalance (DI).

(Nayak, Nanda, Nayak, Naik & Behera, 2014) suggested a research work entitled “A Model of Load Balancing Using Firefly Algorithm”. A hybrid fuzzy load-balancing algorithm was also employed to control the use of virtual machines optimally. The idea behind the proposed algorithm is to minimize total time, cost, and response time by optimizing the energy use of the system. Cooperative Firefly also helps to distribute the load evenly over the cloud resources to act as an agent. The output of the data is managed by fuzzy rules that then regulate the hybrid algorithm. Evaluation of the obtained experimental results is as follows: it is observed that the proposed fuzzy-based hybrid optimization approach yields improved results than other metaheuristic algorithms.

(Azodolmolky, Wieder & Yahyapour, 2013) presented a study titled "Cloud Computing Networks: Key Issues of Innovations Contemplation: Challenges and Opportunities for Innovations. Cloud computing is an implementation of the utility computing concept. This paper describes the architecture of IaaS and the major issues under consideration here are virtual networks and cloud networking. IaaS has the added benefit of being more adaptable; customers pay an amount per a certain amount of computation, storage, and time needed. It is worth noting some of the challenges arising in current cloud networks like the poor performance of applications when migrated from on-premise to cloud facilities, difficulty in bringing devices like deep packet inspection, intrusion detection systems, and firewalls flexibly into the cloud, or complications arising out of policy or topology dependence and the like. A conventional three-level DC network structure has the TOR layer which interconnects the servers in a rack, the aggregation layer, and the core layer which hosts the connectivity towards the Internet edge. This multilayered architecture makes it quite challenging to define the L2 scopes, L2 boundaries of domains and networks, L3 transport policies, and vendor-specific networking equipment at each layer. Applications should be expected to work right out of the box just like regarding the IP addresses and failover mechanisms that rely on networking. Network devices and servers, such as the hypervisors, are naturally associated with a fixed physical infrastructure layout which introduces a location dependence constraint by default. Besides CPS, SDN architecture presents and offers a set of APIs that facilitate the application of basic network services in the network.

(Zhao *et al.*, 2020) In a paper entitled “Energy-Efficient Load Balancing in Data Center Networks with SDN,” Zhao and his colleagues proposed energy-efficient load-balancing approaches for data center networks with SDN. According to the existing knowledge, the regulation of energy consumption is a crucial aspect of today’s data centers that is overshadowed by SDNs’ prevalence and the rising need for bandwidth-intensive applications. The authors presented a new load-balancing approach that aims at not only distributing the load but also minimizing the energy consumption while utilizing SDN peculiarities like the division of the control and data planes and the enhanced programmability of the network. The work presented contributes to the field of SDN-based cloud systems by proposing an efficient method of load distribution that offers better energy efficiency when compared to traditional methods, while maintaining load balancing.

(Belgaum *et al.*, 2020) conducted a comprehensive survey titled "A Comprehensive Survey on SDN: This paper provides an understanding of Security Perspective: Software-Defined Networking (SDN). Security concerns of SDN are presented in this study in an elaborate manner focusing on concerns including data integrity, confidentiality, and availability. Despite the fact that this work concentrates on security problems instead of load balancing and resource management, the presence of these insights is essential for the correct and secure application of SDN in cloud services. An appreciation of these security aspects is key to the designing of efficient SDN-enabled systems that would meet future security challenges in contemporary cloud networks.

(Goudarzi, Anisi, Ahmadi & Musavian, 2020) In the paper titled “A Novel Hybrid Approach for Load Balancing in SDN-Enabled Cloud Environments,” proposed a novel load-balancing approach that leverages heuristic algorithms and machine learning tools. The need to find more effective solutions for load distribution in growing and more complex cloud systems is what the research aims to fill. The presented idea involves the combination of heuristic methods and machine learning to enhance load distribution, decrease latency, and optimize the overall network performance. This work is highly relevant to load balancing and resource management in the context of the SDN-based cloud environment where the primary focus is on increased efficiency and cost optimization.

(Kurroliya, Mohanty, Kanodia & Sahoo, 2020) proposed a GWO called Grey Wolf Aware Energy-saving and Load-balancing in Software Defined Networks Considering Real-Time Traffic in 2020. Energy exhaustion, a crucial issue in current SDN implementation as applications with high bandwidth demands increase is well explained and discussed in the study. This proposed optimized GWO algorithm where experimented provides an overall enhancement in energy efficiency parameters without a large compromise on load-balancing capabilities, especially in the real-time traffic conditions. This research has implications for future work focused on achieving the best resource allocation schemes in SDN especially when deployed on the cloud where power consumption and load balancing are of essential importance.

(Belkadi, Vulpe, Laaziz & Halunga, 2023) In the article "ML-Based Traffic Classification in an SDN-Enabled Cloud Environment," focused on traffic classification through the integration of machine learning algorithms in a software-defined network (SDN) cloud environment. The work fits into the emerging scenarios of the establishment of Internet traffic and conflict resolution in cloud network sets. This research applies different algorithms like Naïve Bayes, SVM, RF, and C4.5 in the classification of network traffic that with high precision is useful in maintaining QoS as well as securing networks. The combination of ML-based traffic classification with SDN offers an effective means of controlling clouds that will be useful in the development of enhanced SDN and cloud technologies, which qualify this paper as a valuable addition to the existing knowledge.

2.3 Summary

The background section is organized by defining what Software-Defined Networking is, the structure of SDN, and how it is better compared to conventional network architectures. SDN splits the combination of control and data planes so that they can be rearranged in a much more flexible, programmatic, and scalable manner. All these features make SDN especially beneficial in cloud systems where resources have to be controlled, and changes have to be made promptly. The combination of SDN with IoT is presented as a major innovation that provides centralized management control and self-organizing traffic management, which plays a most important role particularly if addressing the issues of numerousness and dynamics of IoT networks.

In addition, the review of different load-balancing algorithms in cloud structures pays attention to the static and dynamic types. Dynamic load balancing although more complicated is demonstrated to be effective in offering better performance and tolerance to network faults because it considers real conditions in the network. The review also includes some of the major open problems in SDN like the controller bottleneck issue the scalability issue and the deficiency in security. Furthermore, the chapter discusses OpenFlow as the central protocol for SDN and describes its networking structure and flow-handling mechanisms.

As this chapter draws to a close, several research sections are explored with a focus on the shortcomings in the literature primarily in the area of dynamic load balancing methods and resource allocation for SDN-based cloud environments. These form the premises from which the proposed study is derived to address some of these challenges by designing a new approach based on fuzzy logic and optimization for load balancing and networks.

CHAPTER 3

DESIGN AND IMPLEMENTATIONS

3.1 Implementation of the Hybrid Neural Fuzzy and Grey Wolf Algorithm for Optimal Network Resource Management in a Cloud Environment

In this research, a metaheuristic algorithm is used for classification. In SDN connected to the cloud, the final classifier model is built based on the features that had the best performance, and the results are reported. A neural network-based classifier is used for data classification in this study.
The proposed algorithm combines the strengths of the Fuzzy Neural Network (FNN) and Grey Wolf Optimizer (GWO) to achieve optimal resource management in Software-Defined Networks (SDN) connected to cloud environments.

Fuzzy Neural Network (FNN)

The Fuzzy Neural Network (FNN) integrates fuzzy logic and neural networks to effectively manage uncertainty and learn from data. It processes inputs using fuzzy rules and dynamically adjusts weights during training. The key features of the FNN algorithm in this study are as follows:
We analyzed the following input features: node distance to resource nodes, CPU utilization, and bandwidth usage.
The outputs of FNN include fuzzy categorization of distance (e.g., Low, Medium, High) and fuzzy classification of load (e.g., Low usage, Moderate usage, Overloaded).

Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) algorithm complements FNN by using the outputs of FNN along with resource constraints to determine optimal resource allocation decisions. The key features of the GWO algorithm are as follows: Input features for GWO include FNN fuzzy outputs (categories or membership values) and actual capacity metrics of resource nodes (e.g., CPU Capacity).

The outputs of GWO include optimal resource selection or location, as well as predicted improvements in error resolution time and load balancing.

Process Overview

In this process, the FNN first classifies and evaluates resource loads and distances, while the GWO utilizes this information to allocate resources efficiently. The integration of these algorithms ensures:

- Better decision-making under uncertainty
- Improved optimization
- Effective resource allocation in SDN-based cloud environments

The complexity of optimization issues has increased to the point where it is nearly impossible to identify the global optimal solution. Additionally, because computation is costly and time-consuming, convergence is slow and operations are often complex.

The following flowchart in Figure 3.1 shows the algorithms that have been added to the enhanced hybrid neural fuzzy and Grey Wolf algorithm.

As far as the feature engineering section, proposed in the system, the applied is the improved system of fuzzy neural networks. First of all, the studied dataset is introduced to the improved system. The new feature is obtained by adding the result of the prediction from the improved fuzzy neural system to the original dataset. In the next stage, the adopted Grey Wolf Optimization algorithm is used to identify the appropriate features to be used. Combining all the iterations of this algorithm, the final result shows the available feature to be the best in the current dataset.

The method for learning the objective function of the Grey Wolf Optimization algorithm is based on the usage of an artificial neural network. In other words, the position of each Grey wolf corresponds with a given feature or characteristic. From these features, the relevant neural network is subsequently produced. Specifically, the output of the neural network which is the position of each Grey wolf is employed as fitness in the objective part of the Grey Wolf Optimization algorithm. The best feature solution that solves the recommendation system problem is highlighted after performing the Grey

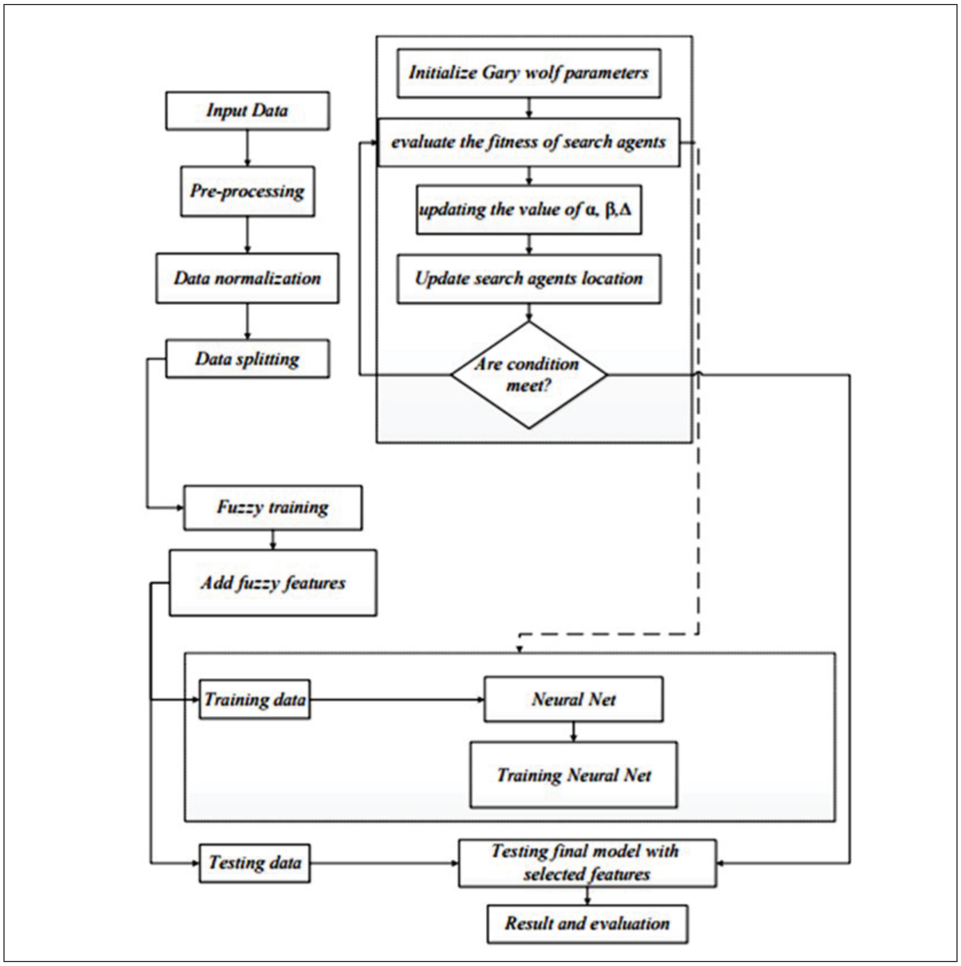

Figure 3.1 Flowchart of the Improved Hybrid Neural Fuzzy and Grey Wolf Algorithm

Wolf Optimization algorithm. Finally, a demonstration and comparison of the suggested system with the specified features will be performed.

3.2 Optimal Network Resource Management and Resource Allocation

After the tasks are queued and before resource allocation, a threshold must be determined. To set the threshold, tasks are initially sorted based on their execution time. The execution time in the middle is selected as the threshold. The tasks at the beginning of the queue are selected according to the number of available resources. For example, the first processor is allocated to the first ready task, the second processor to the second ready task in the queue, etc. The proposed fuzzy neural algorithm has been implemented using MATLAB software.

Table 3.1 Optimal Network Resource Management and Resource Allocation

Parameter related to activities and resources	Range & values
Resource Load	10% - 30%
Resource Bandwidth (Mbps)	150 - 350
Resource of CPU Speed (MHz)	2500 - 3500

Table3.1 outlines key parameters related to network activities and resource allocation in our system. It includes factors such as CPU speed, resource load, and bandwidth, which are crucial for optimizing system performance under varying conditions. The specified ranges provide flexibility to adapt to different power demands and network loads, ensuring efficient resource management.
In this study, we utilized a dataset comprising 1,000 records. Of these, 750 records were allocated for training the algorithm, while the remaining 250 were used for testing its performance.
The selection of hidden layers was predicated on avoiding a performance plateau, in which adding more layers no longer improves outcomes. The ultimate number of hidden layers used two layers, with 64 neurons in the first layer and 32 neurons in the second layer.

3.3 System Environment

The system used to conduct the experiments was configured with an Intel processor operating at 2.80 GHz, 16 GB of physical memory (RAM), and Windows 11 as the operating system.

3.4 Study Area and Cloud Services Positioning in the First Cloud Layer Connected to the Network and Resources in the Second Cloud Laye

In this section, SDN in cloud services is introduced. Specifically, the cloud environment, the number of services, the distance between services, and the type of connections in this space are identified. This is essential for analyzing the first field of optimal network resource management, where services are spaced in separate locations.

In this space, based on the type of services, the distance between services, and the type of current needs, two primary locations are produced from the core of the physical system positioning tree, which is based on the optimization algorithm.

A hybrid network with 16 access points for cloud services in the first cloud layer connected to the network and 4 access points for resources in the second cloud layer is proposed to cover a limited area with dimensions of 20×20 meters and a height of figure 3.2. These points are equally spaced within the SDN cloud services environment, as shown in figure 3.2

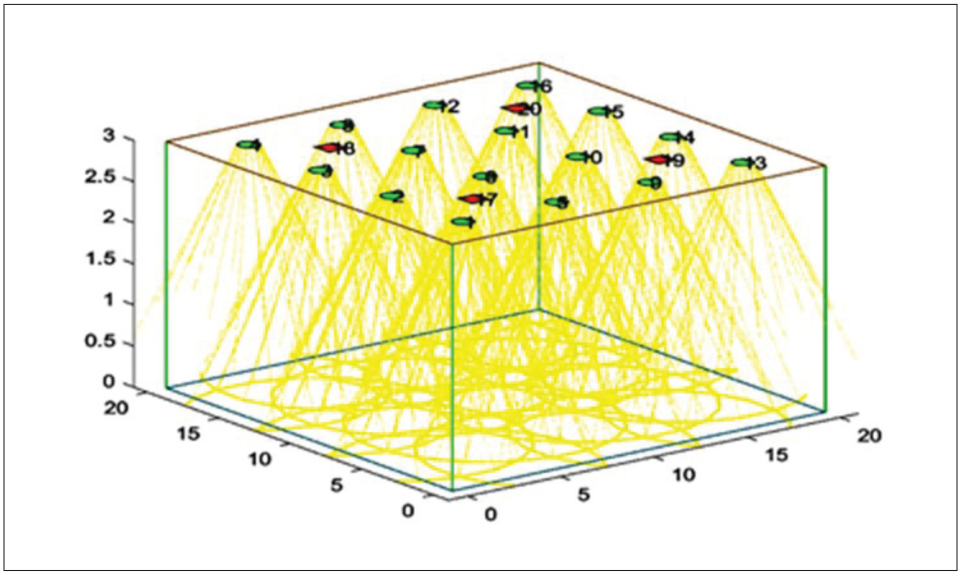

Figure 3.2 Positioning of Cloud Services in the First Cloud Layer Connected to the Network and Resources in the Second Cloud Layer

We should assume that the SDN cloud services environment has dimensions of 20×20 meters.

access points for cloud services in the first cloud layer connected to the network use a unified data transmission protocol from the designated location. The coverage space between adjacent cloud services in the first cloud layer connected to the network is considered a blind spot and is not accounted for in the overall system.

3.5 Chapter Summary

In this chapter, we examined the research methods and demonstrated that this study is applied in terms of its objective and descriptive in terms of its execution. We showed that network resource management applications in the cloud computing environment are addressed using a network resource tree management approach based on the neural fuzzy and Grey Wolf algorithm in the data center using MATLAB software was evaluated.

We proposed utilizing the hybrid neural fuzzy and Grey Wolf algorithm for the process of optimal network resource management. We also explored the implementation of the hybrid neural fuzzy and Grey Wolf algorithm for managing network resources efficiently in a cloud environment. Additionally, we explained how optimal network resource management and resource allocation are performed. Finally, we discussed the implementation of the proposed method.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, we use the intended solution approaches to address the proposed model's solution. First, we explain how the solution string is defined and then describe the method for satisfying constraints. Following this, we outline the management of the physical position of systems connected to the network (SDN) and device spacing based on the hybrid neural fuzzy and Grey Wolf algorithm employed in this study.

The goal of the present research in this section is to minimize the suboptimal placement of systems in a cloud environment connected to the network (SDN) by using optimal network resource management to enhance interactions between services and identify random locations for optimized resource management. To achieve this, services are spaced based on the hybrid neural fuzzy and the Grey wolf optimization algorithm and its coordinates, taking into account the type of connection, geographical location in the cloud environment, data exchange rate, and persistence in communication channels. Device spacing aims to create a phase for physical system placement to reduce suboptimal system placements from outside the cloud-connected network (SDN) to within the cloud-connected network (SDN) and from inside the cloud-connected network (SDN) to other locations. By using device spacing based on the algorithm, services in the cloud-connected network (SDN) are placed in separate locations, creating a degree of homogeneity in relation to one another and heterogeneity in relation to other users of different locations. Thus, in the first phase of zoning, device spacing is performed based on the homogeneity of activities and the location of services.

In the next phase, to exchange information between homogeneous locations of services and even between different locations, optimal network resource management is applied to communication channels based on the types of demands and the nature of the channels. Moreover, for optimizing the system placement, largely based on the neural fuzzy optimization algorithm, each service is assigned an activity location code for each time period in the cloud environment. Consequently, a large number of location codes can be generated for each activity field, which ultimately form the structure of a neural fuzzy optimization algorithm capable of creating a secure environment with minimal intrusion for services. Finally, in the cloud-connected network (SDN), the process of device spacing, spatial relationships, and physical system location are examined, culminating in the creation of a neural fuzzy optimization algorithm to reduce suboptimal system placements in SDN cloud services.

4.2 Simulated Network

In the next phase, to exchange information between homogeneous service locations and even between different locations, optimal network resource management of communication channels is performed based on the type of demands and the type of channels.

Taking the neural fuzzy optimization algorithm into account, the method used for simulation involves the hybrid neural fuzzy and Grey Wolf algorithm in the first cloud layer connected to the network and the resources in the second cloud layer. This is aimed at optimizing service placement (data transmission protocol location, sensitivity factor for each physical resource, and data exchange speed).

In continuation, for the optimal placement of systems, which is primarily based on the management of the neural fuzzy optimization algorithm, each service is assigned an activity location code for each time period in the cloud environment. The SDN-connected protocol for computer network analysis, transient stability analysis of the network, and optimal connection determination for services were implemented using MATLAB software.

In this section based on the scenario that we provided earlier in Chapter 3 (Figure 3.2), the results of implementing the SDN-connected protocol on a test network with 4 geographic locations in the SDN-connected cloud and 16 cloud services in the first cloud layer connected to the network are examined.

With the primary placement of physical resources occurs in cloud services at one node in the first cloud layer connected to the network. After a time of $t_{cl} = 0.2\,t$ seconds, the issue is resolved by opening the transmission line between cloud services by replacing the other node in the first cloud layer connected to the network. To determine the system conditions at the moment when the error is corrected, the system is simulated in the time domain during the fault, and the rotor angle variation curve for the resources in the second cloud layer is illustrated in the figure. During the simulation process, initially, given the number of cloud services in the first cloud layer and resources in the second cloud layer—16 cloud services in the first cloud layer connected to the network and 4 resources in the second cloud layer—we aim to analyze the impact of optimal resource management in the second cloud layer over the time period from 0 to 2 seconds, with the goal of increasing connection speed and the sensitivity factor for each physical resource at its location.

For this purpose, we consider typical conditions in a location with a certain number of data exchange oscillations for a feeder line at a specific location as follows:

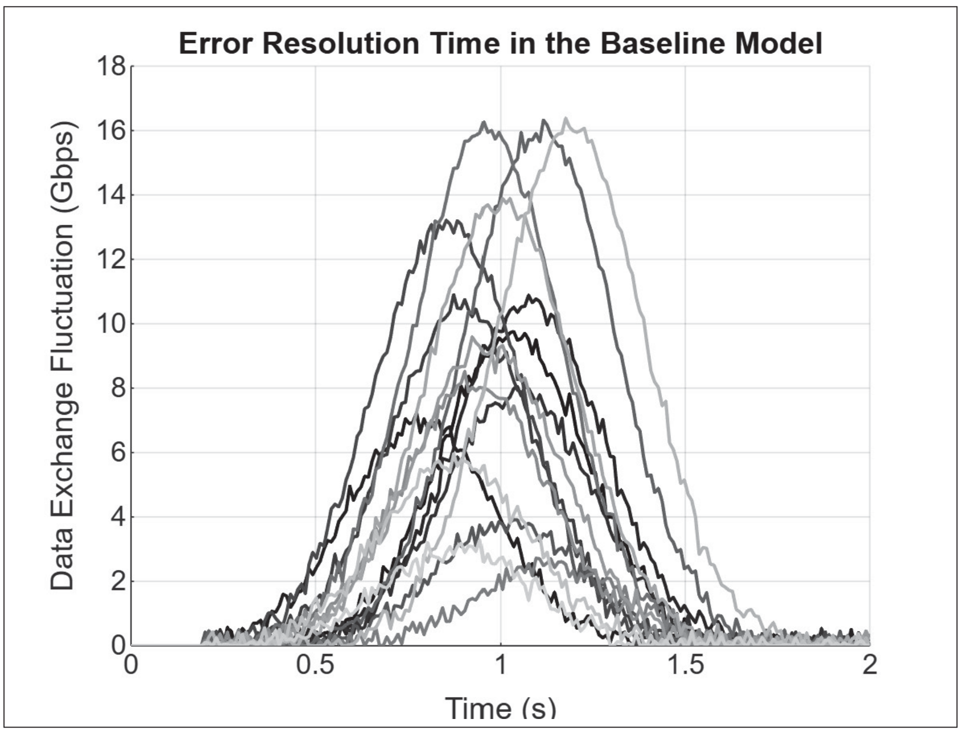

Figure 4.1 Data Exchange Oscillations of Cloud Services after Error

Figure 4.1 shows the baseline model of the system (SDN based on the cloud with the Round Robin algorithm) before applying our proposed algorithm, data exchange through cloud services varies over the simulation time of 0 to 2 seconds, with the X-axis representing time and the Y-axis representing the extent of data exchange fluctuation. Finally, the graph shows the dynamic behavior of the SDN-connected cloud network due to a fault introduced between cloud services at locations of the first cloud layer. The plots of each curve correspond to an individual cloud service and demonstrate the difference in the communication traffic during the fault. The highest points of all the curves represent the degree of oscillation in the exchange of data, a situation that arises when the network is dealing with a fault and trying to reallocate resources. The high fluctuation at the beginning of the period, followed by a slump for about 0.2 seconds when the fault had been corrected indicates the unstable behavior of the system. The decrease in fluctuation stands for enhanced resource management in the second layer of clouds after the implementation of the optimal resource allocation. By allowing a better understanding of the behavior of the network through its analysis, this research demonstrates how the hybrid neural fuzzy and Grey Wolf algorithm can reduce disruptions and regain stability in the network efficiently.

4.3 Distribution of Cloud Services Across Layers

In this section, to establish a communication space for services, we used a distribution method based on location theory to create a connection space for the services. In other words, we applied collaborative location theory to partition the SDN in cloud services, where the cloud services are present. Given the movement of cloud services within the network space (SDN in the cloud services under study), we considered each part of the space in such a way that a resource in the second cloud layer is connected to several cloud services in the first cloud layer connected to the network. This ensures that an appropriate and suitable data transmission protocol can be determined for the cloud services at that specific time and location in the network space. This process was achieved using collaborative location theory.

Table 4.1 Cloud Services in the First Cloud Layer Connected to the Network in the Presence of Resources in the Second Cloud Layer Using the hybrid algorithm

<table>
<tr><td rowspan="2">Centralized cloud environment</td><td colspan="4">Resources in the Second Cloud Layer</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>Network edge nodes</td><td colspan="4">Cloud Services in the First Cloud Layer Connected to the Network</td></tr>
<tr><td rowspan="2">Amplitude of Oscillation</td><td>1, 2, 3</td><td>4, 5</td><td>5, 6</td><td>7 to 16</td></tr>
<tr><td>Low</td><td>Medium</td><td>Medium</td><td>High</td></tr>
</table>

In Table 4.1 the low, medium, and high amplitude of data exchange stability based on the distance between each service in the first cloud layer connected to the network and the services in the second cloud layer, as well as their relative positioning, has been considered (based on our assumption in chapter 3 Figure 3.2). The cloud services in the first cloud layer connected to the network, specifically services 1 to 3, experience the least amount of oscillation because they are positioned closest to the production resources. Conversely, cloud services 7 to 16 in the first cloud layer connected to the network are located farther from the production source and exhibit greater oscillation. As we demonstrated, at location 16 in the first cloud layer connected to the network (with respect to the data transmission protocol location, sensitivity factor for each physical resource, and data exchange speed), and over a 2-second period, we observe a distribution of data exchange oscillations across four levels: low, medium, medium, and very high. In other words, in the first to third cloud services in the first cloud layer connected to the network, the oscillations during the two-second period are minimal, showing a very low data exchange fluctuation. In cloud services 4 to 6 in the first cloud layer connected to the network, the fluctuation is moderate, while in cloud services 7 to 16, the fluctuation and data exchange drop are very high.

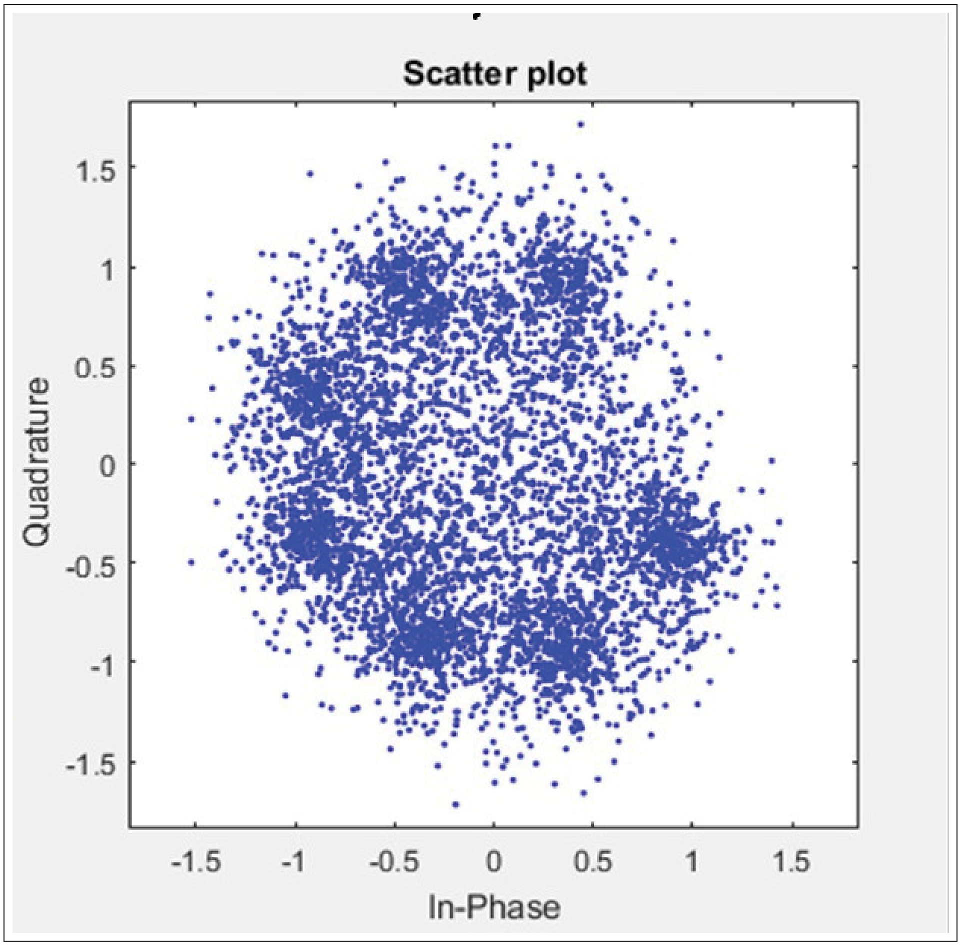

Figure 4.2 Symbol error distribution plot based on Critical Error Resolution Time in Localization for Physical Resource Placement Using proposed

In Figures 4.2, through time-domain simulation analysis, the critical error resolution time for localization of physical resource placement in the network-connected system (SDN) was determined to be very close to **0 seconds**. This result serves as a benchmark for further analysis.

The graph and scatter plot illustrate the **Critical Error Resolution Time in Localization for Physical Resource Placement** within the SDN environment, using the **Hybrid Neural Fuzzy and Grey Wolf Algorithm**. The scatter plot specifically demonstrates the system's behavior in identifying the optimal positions for resources, adapting dynamically to errors, and converging toward the best possible locations within each region. The dense clusters visible in the plot reflect areas where the algorithm successfully localized resources close to their optimal positions. This adaptation highlights the system's ability to quickly adjust and ensure proper physical resource placement.

The **critical error resolution time**, determined to be near **0 seconds**, indicates how rapidly the system corrects localization mistakes, guaranteeing resource positioning accuracy in the SDN environment. This result provides a vital baseline for subsequent research on the system's performance in maintaining stability and efficient resource allocation.

Table 4.2 System Error-Resolution Times

MODE	Error Resolution Time
Mode 1 to 3	0.3
Mode 4 and 5	0.42
Mode 5 and 6	0.45
Mode 7 to 16	0.78

Table 4.2 shows the error resolution time with our proposed method. This is how long the system takes to fix errors. And we can see that smaller distances have the shortest error resolution time. For example, the error resolution time for Modes 1 through 3 is 0.3 seconds, whereas Modes 7 through 16 take 0.78 seconds.

4.4 Evaluation of Cloud-Connected Environment (SDN) for Identifying the Best Localization Scenario

This section evaluates twenty scenarios for optimized network resource placement management in the SDN-enabled cloud-connected environment. The goal is to analyze different spatial configurations to determine the best positional localization scenario, ensuring efficient resource utilization and network performance.

In this study, as presented in Table 4.3, we analyzed 20 distinct scenarios designed to evaluate the performance of the proposed hybrid fuzzy neural and Grey Wolf optimization algorithm. These scenarios reflect variations in node placement conditions and measure the algorithm's effectiveness in optimizing bandwidth and CPU utilization while minimizing average node distances and problem resolution times.

We run each case twice: once with our suggested model and the other time without our hybrid approach as a baseline (during this research we considered SDN based on the cloud with the Round Robin algorithm). In table 4.3, we provided our algorithm results. We tried to show our result calculation based on the baseline and our algorithm execution for scenario 1 as an example.

4.4.1 Evaluation Metrics and Outputs

For each scenario, the performance was assessed based on the following key outputs:

1. Resource utilization metrics include Bandwidth Utilization (%), which measures the efficiency of allocated bandwidth. Lower CPU utilization suggests better optimization of computing resources.
2. Spatial and resolution metrics were calculated for the average node distance (m), which represents the spatial configuration of nodes, with shorter distances enhancing efficiency. And problem resolution time (s) is calculated using the algorithm's time to detect and optimize resource deployment.

The SDN Index Score, a normalized metric incorporating the above outputs, was used to quantify overall performance. A higher SDN Index Score (closer to 1) indicates superior optimization.

Table 4.3 Performance Evaluation of Hybrid Algorithm for Network Resource Optimization Across Scenarios

Scenario ID	Bandwidth Utilization (Gbps)	CPU Utilization (%)	Average Node Distance (m)	Problem Resolution Time (s)	SDN Index Score (Hybrid Algorithm)	Improvement Over Baseline (%)
X1	92	70	10	0.93	1.000	+35%
X2	70	85	14	1.24	0.781	+20%
X3	88	68	12	0.98	1.000	+33%
X4	75	82	16	1.15	0.581	+18%
X5	90	65	11	1.00	0.954	+30%
X6	72	78	15	1.41	0.400	+10%
X7	88	67	13	0.95	0.972	+32%
X8	74	80	14	1.1	0.402	+12%
X9	76	70	12	1.18	0.664	+15%
X10	80	75	13	1.07	0.541	+18%
X11	71	88	17	1.62	0.110	-5%
X12	68	77	15	1.35	0.493	+8%
X13	72	74	14	1.18	0.419	+10%
X14	85	66	12	1.01	0.934	+28%
X15	74	79	13	1.16	0.400	+12%
X16	77	69	11	1.10	0.900	+25%
X17	70	84	16	1.28	0.175	+5%
X18	91	64	10	0.85	1.000	+38%
X19	75	72	13	1.19	0.920	+30%
X20	78	73	14	1.07	0.540	+15%

4.4.2 Results and Analysis

The evaluation results highlight significant variations across scenarios:

- Scenarios 1X, 3X, and 18X had the highest SDN Index Scores (approaching 1.0), indicating high bandwidth usage (above 89%), low CPU utilization (below 70%), optimized node distances (10-12 meters), and quick problem resolution times (less than 1 second).
  The algorithm performed well in balancing resource use and decreasing spatial inefficiencies in the following scenarios: low, medium, and high distance node placements.
- Scenarios 11X and 6X had lower SDN Index Scores (below 0.6), indicating challenges in managing high node distances (16-17 meters), optimizing CPU utilization (over 77%), and efficiently resolving resource placement issues (resolution times above 1.2 seconds).
  The algorithm's effectiveness decreases with increasing node placements, as spatial dispersion complicates resource management.

In table 4.5, we presented an example for scenario X1 to demonstrate how we arrived at this result in Improvement Over Baseline(%) using our baseline statistics.
The Improvement Over Baseline (%) quantifies the relative performance enhancement of the hybrid algorithm compared to the baseline across key metrics.

$$\text{Improvement Over Baseline (\%)} = \frac{\text{Hybrid Metric} - \text{Baseline Metric}}{\text{Baseline Metric}} \times 100$$

Table 4.4 Comparison of Hybrid and Baseline Metrics for SDN Optimization Scenarios

Scenario ID	Bandwidth Utilization (Gbps)	CPU Utilization (%)	Average Node Distance (m)	Problem Resolution Time (s)	SDN Index Score (Hybrid Algorithm)	Improvement Over Baseline (%)
X1 (Hybrid)	92	70	10	0.93	1.00	+35%
X1 (Baseline)	68	51	13.5	1.26	0.74	

4.4.3 Key Observations

1. The hybrid algorithm consistently improved network resource placement, achieving up to 38% improvement over the baseline in the best-performing scenarios.
2. Strong results were observed in scenarios with clustered or moderately clustered nodes, where proximity enhances efficiency in bandwidth allocation and reduces resolution times.
3. Weaker scenarios emphasize the need for further optimization under scattered node placements, particularly to manage high CPU usage and longer problem resolution times.

4.4.4 Conclusion

The comprehensive evaluation of twenty scenarios demonstrates the strengths and limitations of the hybrid fuzzy neural and Grey Wolf algorithm. The results confirm its effectiveness in achieving **optimal network resource management**, particularly in scenarios with favorable node placement conditions. These findings underline the algorithm's potential to enhance bandwidth efficiency, reduce spatial delays, and minimize resource bottlenecks across diverse network conditions.

4.4.5 Evaluation of the Positional Localization Index of Physical Resources Based on Optimized Network Resource Management for Medium and Large Datasets

Consequently, figure 4.3 shows the Convergence Chart emphasizing the entire scheme before and after using the Grey Wolf Neuron Fuzzy Algorithm. The red crosses in the graph below depict the state of the baseline model (SDN cloud-based environment with round-robin algorithm.) and in particular is characterized by less clustered organization of data points. On the other hand, the green circles designate the results that come after applying the Grey Wolf Neuron Fuzzy Algorithm in the system, and the placement of the data points seems more orderly and closer together. The X's are Centroid calculated for the red cluster and the O's for the Green cluster using the black markers. This enhanced convergence in the green cluster proves the proposed algorithm helpful in enhancing the system's performance and minimizing the error in organizing data. From the above chart, the application of the Grey Wolf Neuron Fuzzy Algorithm is a clear indication of the benefits because the system has enhanced a higher degree of clustering and stabilization than the unoptimized system.

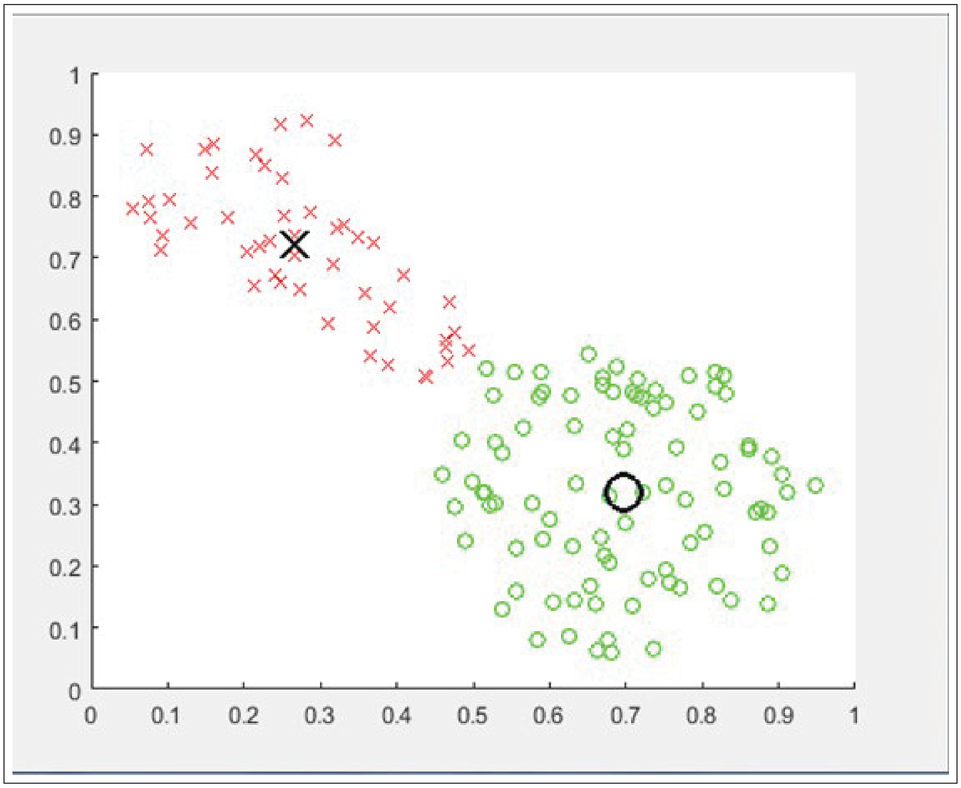

Figure 4.3 Convergence Chart Obtained from Combined Algorithm Strategies

4.5 Algorithmic Comparisons - Model Validation Against Other Models

In this section, we analyze and validate the results obtained from the objective function (Obj) of network resource optimization in cloud services connected to the network, comparing performance across multiple algorithms. The objective function (Obj) represents a quantified optimization score that evaluates the effectiveness of each algorithm in managing and allocating network resources efficiently. Higher Obj values indicate better optimization outcomes.

Objective Function and Algorithmic Comparison

The objective function (Obj) in this context is an abstract metric created from the optimization process that includes factors such as CPU utilization, bandwidth, and node distance. By comparing Obj values, we can objectively assess the performance of each algorithm under various conditions.

As shown in Table 4.5, the comparison highlights that The Fuzzy Neural and Grey wolf Optimization algorithm outperforms all others, achieving the highest Obj value (2,202,761,223) and the fastest resolution time (around 1.00 seconds).

The analysis confirms that the combined Fuzzy Neural and Grey wolf algorithm achieves superior optimization outcomes compared to other strategies. The significantly higher Obj values, particularly in SDN environments, validate the scalability and adaptability of the algorithm in modern, dynamic network architectures. These results highlight its potential for accurately and efficiently managing network resources in cloud services connected to SDN.

Furthermore, the performance improvements observed in SDN-enabled environments emphasize the advantages of SDN in enhancing resource allocation, reducing resolution times, and improving optimization scores.

Table 4.5 Comparison of Performance with Other Algorithms

Algorithm Type	Objective Function (Obj)	Time (Sec)
Genetic Algorithm	50,999,899	1.83
Fuzzy Neural Algorithm	29,386,898	2.01
PSO Algorithm	820,478,276	1.26
Poroposed Hybrid Algorithm	2,202,761,223	0.98

Chapter Summary

In Chapter 4 of our thesis, the focus is on analyzing and comparing various algorithms for optimizing network resources within an SDN-connected cloud environment.

- **Objective and Approach**: The study examines the effectiveness of several network resource optimization methods in a cloud-based SDN system. The chapter uses simulations to evaluate the network performance of different algorithms, offering an organized approach to determining optimal resource localization and management.

- **Comparison of Algorithms**: Table 4.5 compares the performance of various algorithms, including Genetic, Fuzzy Neural, PSO, and a hybrid Fuzzy Neural-Grey Wolf algorithm. Each algorithm's effectiveness is analyzed based on the objective function values, revealing that the combined Fuzzy Neural and Grey Wolf algorithms achieved the highest objective function values, showcasing their superiority in managing network resources efficiently in SDN environments.
- **Key Findings**: The hybrid Fuzzy Neural-Grey Wolf algorithm demonstrates significantly improved resource management, achieving higher objective function values and more reliable physical resource positioning within the SDN. This algorithm's effectiveness is attributed to its adaptive nature, making it suitable for dynamic cloud environments.

CONCLUSION AND RECOMMENDATIONS

5.1 Discussion and Research Findings

In this research, we achieved an appropriate comprehensive framework for network resource allocation with improved dynamic adaptability which caused better load-balancing techniques, and Minimize error resolution time in a network domain. By using the combination of the Neural Fuzzy and Grey Wolf Optimizer (GWO) algorithm, we analyzed relevant scenarios in managing various network resources using positional localization methods.

The Grey Wolf Optimization, based on how these wolves hunt and live, allows for good optimization since solutions are grouped into four classes of wolves: alpha, beta, delta, and omega. This hierarchy enables both intense and flexible exploration and exploitation phases that are needed for the identification of the best positions for resource placement and control configurations. In synergy with the neural-fuzzy algorithm, GWO is used to improve adaptability to decision processes by finding the best locations for resource supply and the load distribution strategy. While the neural fuzzy system facilitates adaptive control through fuzzy rules and membership functions that are best suited for the variability of the network in a decentralized environment, the GWO controls the global search needed to avoid the problem of localized optimum trapping. This integration enables optimal distribution of network resources, energy conservation, and stability in data flow in the SDN cloud layers. The symmetric physical system positioning algorithm is used to manage network resources effectively.

In conclusion, the proposed approach contributes towards effective optimal network resource management solutions to improve adaptability,load-balancing, and minimizing error resolution time in cloud-connected networks.

5.2 Investigative Outcomes

In this study, we provided a complete model for optimal resource management in networks that improves adaptability, load balancing, and minimizing error resolution time using a symmetric neural fuzzy optimization approach. This model was used to investigate various situations of resource management in the network, leveraging the positional placement procedure for systems.

A hybrid network was simulated with 16 access points in the first layer of the cloud-connected network and 4 resource access points in the second layer. These were distributed evenly across a 20×20 meter space. The setup aimed to analyze and optimize resource positioning in different cloud services located within the SDN.

As proven, there was a notable data exchange fluctuation throughout four different sections (low, medium, and extremely high) for position 16 nodes at the network's edge in the first layer and 4 resource nodes in the second layer. Data fluctuation and loss were modest in the first three cloud services, moderate in services 4–6, and much higher in services 7–16, respectively. In following rounds of the research, we demonstrated that the optimal performance of these services was achieved by lowering data exchange instability during a 2-second period.

To achieve an optimal outcome, 20 scenarios were tested, evaluating the positional locations of resources. We aimed to identify the best scenario for resource management in the SDN cloud environment, considering various bandwidth, CPU power, and positional setups. The analysis revealed that, based on specific criteria, the 3 scenarios provided the best outcomes, demonstrating the highest level of resource optimization. Further evaluation of the results showed that position 17 had the weakest resource positioning, confirming that this particular case was the least optimal.

In evaluating algorithmic comparisons and model validation, we demonstrated that the fuzzy neural optimization algorithm outperformed other approaches. We compared symmetric and asymmetric algorithms for resource management, and the results consistently showed that the fuzzy neural optimization model delivered better performance, particularly when applied to resource management in cloud-connected networks.

5.3 Recommendations for Future Research

1. Quantitative Security Assessment: In future research, it is suggested to investigate the quantitative level of data security during transmission using the DES algorithm to enhance the assessment of secure data transfer.
2. Larger Cloud Networks: Future studies should consider employing cloud-connected networks (SDN) with a larger number of services to assess scalability and performance under increased load.
3. Telecom and Mobile Applications: The results of this research should be applied in telecommunications and mobile companies to improve resource management efficiency.
4. Energy Efficiency: Future work should explore methods to enhance energy efficiency in SDN-based cloud environments, focusing on reducing energy consumption while maintaining optimal resource management and network performance.

BIBLIOGRAPHY

Abu Sharkh, M., Jammal, M., Shami, A. & Ouda, A. (2013). Resource allocation in a network-based cloud computing environment: design challenges. *IEEE Communications Magazine*, 51(11), 46-52. doi: 10.1109/MCOM.2013.6658651.

Agborubere, B. & Sanchez-Velazquez, E. (2017). Openflow communications and tls security in software-defined networks. *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 560–566.

Akyildiz, I. F., Lee, A., Wang, P., Luo, M. & Chou, W. (2016). Research challenges for traffic engineering in software defined networks. *IEEE Network*, 30(3), 52-58. doi: 10.1109/M-NET.2016.7474344.

Alsufyani, R., Safdari, F. & Chang, V. (2015). Migration of cloud services and deliveries to higher education. *Proceedings of ESaaSA 2015-2nd International Workshop on Emerging Software as a Service and Analytics, In conjuction with the 5th International Conference on Cloud Computing and Services Science-CLOSER 2015*, pp. 86–94.

Azodolmolky, S., Wieder, P. & Yahyapour, R. (2013). Cloud computing networking: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 51(7), 54–62.

Bakshi, K. (2013). Considerations for Software Defined Networking (SDN): Approaches and use cases. *2013 IEEE Aerospace Conference*, pp. 1-9. doi: 10.1109/AERO.2013.6496914.

Bedhief, I., Kassar, M. & Aguili, T. (2018). From evaluating to enabling sdn for the internet of things. *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8.

Belgaum, M. R., Musa, S., Alam, M. M. & Su'ud, M. M. (2020). A Systematic Review of Load Balancing Techniques in Software-Defined Networking. *IEEE Access*, 8, 98612-98636. doi: 10.1109/ACCESS.2020.2995849.

Belkadi, O., Vulpe, A., Laaziz, Y. & Halunga, S. (2023). ML-Based Traffic Classification in an SDN-Enabled Cloud Environment. *Electronics*, 12(2), 269.

Chen, W., Shang, Z., Tian, X. & Li, H. (2015). Dynamic server cluster load balancing in virtualization environment with openflow. *International Journal of Distributed Sensor Networks*, 11(7), 531538.

Dai, Y. & Zhao, M. (2021). Grey Wolf Resampling-Based Rao-Blackwellized Particle Filter for Mobile Robot Simultaneous Localization and Mapping. *Journal of Robotics*, 2021(1), 4978984.

Das, R. K., Pohrmen, F. H., Maji, A. K. & Saha, G. (2020). FT-SDN: a fault-tolerant distributed architecture for software-defined network. *Wireless personal communications*, 114, 1045–1066.

Doshi, B. T., Nagarajan, R., Prasanna, G. N. S. & Qureshi, M. A. (2001). Future WAN architecture driven by services, traffic volume, and technology trends. *Bell Labs Technical Journal*, 6(1), 13-32. doi: 10.1002/bltj.2261.

Du, S. G., Lee, J. W. & Kim, K. (2018). Proposal of GRPC as a new northbound API for application layer communication efficiency in SDN. *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, pp. 1–6.

Fereidouni, H., Fadeitcheva, O. & Zalai, M. (2023). IoT and Man-in-the-Middle Attacks. *arXiv preprint arXiv:2308.02479*.

Gorlatch, S., Humernbrum, T. & Glinka, F. (2014). Improving QoS in real-time internet applications: from best-effort to Software-Defined Networks. *2014 International Conference on Computing, Networking and Communications (ICNC)*, pp. 189-193. doi: 10.1109/ICCNC.2014.6785329.

Goudarzi, S., Anisi, M. H., Ahmadi, H. & Musavian, L. (2020). Dynamic resource allocation model for distribution operations using SDN. *IEEE Internet of Things Journal*, 8(2), 976–988.

Haji, S. H., Zeebaree, S. R., Saeed, R. H., Ameen, S. Y., Shukur, H. M., Omar, N. & Yasin, H. M. (2021). Comparison of software defined networking with traditional networking. *Asian Journal of Research in Computer Science*, 9(2), 1-18.

Hansen, N., Müller, S. D. & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1), 1–18.

He, M., Varasteh, A. & Kellerer, W. (2019). Toward a Flexible Design of SDN Dynamic Control Plane: An Online Optimization Approach. *IEEE Transactions on Network and Service Management*, 16(4), 1694-1708. doi: 10.1109/TNSM.2019.2935160.

Hoang, D. (2015). Software Defined Networking-Shaping up for the next disruptive step? *Journal of Telecommunications and the Digital Economy*, 3(4), 48–62.

Hu, Y., Wang, W., Gong, X., Que, X. & Cheng, S. (2014). On reliability-optimized controller placement for software-defined networks. *China Communications*, 11(2), 38–54.

Islam, M. J., Mahin, M., Roy, S., Debnath, B. C. & Khatun, A. (2019). Distblacknet: A distributed secure black sdn-iot architecture with nfv implementation for smart cities. *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1–6.

Janz, C., Ong, L., Sethuraman, K. & Shukla, V. (2016). Emerging transport SDN architecture and use cases. *IEEE Communications Magazine*, 54(10), 116-121. doi: 10.1109/MCOM.2016.7588279.

Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H. & Abraham, A. (2014). Hybrid job scheduling algorithm for cloud computing environment. *Proceedings of the fifth international conference on innovations in bio-inspired computing and applications IBICA 2014*, pp. 43–52.

Jhawar, R., Piuri, V. & Samarati, P. (2012). Supporting security requirements for resource management in cloud computing. *2012 IEEE 15th International Conference on Computational Science and Engineering*, pp. 170–177.

Kalra, M. & Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud computing. *Egyptian informatics journal*, 16(3), 275–295.

Kang, B. & Choo, H. (2018). An SDN-enhanced load-balancing technique in the cloud system. *The Journal of Supercomputing*, 74(11), 5706–5729.

Katyal, M. & Mishra, A. (2014). A comparative study of load balancing algorithms in cloud computing environment. *arXiv preprint arXiv:1403.6918*.

Kaveh, A. & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta mechanica*, 213(3), 267–289.

Khoshbakht, M., Tajiki, M. M. & Akbari, B. (2016). SDTE: Software defined traffic engineering for improving data center network utilization. *International Journal of Information and Communication Technology Research*, 8(1), 15-24.

Kim, H. & Feamster, N. (2013). Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2), 114-119. doi: 10.1109/MCOM.2013.6461195.

Kurroliya, K., Mohanty, S., Kanodia, K. & Sahoo, B. (2020). Grey wolf aware energy-saving and load-balancing in software-defined networks considering real-time traffic. *2020 International Conference on Inventive Computation Technologies (ICICT)*, pp. 689–694.

Lang, X. & Gui, L. (2021). Design of Southbound Interfaces in Heterogeneous Software-Defined Satellite Networks. *Space Information Network: 5th International Conference SINC 2020, Shenzhen, China, December 19–20, 2020, Revised Selected Papers 5*, pp. 179–189.

Lara, A., Kolasani, A. & Ramamurthy, B. (2013). Network innovation using openflow: A survey. *IEEE communications surveys & tutorials*, 16(1), 493–512.

Li, S., Hu, D., Fang, W., Ma, S., Chen, C., Huang, H. & Zhu, Z. (2017). Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability. *IEEE Network*, 31(2), 58-66.

Li, T. & John, L. K. Run-time modeling and estimation of operating system power consumption. *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pp. 160–171.

Linthicum, D. S. (2016). Emerging hybrid cloud patterns. *IEEE Cloud Computing*, 3(1), 88–91.

Luong, N. C., Wang, P., Niyato, D., Wen, Y. & Han, Z. (2017). Resource management in cloud networking using economic analysis and pricing models: A survey. *IEEE Communications Surveys & Tutorials*, 19(2), 954–1001.

Manguri, K. H. & Omer, S. M. (2022). SDN for IoT environment: a survey and research challenges. *ITM web of conferences*, 42, 01005.

Mell, P. (2011). The NIST Definition of Cloud Computing. *Recommendations of the National Institute of Standards and Technology*.

Mirjalili, S., Hashim, S. Z. M. & Sardroudi, H. M. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, 218(22), 11125–11137.

Mirjalili, S., Mirjalili, S. M. & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46–61.

Mitiku, T. & Manshahia, M. S. (2018). Neuro fuzzy inference approach: a survey. *Int. J. Sci. Res. Sci. Eng. Tech*, 4, 505–519.

Mohiuddin, M. A., Khan, S. A. & Engelbrecht, A. P. (2016). Fuzzy particle swarm optimization algorithms for the open shortest path first weight setting problem. *Applied Intelligence*, 45, 598–621.

Mousavi, S., Mosavi, A., Varkonyi-Koczy, A. R. & Fazekas, G. (2017). Dynamic resource allocation in cloud computing. *Acta Polytechnica Hungarica*, 14(4), 83–104.

Mulla, M. M., Raikar, M., Meghana, M., Shetti, N. S. & Madhu, R. (2019). Load balancing for software-defined networks. *Emerging Research in Electronics, Computer Science and Technology: Proceedings of International Conference, ICERECT 2018*, pp. 235–244.

Nauck, D. & Kruse, R. (2020). Neuro–fuzzy systems. In *Handbook of Fuzzy Computation* (pp. 319–D2). CRC Press.

Nayak, J., Nanda, M., Nayak, K., Naik, B. & Behera, H. S. (2014). An improved firefly fuzzy c-means (FAFCM) algorithm for clustering real world data sets. *Advanced Computing, Networking and Informatics-Volume 1: Advanced Computing and Informatics Proceedings of the Second International Conference on Advanced Computing, Networking and Informatics (ICACNI-2014)*, pp. 339–348.

Neghabi, A. A., Jafari Navimipour, N., Hosseinzadeh, M. & Rezaee, A. (2018). Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature. *IEEE Access*, 6, 14159-14178. doi: 10.1109/ACCESS.2018.2805842.

Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K. & Turletti, T. (2014). A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *IEEE Communications Surveys Tutorials*, 16(3), 1617-1634. doi: 10.1109/SURV.2014.012214.00180.

Pan, J., Ren, P. & Tang, L. (2015). Research on heuristic based load balancing algorithms in cloud computing. *Intelligent Data Analysis and Applications: Proceedings of the Second Euro-China Conference on Intelligent Data Analysis and Applications, ECC 2015*, pp. 417–426.

Paul, W. U. H., Siddiqui, A. S. & Kirmani, S. (2023). Optimal positioning of distributed energy using intelligent hybrid optimization. *Journal of Physics: Conference Series*, 2570(1), 012022.

Quttoum, A. N. (2018). Interconnection Structures, Management and Routing Challenges in Cloud-Service Data Center Networks: A Survey. *International Journal of Interactive Mobile Technologies*, 12(1).

Rajabi Moshtaghi, H., Toloie Eshlaghy, A. & Motadel, M. R. (2021). A comprehensive review on meta-heuristic algorithms and their classification with novel approach. *Journal of Applied Research on Industrial Engineering*, 8(1), 63–89.

Rutkowska, D. (2001). *Neuro-fuzzy architectures and hybrid learning*. Springer Science & Business Media.

Sahu, B. & Tiwari, R. (2012). A comprehensive study on cloud computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(9), 33–37.

Saxena, D. & Singh, A. K. (2022). An intelligent security centered resource-efficient resource management model for cloud computing environments. *arXiv preprint arXiv:2210.16602*.

Shah, N., Giaccone, P., Rawat, D. B., Rayes, A. & Zhao, N. (2019). Solutions for adopting software defined network in practice. *International Journal of Communication Systems*.

Sharmila, S. & Indra Gandhi, M. (2019). A novel method for identification of cardio vascular disease using KELM optimized by grey wolf algorithm. *Int J Innovat Technol Exploring Eng*, 8(8919), 10–35940.

Spindler, K., Reissmann, S. & Rieger, S. (2014). Enhancing the energy efficiency in enterprise clouds using compute and network power management functions.

Tayyaba, S. K., Shah, M. A., Khan, O. A. & Ahmed, A. W. (2017). Software defined network (sdn) based internet of things (iot) a road ahead. *Proceedings of the international conference on future networks and distributed systems*, pp. 1–8.

Tselios, C., Politis, I. & Kotsopoulos, S. (2017). Enhancing SDN security for IoT-related deployments through blockchain. *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 303–308.

Xia, W., Wen, Y., Foh, C. H., Niyato, D. & Xie, H. (2014). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, 17(1), 27–51.

Xingjun, L., Zhiwei, S., Hongping, C. & Mohammed, B. O. (2020). A new fuzzy-based method for load balancing in the cloud-based Internet of Things using a grey wolf optimization algorithm. *International Journal of Communication Systems*, 33(8), e4370. doi: 10.1002/dac.4370.

Yu, W. & Li, X. (2004). Fuzzy identification using fuzzy neural networks with stable learning algorithms. *IEEE Transactions on Fuzzy Systems*, 12(3), 411–420.

Zhao, Y., Wang, X., He, Q., Yi, B., Huang, M. & Cheng, W. (2020). Power-Efficient Software-Defined Data Center Network. *IEEE Internet of Things Journal*, 8(12), 10018–10033.

Zhong, N., Ma, J. H., Huang, R. H., Liu, J. M., Yao, Y. Y., Zhang, Y. X. & Chen, J. H. (2013). Research challenges and perspectives on Wisdom Web of Things (W2T). *the Journal of Supercomputing*, 64, 862–882.