Computer Standards & Interfaces 76 (2021) 103518

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Optimal 5G network slicing using machine learning and deep learning concepts

Mustufa Haider Abidi [a,*], Hisham Alkhalefah [a], Khaja Moiduddin [a], Mamoun Alazab [b], Muneer Khan Mohammed [a], Wadea Ameen [a], Thippa Reddy Gadekallu [c]

[a] Advanced Manufacturing Institute, King Saud University, Riyadh-11421, Saudi Arabia
[b] IT and Environment, Charles Darwin University, 0909 Darwin, Australia
[c] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore-632014, Tamilnadu, India

ARTICLE INFO

Keywords:
Network Slicing
Deep Learning
Neural Network
Deep Belief Network
Optimal Weighted Feature Extraction
Glowworm Swarm-Deer Hunting Optimization Algorithm

ABSTRACT

Network slicing is predetermined to hold up the diversity of emerging applications with enhanced performance and flexibility requirements in the way of splitting the physical network into numerous logical networks. Consequently, a tremendous data count has been generated with an enormous number of mobile phones due to these applications. This has made remarkable challenges and has a considerable influence on the network slicing performance. This work aims to design an efficient network slicing using a hybrid learning algorithm. Thus, we proposed a model, which involves three main phases: (a) Data collection, (b) Optimal weighted feature extraction (OWFE), and (c) Slicing classification. First, we collected the 5G network slicing dataset, which involves the attributes associated with various network devices like "user device type, duration, packet loss ratio, packet delay budget, bandwidth, delay rate, speed, jitter, and modulation type." Next, we performed the OWFE, in which a weight function is multiplied with the attribute values to have high scale variation. We optimized this weight function by the hybridization of two meta-heuristic algorithms—glowworm swarm optimization and deer hunting optimization algorithm (DHOA)—and named the proposed model glowworm swarm-based DHOA (GS-DHOA). For the given attributes, we classified the exact network slices like "eMBB, mMTC, and URLLC" for each device by a hybrid classifier using deep belief and neural networks. The weight function of both networks is optimized by the GS-DHOA. The experiment results revealed that the proposed model could influence the provision of accurate 5G network slicing.

1. Introduction

With the rapid development of mobile devices, development of several appearing applications, and ever-enhancing necessities for increasing Quality of Service (QoS), 5G communications system is anticipated to confront the unmatched demand and help several vertical industries and new services including automotive communication, augmented reality (AR), Industry 4.0, virtual reality (VR) and remote healthcare [1-3]. Mobile device users and their corresponding data traffic are anticipated to develop unnecessarily in the coming years, affecting an enhanced eight-fold differentiated to 2015 [3, 4]. Third Generation Partnership Project (3GPP) has let out the initial 5G network recommendations giving help for analytical communications, huge vehicular-to-everything, and machine-type communications [5]. Among the remaining, the 5G network central feature is to provide identical infrastructure along with clashing necessities in a feasible manner. This would need an adaptable network architecture that constructs on (i) the new idea of slicing the network, and (ii) the network virtualization pattern permitting for the on-demand initiation of extremely different services in a divided infrastructure that permits to locate various network occurrences of several services [6-8].

Network slicing present in 5G systems determines self-contained, logical networks, which consist of a combination of dedicated and shared instances of resources, such as network equipment or radio spectrum, and virtual network functions (VNF) [4]. Virtualization permits the disjoint of network functions from dominion hardware appliances for generating different building blocks, which can be elastically chained to generate value-added communication services [9-11]. The network slicing pattern permits infrastructure providers such as MNOs (mobile network operators) to unlock their physical network potentials to multiple occupants through the icon of rational self-contained

* Corresponding author.
E-mail address: mabidi@ksu.edu.sa (M.H. Abidi).

https://doi.org/10.1016/j.csi.2021.103518
Received 29 May 2020; Received in revised form 11 September 2020; Accepted 23 January 2021
Available online 26 January 2021
0920-5489/

Nomenclature

Abbreviation	*Description*
GSO	Glowworm Swarm Optimization
5G	Fifth Generation
QoS	Quality-of-Service
DHOA	Deer Hunting Optimization Algorithm
AR	Augmented reality
VR	Virtual Reality
GS-DHOA	Glowworm Swarm-based DHOA
MNOs	Mobile Network Operators
MCDM	Multi-Criteria Decision Making
RL-NSB	Reinforcement Learning-based 5G Network Slice Broker
SGW	Serving Gateway
MME	Mobility Management Entity
PGW	Packet data network gateway
SMF	Session Management Function
AMF	Access and Mobility Management Function
AUSF	Authentication Server Function
DRL	Deep Reinforcement Learning
MEC	Mobile Edge Computing
C-RAN	Cloud Radio Access Network
CPU	Central Processing Unit
NFV	Network functions virtualization
MILP	Mixed Integer Linear Programming
VNF	Virtual Network Functions
GWO	Grey Wolf Optimization
ML	Machine Learning
DLNN	Deep Learning Neural Network
NN	Neural Network
SVM	Support Vector Machine
E2E	End-to-End
DBN	Deep Belief Network
RBMs	Restricted Boltzmann Machines
PSO	Particle Swarm Optimization
KNN	K-Nearest Neighbor
SLA	Service Level Agreement
KPIs	Key Performance Indicators

networks, arranged in distinct ways on the basis of the occupant's particular service demands. Such network slices are (presently) possessed and controlled by the particular occupants [12, 13]. In contrast to conventional resource provisioning, network slicing resource management trades with combined multi-flow resource contracts per slice other than single per-flow contracts that are yet supposed to be controlled with the help of schedulers [14].

A large amount of data of composite nature is required to be examined to perform smart decision for the construction, operation, management, design, deployment, and administration of a network slice, so that it can efficiently fulfill the QoS necessities of the service deliberated to be distributed through it, in spite of the network conditions and time-varying workloads [15-17]. It is complex for humans to generate and work network slices by handling the huge volumes of data in a limited time [18, 19]. Thus, there are demands to make these tasks perform automatically. Machine learning (ML) methods are the organizer for the automation of functions in network slicing. ML has the ability of mining (e.g., service classification), preserving [20, 21], reasoning (e.g., configuration of system parameters for adaptation) [2, 22], security [23, 24], authentication [25, 26], predicting (e.g., predicting user or traffic trend) [27, 28] and sensing (e.g., anomaly detection) [29-31]. Specifically, it can inspect a large quantity of data in a very short time for modulating the system to time-changing environments, predict future occurrences with better accuracy, and direct energetic solutions [32-34]. Therefore, to solve these kinds of problems, bio-inspired algorithms can be applied. Bio-inspired algorithms are gaining admiration in ML field because of their advantages such as they are simple yet effective, robust, self-organized, parallelism in essence and non-determinism [35, 36].

The major contributions of this work are:

- To develop an efficient network slicing with three phases such as data collection, optimal weighted feature extraction (OWFE), and slicing classification, with the aid of hybrid learning.
- To collect the relevant data attributes highly associated with network slicing and accomplish the OWFE process.
- To optimize the weight function with the help of a hybrid meta-heuristic model GS-DHOA (Glowworm Swarm-based Deer Hunting Optimization Algorithm) to attain high classification accuracy.
- To classify the exact network slices with the help of DBN (Deep Belief Network) and NN (Neural Network), with weight optimization by the GS-DHOA, which categorizes the network slicing types into eMBB (enhanced Mobile Broadband), mMTC (machine Type Communications), and URLLC (Ultra-reliable low-latency communication).

The rest of this paper is organized as follows: Section 2 presents the literature survey of related works; Section 3 presents a brief explanation about the combination of ML and DL (Deep learning) for secured 5G network slicing; Section 4 presents the OWFE for 5G network slicing; Section 5 describes the network slicing with the contribution of NN and DBN; Section 6 describes the optimal network slicing using the hybrid meta-heuristic model; Section 7 presents the results and discussions; in the last step, Section 8 concludes the paper.

2. Literature Survey

Bagaa et al. [37] proposed an algorithm, which firstly attained the best count of effective occurrences of 4G (SGW (serving gateway, MME (mobility management entity), and PGW (packet data network gateway)) or 5G (SMF (session management function), AMF (access and mobility management function), and AUSF (authentication server function)) central network components to satisfy the necessities of a particular mobile traffic. They also proposed an algorithm for the positioning of these effective occurrences over a combined cloud. While the first algorithm depended on mixed integer linear programming (MILP), the second depended on coalition formation games; therefore, their objective was to construct unions of cloud networks to host the effective occurrences of the vEPC/5G (virtualized evolved packet core) central components. The attained outcomes obviously demonstrated the advantages of the proposed algorithms in securing the QoS, and provided a constant cost for vEPC/5G central distribution; thereby, maximizing the cloud operator's profits.

Song et al. [38] developed an adaptable grouping edge cloud architecture to authorize 5G optical mobile fronthaul network slicing. Moreover, a related united network resource management technique was developed to help several QoS necessities and different resource demands of the network slices. Experiment outcomes represented that the developed technique was capable of cooperatively allotting the bandwidth resources to network slices and perceived cloud-computing offloading to reach several QoS necessities and consequently, minimized the front haul bandwidth burden.

Li et al. [39] proposed a heuristic 5G core network slice provisioning algorithm, known as VIKOR-CNSP (Core Network Slice Provisioning) based on VIKOR, which was a MCDM (multi criteria decision making) technique. In the slice node provisioning step, they categorized the node significance with the VIKOR technique by considering the topological attributes and node resource. Then, they maintained the slice nodes with the ranking outcomes. In the case of the slice link provisioning step, they carried out the shortest path algorithm to attain the candidate physical tracks for the slice link and developed a scheme for choosing a candidate physical track to enhance the slice acceptance ratio. The scheme initially computed the path factor that was the multiplication of the maximum link bandwidth utilization of the candidate physical path and its related hop-count, and then selected the candidate physical path containing the lowest for hosting the slice link. Numerous experiments revealed that the proposed algorithm attained the maximum slice acceptance ratio and the highest provisioning revenue-to-cost ratio, fulfilling the security conditions of 5G core network slice demands.

Sciancalepore et al. [40] labeled the conflict by shaping the succeeding building blocks for helping network slicing: (i) a learning and predicting technique per slice, (ii) a reinforcement technique to drive the system in the path of optimal states, (iii) traffic and user mobility analysis, and (iv) optimal admission control decisions on the basis of traffic and spatial information. A technique, known as RL-NSB infrastructure providers does admission control by considering the SLA of the several occupants along with their user distribution and traffic usage, and improved the comprehensive technique with the help of reinforcement process and the learning that contemplated traffic models and heterogeneous mobility within various slices. The outcomes revealed that by depending on properly tuned predicting techniques, the proposed technique gave very considerable probable gains with respect to system utilization while contacting the occupant's SLAs.

Wang et al. [41] proposed a ML-based technique for dynamic resource scheduling for network slicing, focused on attaining effective and automatic E2E service reliability and resource optimization. Therefore, it was complex to attain the user-related data; hence, it was critical to learn the demands and user characteristics owing to the safety problem. However, DRL was weighted to withdraw learning from occurrences by interrelating with the network and authorize energetic adaptation of the resources allotted to several slices for maximizing the resource utilization; thereby, undertaking the QoS. The simulation outcomes revealed that the proposed resource scheduling technique can energetically allot resources for multiple slices and fulfill the related QoS necessities.

Wang et al. [42] addressed a SliceNet framework based on customizable and advanced network slicing to label a few of the spotlighted provocations in transferring eHealth telemedicine services to 5G networks. An outline of the component forms the technical methods in the past of the existing network slicing. Consequently, the shape and modeling of a media-centric eHealth use case was highlighted, concentrating on a group of novel enablers on the path of attaining E2E (end-to-end) QoS-aware network slicing abilities, involved by this demanding use case. Simulation outcomes proved the modeled enablers and revealed the applicability of the developed component in such type of media-rich use case.

AlQahtani and Alhomiqani [43] proposed a model for 5G mobile networks, including MEC (multi-access edge computing), C-RAN (Cloud radio access network), and cloud data center concerning the network slicing problem. The researchers designed the network slicing system on the basis of queueing theory, which can be employed to obtain the major behavior metrics, such as system throughput, an average number of message requests, average waiting time, CPU (central processing unit) utilization, system drop rate, and average response time. The researchers gave significant illustrations to display how this proposed technique could be appealed to determine the system behavior and cost for a network slicing system in 5G mobile networks; and, the count of MEC and C-RAN cores demanded below different 5G traffic conditions. The systematic outcomes and experimental processes demonstrated that the proposed model has a robust capability to allocate the count of MEC and C-RAN cores required to attain the quality of service targets of 5G slices.

Table 1
Features and Challenges of Conventional Network Slicing in 5G using ML Algorithms

Author [citation]	Methodology	Features	Challenges
Song et al. [38]	NFV	• It is leveraged by the network slicing for realizing both NFV and infrastructure virtualization. • It is employed for enabling the dynamic sharing of network resources over operators.	• Its performance needs enhancement.
Li et al. [39]	K-shortest path algorithm	• It is utilized for obtaining the physical paths of the candidate in the slice link, which is used for enhancing the slice acceptance ratio. • It is used for solving the link mapping.	• It cannot hold the negative edges.
Sciancalepore et al. [40]	RL	• It properly assigns the cell resources to various tenants. • It has high performance.	• A huge reinforcement will cause a state overload, which reduces the outcomes.
Bagaa et al. [37]	MILP	• It is used to devise the optimal count of virtual resource instances related to various VNFs of vEPC/5G core. • It is employed to attain the best optimal solution.	• It cannot consider non-linear effects.
Wang et al. [41]	DRL	• It is leveraged for extracting the knowledge from experience by interacting with the network. • To optimize the resource allocation of each slice, this method is employed for automatic decision making.	• It is not appropriate for solving small issues.
Wang et al. [42]	ML algorithm	• It is used to enhance the pathways of the patient treatment. • Has high performance.	• It requires more amount of training data.
AlQahtani and Alhomiqani [43]	NFV	• It is used to split single physical infrastructure into several virtual wireless networks. • Has improved operational simplicity.	• It needs to manage IT in the abstract.
Thantharate et al. [44]	DLNN	• It is used to predict the network resource reservation for further processing. • It works well even the data is unstructured and vast.	• It is quite expensive for training due to complicate datasets.

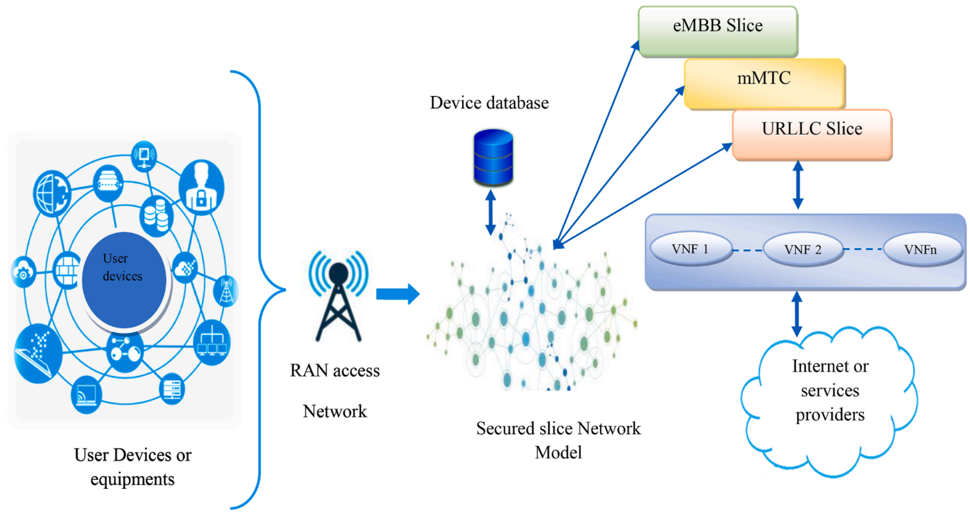

Fig. 1. Overview of secured 5G network slicing model

Thantharate et al. [44] proposed a DeepSlice technique by executing DLNN (deep learning NN) to handle network availability and network load efficiency, employing a prediction and network DL. The researchers employed available network KPIs to direct the technique to examine incoming traffic and predict the network slice for an unrevealed device type. Intelligent resource allocation permits to employ the obtainable resources on conventional network slices effectively and offered load balancing. The proposed technique can provide smart decisions and chose the most suitable network slice, though in times of a failure of a network.

Although there are several ML algorithms used for 5G network slicing, there are still some conflicts with the conventional algorithms. Therefore, a new method needs to be introduced for efficient network slicing. The benefits and limitations of the existing ML algorithms

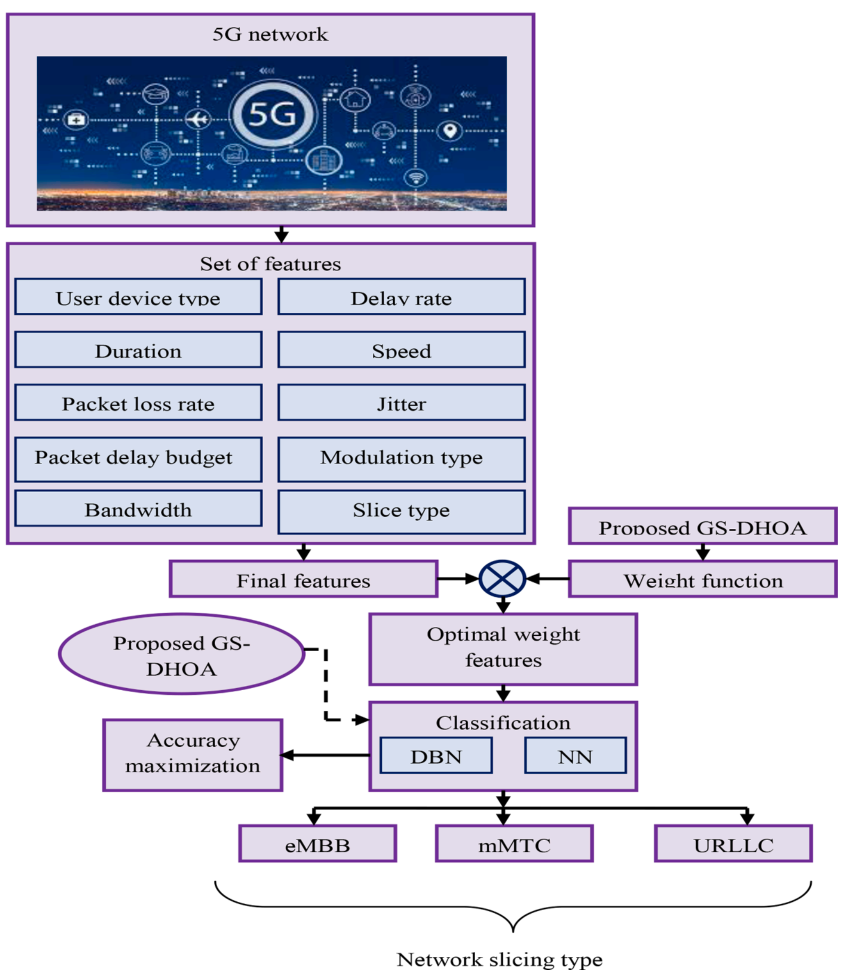

Fig. 2. Architectural diagram of the proposed 5G network slicing model using DL

employed for network slicing are summarized in Table 1. Among these algorithms, NFV (Network functions virtualization) [38, 43] is leveraged by the network slicing for realizing both NFV and infrastructure virtualization. It is employed for enabling the dynamic sharing of network resources over operators, it is used to split single physical infrastructure into several virtual wireless networks, and it has improved operational simplicity. However, its performance needs further enhancement. K-shortest path algorithm [39] is utilized for obtaining the physical paths of the candidate in the slice link, which is used for enhancing the slice acceptance ratio and solving the link mapping. However, it cannot hold the negative edges. RL (Reinforcement Learning) [40] properly assigns the cell resources to various tenants and has high performance. However, a huge reinforcement will cause a state overload, which reduces the outcomes. MILP [37] is used to devise the optimal count of virtual resource instances related to various VNFs of vEPC/5G core and is employed to attain the best optimal solution. However, it cannot consider non-linear effects. DRL (Deep reinforcement learning) [41, 45] is leveraged for extracting the network integration data, and to optimize the resource allocation of each slice, and this method is employed for automatic decision making. However, it requires more amounts of training data. ML [42] is used to improve the pathways of the patient treatment, and has high performance. However, it requires more amount of training data. DLNN [44] is used to predict the network resource reservation for further processing, and it works well even if the data is unstructured and vast. However, it is quite expensive for training due to complicating datasets. The above-mentioned challenges might guide a researcher in developing an efficient model for network slicing.

3. Combination of ML with DL for Secured Network Slicing in 5G Network

3.1. Secured Network Slicing

Owing to the rising situation of the interconnection of the huge count of new services and new devices; the 5G communications have reached a complex point. In this scenario, the effective management of network resources has become crucial while arranging the virtual network slices that are adaptable for offering more favorable services for various users. Fig. 1 shows the overview of 5G network slicing.

3.2. Proposed Architecture

Network slicing is considered as an inventive and emerging notion to attain diverse configurable slices from a physical network. Thus, it could allow the network to congregate the various needs of expected services in the 5G era. The analysis of the massive quantity of data in a short duration, proper learning of operations in different environments, and future prediction with high accuracy could be made possible by the development of ML algorithms. Fig. 2 shows the proposed architecture of network slicing classification.

Initially, we observed the number of devices communicated in the 5G network. It involves smartphones, smart transportations, industry 4.0, IoT devices, gaming, public safety, healthcare, etc. From the different devices or users, we collected attributes, such as user device type, duration, packet loss ratio, packet delay budget, bandwidth, delay rate, speed, jitter, and modulation type. After collecting these attributes, we normalized the data to make the values of attributes in between 0 and 1, which we used to minimize the occurrence of redundant data. Then, we performed the OWFE, in which the proposed GS-DHOA supports to optimize the weight function. With the newly extracted weight-optimized features, we accomplished the network slicing prediction with the aid of hybridizing NN and DL model called DBN through AND operation. The weight of both the NN and DBN gets optimized by the developed GS-DHOA. The main intent of the OWFE and optimized hybrid classification is to maximize the classification of network slicing. The types of predicted slices are eMBB (enhanced mobile broadband), mMTC (massive machine type communications), and URLLC (ultra-reliable low-latency communication).

3.3. Feature Description

The various features used for the network slicing in 5G networks are described below.

User device type: It is a group of properties, which describe the various parts and characters of a type of device. Example: Smartphone and IoT devices.

Duration: It describes how long something persists from the starting to the end. Example: 600 and 180 s.

Packet loss rate: It happens when one or many packets of data transferable across a computer network crash to outreach their endpoint. It is calculated as a percentage of packets vanished with respect to packets transmitted. Example: 0.01 and $1.00e^{-05}$.

Packet delay budget: It describes the upper limit of the delay tolerated by a packet. Example: 7 and 22.

Bandwidth: It defines the maximum rate of data transfer of an internet connection or a network. It calculates how much amount of data can be transmitted over a particular connection in a specified amount of time. Example: 60 and 29GHz.

Delay rate: It is the duration of time before an event happens. Example: 900 and 10500 Mbps.

Speed: It is the dimension of the location variation and it represents a scalar quantity. Example: 1082 and 864 Mbps.

Jitter: It represents the variation from the true periodicity of a probably periodic signal, which is mostly in link to a reference clock signal. Example: $4.30e^{07}$ and $1.90e^{07}$ ps.

Modulation type: It is the technique of changing one or more properties of a cyclical waveform, known as the carrier signal, modulating signal that usually has information to be transferred. Example: FBMC, UFMC

4. OWFE for 5G Network Slicing

4.1. Objective Model

The two main contributions performed in the proposed optimal 5G network slicing is to accomplish the OWFE and optimal hybrid classification. We use the proposed GS-DHOA to enhance the performance of both contributions. Hence, the main objective considered for the OWFE and optimal hybrid classification is to maximize the accuracy of classification regarding slicing, which is represented as follows:

$$Obj = \max(Accu) \tag{1}$$

The numerical equation for determining the accuracy (*Accu*), is given by:

$$Accu = \frac{TP_o + TN_o}{TP_o + TN_o + FP_o + FN_o} \tag{2}$$

where, “TP_o, TN_o, FP_o, and FN_o represents true positives, true negatives, false positives, and false negatives, respectively.

4.2. Weighted Feature Extraction

Let $F_{Fr} = F_1, F_2, \ldots\ldots, F_n$ be the normalized features, where n is the length of features Then, the computation of weighted features is as follows:

$$NF_{Fr} = F_{Fr} \times W_{Fr} \tag{3}$$

In the above equation, NF_{Fr} represents the new features, and W_{Fr} represents the weight function to be used to scaling the features.

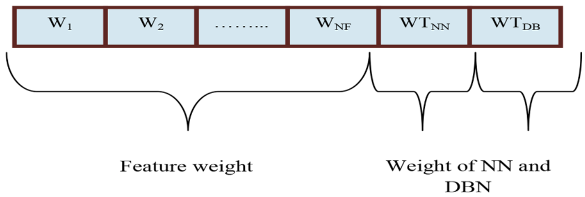

Fig. 3. Solution encoding for OWFE and classification for network slicing

4.3. Solution Encoding

The solution encoding involves optimizing the feature weights and weights of DBN and NN, as shown in Fig. 3, which is done by proposed GS-DHOA.

The minimum and maximum bounding limits of feature weights are 0 and 1, respectively. Moreover, the terms WT_{NN} and WT_{DB} are the weights of NN, and DBN, respectively.

5. Network Slicing with the Contribution of NN and DBN

5.1. Neural Networks

NN [46] is an interrelated group of simple units, processing elements or nodes. The processing capability of the network is kept in the inter-unit connection weights, or strengths, attained by a technique of learning from, or adaptation to, a group of training samples. We used the label "network" to denote any system containing artificial neurons. This may vary from an easy, as a unique node, to a big group of nodes in which every node is linked to every remaining node present in the network. We represent every node by a circle, but weights are complete in the case of all interconnections. We organized the nodes in a covered layout, in which every signal emerges from an input and proceeds through two nodes before outreaching an output after which we do not convert it. This feedforward layout is one among various accessible and is usually employed to spot an input design into one among various categories relative to the evolving design containing outputs. The "knowledge" the network contains is presumed to be kept in its weights, which are developed by a technique of adaptation to stimulus from a group of design illustrations. An input design is given to the network and its response is then differentiated with a selected output. The distinction between the two designs of output describes how the weights are being modified. Every specific method for alteration composes a learning rule. When we did the necessary weight updates, we made another design in which the output is differentiated with the target, and finally, we did the novel alterations. We insistently replicated this succession of occurrences more times until the characteristics of network intersect such that its reaction to every design is near to the relative target. The complete process involving any standard for stopping the technique, requesting of design presentation, etc., includes the training algorithm.

The result of the total network is computed based on the hidden layer output. The hidden layer's output and the output layer are respectively given by:

$$\overline{E}^{(E)} = act\left(\widetilde{D}^{(E)}_{\widehat{Bw}} + \sum_{q=1}^{In(t)} \widetilde{D}^{(E)}_{nh} GT_j\right) \tag{4}$$

$$\widehat{F}_s = act\left(\widetilde{D}^{(M)}_{\widehat{Bo}} + \sum_{r=1}^{HIN} \widetilde{D}^{(F)}_{ho} \overline{E}^{(E)}\right) \tag{5}$$

where, $\widetilde{D}^{(E)}_{(\widehat{Bw})}$ represents the bias weight to the hidden neuron; $\widetilde{D}^{(E)}_{(nh)}$ represents the weight from the input neuron to the hidden neuron; *In(t)* represents the number of input neurons; *act* denotes the activation function; GT_j represents the features of input. In Eq. (5), $\widetilde{D}^{(M)}_{(\widehat{Bo})}$ represents the bias weight of output neuron; $\widetilde{D}^{(F)}_{(ho)}$ represents the weight from hidden to the output neurons; and *HIN* denotes the number of hidden neurons. The term $D^{NN}_{wf} = \{\widetilde{D}^{(E)}_{(\widehat{Bw})}, \widetilde{D}^{(m)}_{(\widehat{Bo})}, \widetilde{D}^{(E)}_{(nh)}, \widetilde{D}^{(F)}_{(ho)}\}$ represents the weight function that is being optimally selected for providing better training to NN by obtaining minimum error difference based on the formula:

$$ER = \underset{\{D^{NN}_{wf}\}}{\arg Min} \sum_{s=1}^{F(t)} \left| E_s - \widehat{E}_s \right| \tag{6}$$

where, E_r and $\widehat{E}_r$, represents the actual and predicted outcomes, respectively. As an enhancement, we optimized the weight function using the proposed GS-DHOA model.

5.2. Deep Belief Network

DBN [47-49] is a learning technique based on deep NN containing the ability of unsupervised pre-learning. DBN consist of several RBMs. DBN initially presents section-wise unsupervised and greedy learning, in which every layer is mentioned by an RBM. DBN is instructed in an unsupervised format to rebuild its inputs. Every section of DBN performs as a feature generator and transforms the input to further abstract description. Following the unsupervised learning, DBN can then be fine-tuned via supervised learning to do regression or classification. This supervised learning following the unsupervised greedy layered-wise training is largely achieved through gradient descent. DBN is a heap of numerous RBMs so that the hidden layer of every RBM performs as a visible layer for the succeeding RBM. The visible layer of the starting RBM is also the visible layer of the DBN, and all remaining layers are hidden layers of DBN. DBN is trained through training a unique RBM at a time. Once the initial RBM has been trained, we transfer training patterns via it, and the output we give the output generated at its hidden layer as input to the visible layer of the succeeding RBM; and, we keep repeating this process. This is known as the layer-wise pre-training of DBN. The unsupervised pre-training and layer-wise greedy of DBN permits to train a network with a huge count of hidden layers; and, this is impossible with the backpropagation algorithm.

DBN employs the Boltzmann network for effectively attaining the outcomes. The output from DBN represents the binary format and it is represented by *oup*. Additionally, the output is composed of the probability of sinusoidal function ($Bnz_{prb}(\lambda)$)as follows:

$$oup = \begin{cases} 1 & with\, 1 - Bnz_{prb}(\lambda) \\ 0 & with\, Bnz_{prb}(\lambda) \end{cases} \tag{7}$$

$$Bnz_{prb}(\lambda) = \frac{1}{1 + e^{\frac{-\lambda}{pte}}} \tag{8}$$

where, *pte* represents the pseudo temperature parameter, which is used to sustain the noise level of the probability. Eq. (9) represents the stochastic system. Depending upon the Boltzmann distribution, we shape the Boltzmann system with the purpose of accurately modeling the input patterns. Eq. (10) represents the Boltzmann model's energy, which is used to construct the neuron states in which *ns* represents the neuron state, $W^{\bullet}_{c,d}$ represents the weight between the neurons, and β represents the biases of neurons. The Boltzmann model in DBN surrounds the neurons with the help of Eq. (11). Moreover, Eqs. (12)–(14) represent the energy-based organization between hidden neurons and visible neurons.

$$\lim_{pte \to 0^+} Bnz_{prb}(\lambda) = \lim_{pte \to 0^+} \frac{1}{1 + e^{\frac{-\lambda}{pte}}} = \begin{cases} 0 & for\, \lambda < 0 \\ \frac{1}{2} & for\, \lambda = 0 \\ 1 & for\, \lambda > 0 \end{cases} \tag{9}$$

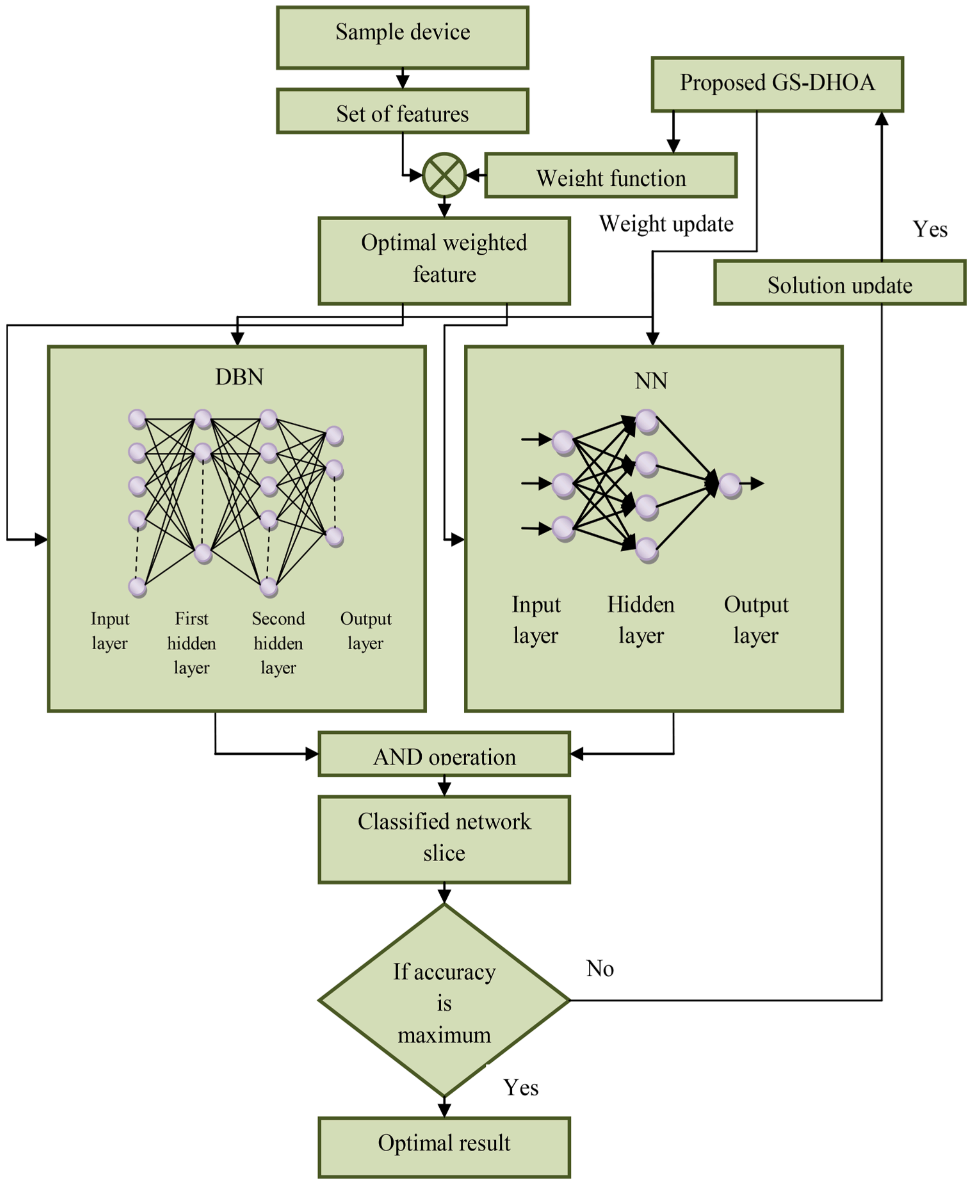

Fig. 4. OWFE and classification for 5G network slicing

$$BnE(nes_1) = \sum_{c<d} \dot{W}_{c,d} nes_c nes_d - \beta_c nes_c \tag{10}$$

$$\Delta BnE(e_c) = \sum_d nes_c \dot{W}_{c,d} + \beta_c \tag{11}$$

$$BnE(vis', hide) = -\sum_{(c,d)} \dot{W}_{c,d} vis_c hide_d - \sum_c vis_c Bj^*_c - \sum_d hide_d hv_d \tag{12}$$

$$\Delta BnE\left(vis_c, \overrightarrow{hide}\right) = \sum_d \dot{W}_{c,d} hide_d + Bj^*_c \tag{13}$$

$$\Delta BnE\left(\overrightarrow{vis}, hide_d\right) = \sum_d W_{c,d} vis_c + Bj^*_c \tag{14}$$

In the above equations, vis_c and $hide_d$ represent the binary state of visible and the hidden neurons, respectively; and hv_c and hv_d represent their bias weight, respectively; and,$Bj^*_c$and Bj^*_drepresent their considered bias, respectively. Since the parameter of the weight function reduces the probability distribution of the input data, the RBM training set reduces the considered probabilities during RBM learning. Additionally, the assistance gained from the energy function makes the RBM capable of giving the probability for every hidden; and, Eq. (15) gives the visible neuron. In Eq. (16), P denotes the partition function.

$$RB'(vis', hide) = \frac{1}{PF} e^{-G(vis', hide)} \tag{15}$$

$$PF = \sum_{\overrightarrow{vis}, \overrightarrow{hide}} e^{-G'(vis', hide)} \tag{16}$$

The standard Boltzmann machine automatically describes the hidden or visible neurons, which is not identical to RBM by calculating the variance of energy between them. RBM is effective for data recreation in an unsupervised manner and it classifies the data by means of unsupervised learning. The time needed by RBM to gain the link by the developed technique is very difficult, and therefore, we selected the CD technique and the algorithm is represented as follows.

1. A training sample v_i is selected and combined with visible neurons.
2. The multiplication of visible vector with the weight matrix WeM returns the hidden neuron probability pr_{hide}as $pr_{hide} = \sigma(vis \bullet WeM)$.

3. Sample the hidden states *hide*from the gained probabilitiespr_{hide}.
4. The positive gradient is computed by multiplying *vis*and pr_{hide}as follows $\varphi^{+} = vis \bullet pr_{hide}^{G}$.
5. The recreation of visible states (vis')is sampled from hidden states ($hide'$). Again resampling is performed in the hidden states from the visible states recreation.
6. The negative gradient is computed by multiplying the outer product of vis' and $hide'$ as follows $\phi^{-} = vis' \bullet hide'^{G}$.
7. The weight updates are computed by subtracting the negative gradient from the positive gradient as follows $\Delta WeM = \eta(\phi^{+} - \phi^{-})$.
8. With the help of the new values, the weights are updated as follows $wt_{c,d}^{*} = \Delta wt_{c,d} + wt_{c,d}$.

The unsupervised learning happens in RBM and the supervised learning happens in NN. We used gradient descend technique in every layer of RBM and NN. Additionally, OA^{ve} and OP^{ve}represent the actual and predicted output vectors, respectively. Finally, the error function between the actual and the predicted vectors is given by:

$$EF = OA^{ve} - OP^{ve} \tag{17}$$

5.3. Hybrid Classifier

Fig. 4 shows the operation of a hybrid classifier in performing the network. We merged the outputs of the two optimized classifiers by AND operation.

6. Optimal Network Slicing using Hybrid Meta-Heuristic Algorithm

6.1. Conventional DHOA

The primary objective of this algorithm [50] is to inspect the excellent place for human beings to hunt the deer. It is remarkable to review the habits of deer. The deer have distinct personalities that make the hunting composite for carnivores. Among these personalities is the eye vision that is five times much better, compared with that of humans.

Initializing the population: We initialized the hunter's population in the first step of this algorithm. Eq. (18) gives the mathematical representation of the population initialization.

$$q = \{q_1, q_2, ..., q_q\};\ 1 < h \leq r \tag{18}$$

where, r represents the total number of hunters that denote the solutions in the population q.

Initialization of parameters: Next to the initialization of the hunter's population, we defined the necessary parameters for describing the optimal hunter's location known as the deer's position angle, as well as the wind angle. We described the search space in a circular format; and therefore, the wind angle revolves around the circumference of a circle. The mathematical representation as follows:

$$\theta_t = 2\pi s \tag{19}$$

Where s represents a random number with a value in the interval range of [0, 1] and t represents the current iteration. The position angle of the deer is mentioned in Eq. (20).

$$\varphi_t = \theta + \pi \tag{20}$$

where, θ represents the wind angle.

Position propagation: The best position of the space is not known at the initial time. Based on the best solution, the algorithm assumes it to be near to the candidate solution, and based upon that, the fitness function is described. The two solutions considered are the leader's position represented as q^{lead} and the next hunter's position (i.e., the successor position) represented as $q^{successor}$.

(i) **Propagation via a leader's position:** Every individual in the population attempts to acquire the optimal solution by describing the best positions and therefore the technique of updating the position starts. The encircling characteristic is modelled as given in Eq. (21).

$$q_{t+1} = q^{lead} - C.a.\left|T \times q^{lead} - q_t\right| \tag{21}$$

where, the term q_t represents the current iteration's location, q_{t+1} represents the next iteration's position, C and T denotes the coefficient vectors and a denotes a random number that is produced by considering the wind speed, whose value ranges among 0 to 2. The coefficient vectors are mathematically computed as represented in Eq. (22) and Eq. (23).

$$C = \frac{1}{4}\log\left(t + \frac{1}{t_{\max}}\right)r \tag{22}$$

$$T = 2.s \tag{23}$$

In the above equations, t_{max} represents the current iteration, r represents a parameter, and s represents a random number in the interval [0,1]. When the value of $r < 1$, the position update takes place as in Eq. (22), where the individual can randomly move in any of the directions without considering the position angle.

(i) **Propagation through position angle:** The update rule describes the position angle for enhancing the search space. We defined the hunter's position by the angle calculation so that the prey is unaware of the attack; therefore, the hunting process will be more efficient. Mathematically, the visualization angle of the prey or the deer is as follows:

$$p_t = \frac{\pi}{8} \times s \tag{24}$$

where p_t is the angle of visualization of the deer or the prey. Based on the difference between the visual angle and the wind angle of the deer, we updated the position angle by a parameter, and it is represented as m_t, given by:

$$m_t = \theta_t - p_t \tag{25}$$

where, θ_t denotes the wind angle. Currently, the position angle is updated for the successor iteration as follows:

$$\varphi_{t+1} = \varphi_t + m_t \tag{26}$$

The position update is done based on the position angle, which is mathematically given by:

$$q_{t+1} = q^{lead} - a.\left|\cos(\nu) \times q^{lead} - q_t\right| \tag{27}$$

where, $G = \varphi_{t+1}, q_t^{*}$represents the best position and a represents the random number. The position of the individual is nearly opposite to the position angle; hence the hunter is not in the vision of the deer.

(i) **Propagation through the successor's position:** The same concept we employed in the encircling behavior can also be used in the exploration phase using the coefficient vector$T$. In the beginning, the presumption was in the path of the random search; and hence, the value of the coefficient vector $T$is known to be less than 1. The location update depends on the position of the successor, except for the first best solution. This follows the global search as follows:

$$q_{t+1} = q^{successor} - C.a.|T \times q^{successor} - q_t| \quad (28)$$

In this equation, $q^{successor}$ represents the successor position of the search agent from the present population. Depending on the acquired best solution from the randomly initialized solutions, the algorithm updates the position of the search agent.

When $|T| < 1$, we select a search agent randomly; while, when $|T| \geq 1$, we select the optimal solution for updating the position of the agent.

Through adjustable variations, the algorithm moves around the exploitation and exploration phases. We performed the position update at every iteration until we acquired the optimal position, which is represented as the ending criteria and depends on the objective function. Algorithm 1 gives the pseudocode of the conventional DHOA.

6.2. Conventional GSO

The GSO algorithm [51] is inspired by a swarm of glowworms that are dispersed randomly in the search space containing object functions. Consequently, these kinds of glowworms transfer a luminescent volume known as luciferin together with them and they contain their self-decision domains$^j_e(0 < s^j_e \leq s_t)$. The glowworms release light in which the intensity is proportionate to the related luciferin and interrelate with the remaining glowworms inside a variable neighborhood. The luciferin intensity of the glowworm is linked to the fitness of their present positions. The more the intensity of luciferin, the better the position of glowworm, otherwise the glowworm denotes a better target value. In other case, the target value is too low. A glowworm j describes another glowworm k as its neighbor if and only if k is inside the neighborhood range of j and the level of luciferin of the neighbor k is more than that of j. In general, we describe the neighborhood as a local-decision domain that contains a variable neighborhood ranges^j_e, which is bounded by using a radial sensor range$s_t(0 < s^j_e \leq s_t)$. Through a probabilistic mechanism, every glowworm chooses a neighbor that contains a luciferin value more than its' self and proceeds in the path of it. This means that, the glowworms are fascinated with the help of neighbors that shine brighter. Additionally, we determine the dimension of the neighborhood range of every glowworm by the amount of glowworms in the range of its neighborhood. The neighborhood range of the glowworm is proportionate to its neighbor's density. If the range of the neighborhood protects less density of glowworms, the range of the neighborhood will be enhanced, vice versa. Usually, a GSO algorithm involves four phases as follows:

Initial distribution of glowworms phase: It is also known as the initialization phase. The objective of this phase is to make the glowworms disperse randomly in the search space of object functions. Consequently, these glowworms transfer the identical intensity luciferin and they contain the identical decision domain.

Luciferin-update phase: The intensity of luciferin of the glowworm is linked to the fitness of their present positions. The more the luciferin's intensity the far better the glowworm's position, in the other case, the glowworm describes a good target value. Otherwise, the value of the target is too bad. In every iteration step of the algorithm, the location of all the glowworms vary; and then, the value of the luciferin also tends to follow the updates. In time t_i, the position of the glowworm j is$y_j(ti)$; the relating value of the objective function at glowworm j'_s location at the time t_i is$K(y_j(ti))$, place the $K(y_j(ti))$into the$lu_j(ti)$. $lu_j(ti)$denotes the level of luciferin related to glowworm j at time t_i. The mathematical formula of the luciferin update phase is represented in Eq. (29).

$$lu_j(t) = (1-\rho)lu_j(t-1) + \gamma K(q_j(t)) \quad (29)$$

where, ρ denotes the luciferin decay constant ($0 < \rho < 1$), and γ denotes the luciferin enhancement constant.

Movement phase: In this phase, each glowworm chooses a neighbor and then goes in its direction with a specific probability. Since the glowworm j'_s neighbor requires to contact two necessities such as the glowworm inside the decision domain of glowworm j and the value of the luciferin should be higher than the glowworm j'_s. Glowworm j goes in the direction of the neighbor k that generates from $O_j(ti)$with a specific probability, the probability is$q_{jk}(ti)$. After the movement, we update the position of the glowworm j. The mathematical formula is as follows:

$$u_{jk}(t) = \frac{lu_k(t) - lu_j(t)}{\sum_{l \in O_j(t)} lu_l(t) - lu_j(t)} \quad (30)$$

When the glowworm j moves, the location is updated as represented in Eq. (31).

$$q_j(t+1) = q_j(t) + ss * \left(\frac{q_k(t) - q_j(t)}{\| q_k(t) - q_j(t) \|} \right) \quad (31)$$

where ss represents the step size.

Neighborhood range update phase: When we update the position of the glowworm, its range of the neighborhood also tends to follow the update. If the range of the neighborhood envelops less volume of glowworms, it will be enhanced. Conversely, it will be minimized, represented as follows:

$$s^j_e(t+1) = \min\{s_t, \max\{0, s^j_e(t) + \beta(o_t - |O_j(t)|)\}\} \quad (32)$$

Here, the parameter o_{tim} and a constant parameter β handles the neighbor's count. Algorithm 2 gives the GSO algorithm pseudocode.

6.3. Proposed GH-DHOA

The usage of optimization algorithms has gained much attention among the scientists. Various alterations and improvements have been done by optimization algorithms for handling complex problem types. Optimization algorithms are commonly used in most of the engineering problems [52, 53]. DHOA [50] is inspired by the hunting nature of a hunter in the path of a deer; it handles the engineering problems very efficiently. However, it suffers from drawbacks, such as it focuses only on the global minima and not on the global maxima. To overcome the drawbacks, GSO [51] is integrated into it and the new algorithm is termed as GS-DHOA. It enhances the convergence in terms of the overall analysis of the algorithm. Hybrid optimization algorithms are analyzed to be a promising one for a particular type of problem. Optimization procedures or mechanisms are linked to generate a hybrid optimization algorithm. In most of the hybrid optimization algorithms, fast convergence is acquired by the applications of several optimization algorithms [12]. The convergence analysis of the hybrid optimization algorithms is observed to be far better than the state-of-the-art algorithms.

Generally, in the existing DHOA, if $|T| \geq 1$, we update the solution by the leader's position, as in Eq. (28). In the proposed GS-DHOA, if $|T| < 1$, we update the solution by the position angle of DHOA, as in Eq. (27). Otherwise, the solution is updated with the movement phase of GSO, as in Eq. (31). Algorithm 3 gives the pseudocode of the proposed GS-DHOA.

7. Results and Discussion

7.1. Experimental Setup

We executed the proposed 5G network slicing in MATLAB 2018a and carried out the performance analysis. We took the population size as 10 and performed a maximum of 25. Herein, we compared the performance of the proposed GS-DHOA-NN+DBN with other conventional optimization-based NN+DBN algorithms, such as PSO-NN+DBN, GWO-NN+DBN, GSO-NN+DBN, and DHOA-NN+DBN, as well as various ML algorithms like SVM, KNN, NN, DBN, and NN+DBN; and, analysed the results. We carried out the analysis with various performance measures,

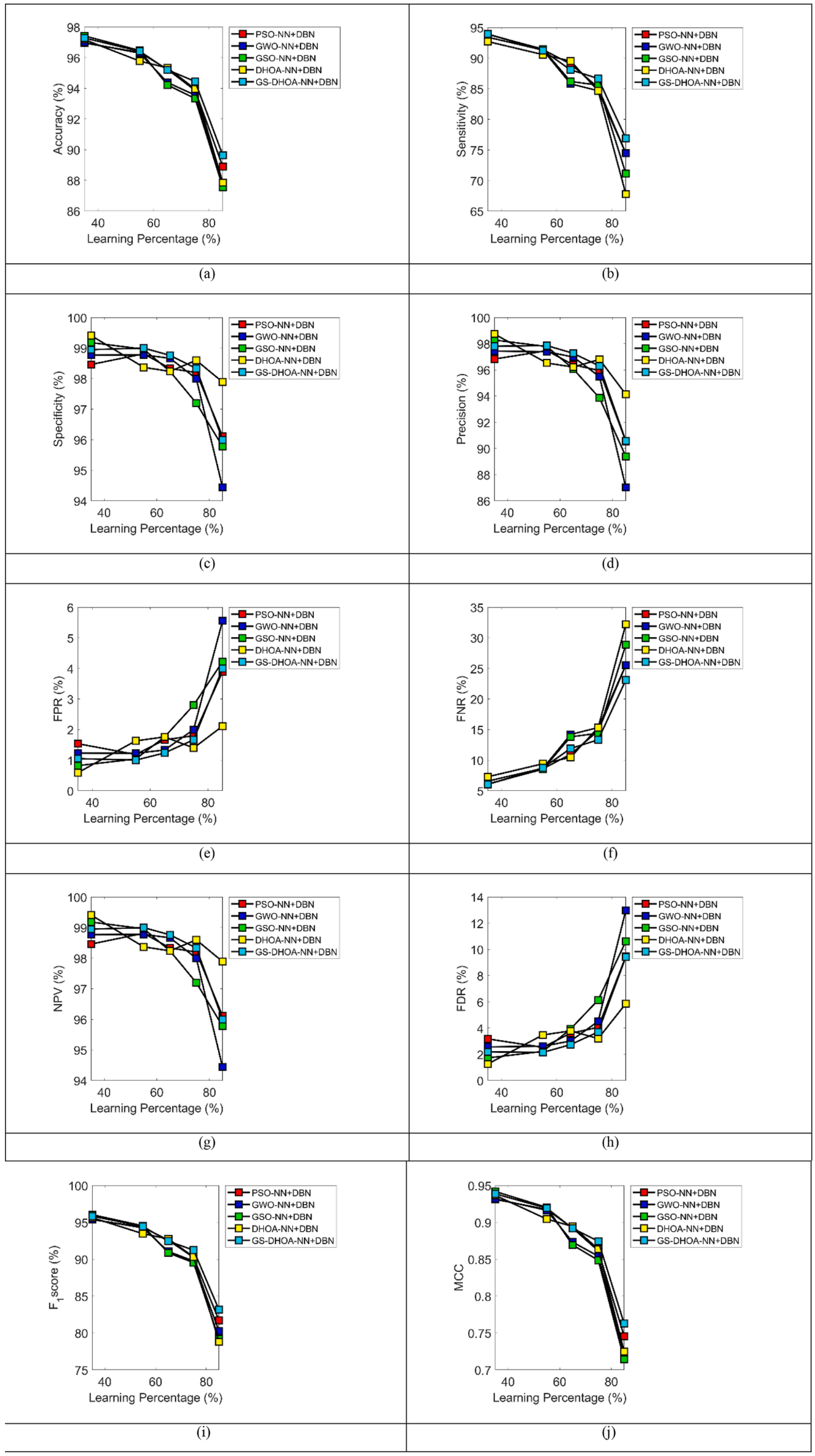

Fig. 5. Performance analysis of the proposed GS-DHOA-NN+DBN over diverse heuristic-based DBN+NN for network slicing in terms of performance measures such as, (a) Accuracy (b) Sensitivity (c) Specificity (d) Precision (e) FPR (f) FNR (g) NPV (h) FDR (i) F1-score (j) MCC

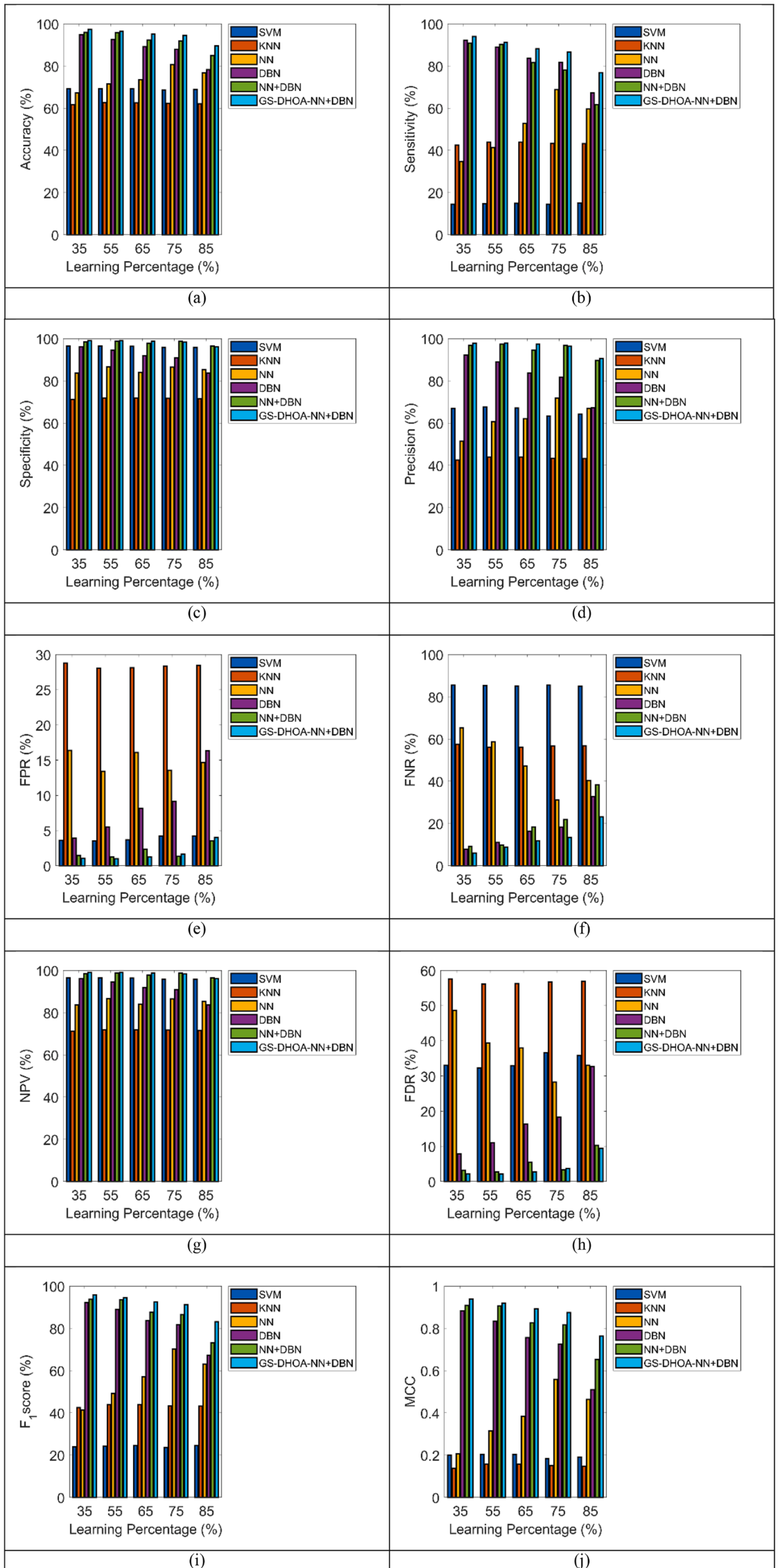

Fig. 6. Performance analysis of the proposed GS-DHOA-NN+DBN over various machine learning algorithms for network slicing in terms of performance measures such as, (a) Accuracy (b) Sensitivity (c) Specificity (d) Precision (e) FPR (f) FNR (g) NPV (h) FDR (i) F1-score (j) MCC

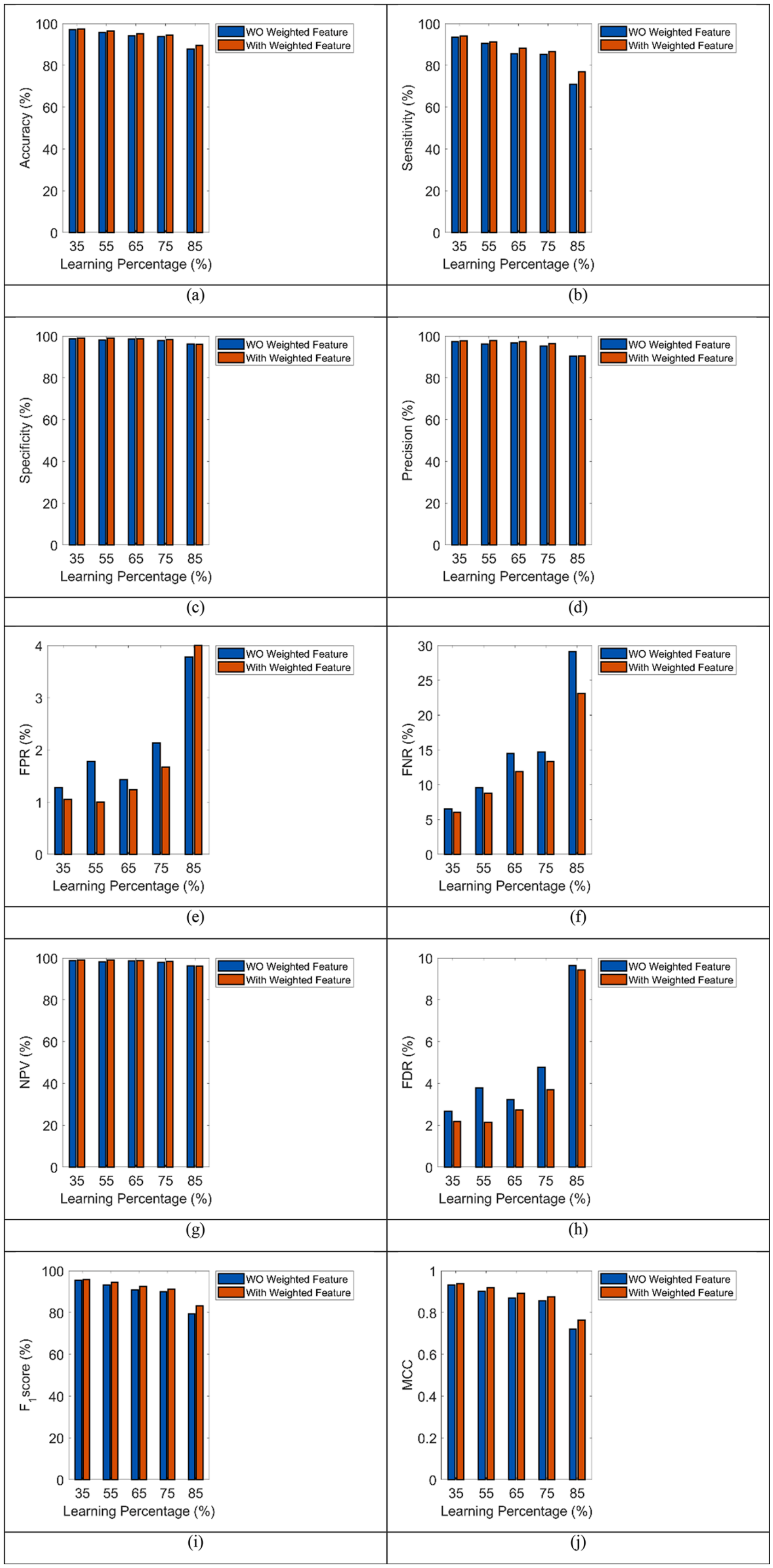

Fig. 7. Effect of OWFE of the proposed GS-DHOA-NN+DBN with and without weighted features in terms of performance measures such as, (a) Accuracy (b) Sensitivity (c) Specificity (d) Precision (e) FPR (f) FNR (g) NPV (h) FDR (i) F1-score (j) MCC

Table 2
Overall performance analysis of the proposed GS-DHOA-NN+DBN with various existing optimization algorithms

Performance measures	PSO-NN+DBN [13]	GWO-NN+DBN [14]	GSO-NN+DBN [51]	DHOA-NN+DBN [50]	GS-DHOA-NN+DBN
Accuracy	0.93867	0.93556	0.93333	0.93956	0.94444
Sensitivity	0.852	0.84667	0.856	0.84667	0.86667
Specificity	0.982	0.98	0.972	0.986	0.98333
Precision	0.95946	0.95489	0.9386	0.96799	0.96296
FPR	0.018	0.02	0.028	0.014	0.016667
FNR	0.148	0.15333	0.144	0.15333	0.13333
NPV	0.982	0.98	0.972	0.986	0.98333
FDR	0.040541	0.045113	0.061404	0.032012	0.037037
F1_score	0.90254	0.89753	0.8954	0.90327	0.91228
MCC	0.86125	0.85405	0.84856	0.86368	0.87439

such as accuracy, sensitivity, specificity, precision, FPR, FNR, NPV, FDR, F1-score, and MCC.

7.2. Performance analysis over diverse heuristic-based DBN+NN

Fig. 5 shows the performance of the proposed and conventional meta-heuristic-based NN+DBN in network slicing. We compared the proposed GS-DHOA with several optimization algorithms, such as PSO, GWO, GSO, and DHOA in terms of various performance measures, such as accuracy, sensitivity, specificity, precision, FPR, FNR, FDR, NPV, F1 Score, and MCC; and, computed the results. It can be seen from the analysis that the proposed GS-DHOA performs better in terms of all performance measures when it is compared with several other optimization algorithms. From Fig. 5(a), at 80% learning percentage, the accuracy of the proposed GS-DHOA is 1.81, 5,65, 2.16, and 2.05% advanced than PSO-NN+DBN, GWO-NN+DBN, GSO-NN+DBN, and DHOA-NN+DBN, respectively. Similarly, it has the best performance in terms of other performance measures. Thus, we confirmed that the proposed GS-DHOA-NN+DBN performs better in terms of various performance analysis when it is compared with the existing optimization-based NN+DBN algorithms.

7.3. Performance analysis over various machine learning algorithms

We compared the performance of the proposed GS-DHOA-NN+DBN with several conventional ML algorithms, such as SVM, KNN, NN, DBN, and NN+DBN; and Fig. 6 shows the computed results. We compared the proposed GS-DHOA with several ML algorithms, such as SVM, KNN, NN, DBN, and NN+DBN in terms of several performance measures, such as accuracy, sensitivity, specificity, precision, FPR, FNR, FDR, NPV, F1 Score, and MCC; and, computed the results. We can see from the results that the proposed GS-DHOA performs well in terms of all performance measures when compared with the various ML algorithms. In Fig. 6(a), at 85% learning percentage, the accuracy of the proposed GS-DHOA-NN+DBN is 32.35, 45.16, 18.42, 15.38, and 2.27% better than SVM, KNN, NN, DBN, and NN+DBN, respectively. Similarly, in terms of the other measures, the proposed network slicing also outperforms the conventional ML models.

Algorithm 1
Conventional DHOA [50].

Input: Initialize the population as q
Output: Initial best solution as $q^{lead}$and next-best solution as $q^{successor}$
Begin
while ($t < t_{\max}$)
for every solution present in the population
Compute the fitness of every solution
Update $p,\ m,\ G,\ a,\ C,\ T,\ s$
if ($a < 1$)
if ($|T| \geq 1$)
Update the solution using Eq. (21)
else
Update the solution using Eq. (28)
end if
Else
Update the solution using Eq. (27)
end if
end for
Compute the fitness of every solution
Update the leader position as $q^{lead}$and the successor position as $q^{successor}$
$t = t + 1$
end while
Return the position of the leader as q^{lead}
Terminate

Algorithm 2
Conventional GSO [51].

Initialize the count of dimensions as n
Initialize the count of glowworms as o
Assume ssas the step size
Assume $y_j(ti)$as the position of the glowworm $j$at time ti
Locate the agents in a random manner
for $j = 1 \quad to \quad o$ do
$lu_j(0) = lu_0 s_e^j(0) = s_p$
Set the maximum iteration count as $t_{\max}$
while ($t < t_{\max}$)do {
for each glowworm $j$do
Perform the luciferin update phase as in Eq. (29)
for each glowworm $j$do
Perform the movement phase as in Eq. (31)
for every glowworm $k \in O_j(t)$do
Calculate the probability $q_{jk}(t)$as in Eq. (30)
end for
end for
end for
end for

7.4. Effect of OWFE

Fig. 7 shows the effect on OWFE on the 5G network slicing. We compared the performance of the proposed GS-DHOA based on “with weighted feature” and “without weighted feature,” in terms of several

Table 3
Overall performance analysis of the proposed GS-DHOA-NN+DBN with various existing optimization algorithms

Performance measures	SVM [54]	KNN [55]	NN [46]	DBN [47]	NN+DBN [46, 47]	GS-DHOA-NN+DBN
Accuracy	0.68711	0.62222	0.80578	0.87822	0.91822	0.94444
Sensitivity	0.14533	0.43333	0.688	0.81733	0.78133	0.86667
Specificity	0.958	0.71667	0.86467	0.90867	0.98667	0.98333
Precision	0.63372	0.43333	0.71766	0.81733	0.967	0.96296
FPR	0.042	0.28333	0.13533	0.091333	0.013333	0.016667
FNR	0.85467	0.56667	0.312	0.18267	0.21867	0.13333
NPV	0.958	0.71667	0.86467	0.90867	0.98667	0.98333
FDR	0.36628	0.56667	0.28234	0.18267	0.033003	0.037037
F1_score	0.23644	0.43333	0.70252	0.81733	0.86431	0.91228
MCC	0.18333	0.15	0.55871	0.726	0.81611	0.87439

Algorithm 3
Proposed GS-DHOA.

```
Input: Initialize the population as q
Output: Initial best solution as q^lead and next-best solution as q^successor
Begin
  while (t < t_max)
    for every solution present in the population
    Compute the fitness of every solution
    Update p, m, G, a, C, T, s
      if (a < 1)
        if (|T| ≥ 1)
          Update the solution using Eq. (21)
          Else
          Update the solution using Eq. (28)
        end if
      Else
      if (|T| < 1)
        Update the solution using Eq. (27)
      else
        Update the solution by means of GSO using Eq. (31)
      end if
    end for
    Compute the fitness of every solution
    Update the leader position as q^lead and the successor position as q^successor
    t = t + 1
  end while
  Return the position of the leader as q^lead
Terminate
```

performance measures, such as accuracy, sensitivity, specificity, precision, FPR, FNR, FDR, NPV, F1 Score, and MCC; and, computed the results. We can see from the outcomes that the proposed GS-DHOA performs better OWFE with respect to all the performance measures. Considering Fig. 7(a), at 85% learning percentage, the accuracy of the proposed GS-DHOA-NN+DBN with weighted feature is 2.27% advanced than that without weighted feature. We observed the same enhanced performance for other measures. Hence, we can conclude that the proposed GS-DHOA-NN+DBN with weighted feature performs better than that without weighted feature.

7.5. Overall performance analysis

We compared the overall performance analysis of the proposed GS-DHOA-NN+DBN with several conventional optimization algorithms, such as PSO-NN+DBN, GWO-NN+DBN, GSO-NN+DBN, and DHOA-NN+DBN, as well as ML algorithms, such as SVM, KNN, NN, DBN, and NN+DBN; and, Table 2 and Table 3 summarize the results. We compared the performance of the proposed GS-DHOA with several other optimization algorithms and ML algorithms in terms of various performance measures, such as accuracy, sensitivity, specificity, precision, FPR, FDR, FNR, NPV, F1 Score, and MCC. We can see from the tables that the proposed GS-DHOA has better performance analysis with respect to various performance measures. From Table II, the accuracy of the proposed GS-DHOA-NN+DBN is 0.61, 0.95, 1.19, and 0.52% superior to PSO-NN+DBN, GWO-NN+DBN, GSO-NN+DBN, and DHOA-NN+DBN, respectively. From Table 3, the accuracy of the proposed GS-DHOA-NN+DBN is 37.45, 51.79, 17.21, 7.54, and 2.86% improved than SVM, KNN, NN, DBN, and NN+DBN, respectively. Hence, the proposed GS-DHOA-NN+DBN performs better in terms of overall analysis, compared with the existing optimization and ML algorithms.

8. Conclusions

Using a hybrid learning algorithm, we proposed an efficient 5G network slicing model, which included three main phases: (a) Data collection, (b) OWFE, and (c) Slicing classification. First, we collected a dataset relating to the 5G network slicing. The dataset included attributes linked with various network devices such as “user device type, duration, packet loss ratio, packet delay budget, bandwidth, delay rate, speed, jitter, modulation type.” Next, we carried out the OWFE, in which we multiplied a weight function with the attribute values; thereby, having high scale variation. We optimized this weight function with the help of hybridizing two meta-heuristic algorithms such as GSO and DHOA; and, we name the proposed algorithm GS-DHOA. We classified the exact network slices such as, “eMBB, mMTC, and URLLC” for each device for the given attributes. This classification was by the hybrid classifier with the help of DBN and NN. In this work, we optimized the weight function of both networks using the hybrid GS-DHOA. The analytical results and simulation models revealed that the proposed model could provide accurate 5G network slicing. Considering the overall analysis, the accuracy of the proposed GS-DHOA-NN+DBN was 0.61, 0.95, 1.19, and 0.52% superior to PSO-NN+DBN, GWO-NN+DBN, GSO-NN+DBN, and DHOA-NN+DBN, respectively. Hence, we could conclude that the proposed GS-DHOA-NN+DBN-based network slicing was efficient for 5G network slicing. Though the developed method needs to be further improved to solve more complex problems.

In future, the developed algorithm will be tested for other types of datasets. Moreover, the parameters of the algorithm will be tuned further. Also for real time analytics and to reduce the latency [56] for processing huge amount of data in the cloud, edge computing can be considered to implement the proposed model [57,58].

CRediT authorship contribution statement

Mustufa Haider Abidi: Conceptualization, Methodology, Software. **Hisham Alkhalefah:** Data curation, Writing - original draft. **Khaja Moiduddin:** Visualization, Investigation. **Mamoun Alazab:** Supervision, Writing - review & editing. **Muneer Khan Mohammed:** Software, Validation. **Wadea Ameen:** Writing - review & editing. **Thippa Reddy Gadekallu:** Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through Research Group no. RG-1441-349.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.csi.2021.103518.

References

[1] W. Guan, X. Wen, L. Wang, Z. Lu, Y. Shen, A Service-Oriented Deployment Policy of End-to-End Network Slicing Based on Complex Network Theory, 6, IEEE Access, 2018, pp. 19691–19701.
[2] M. Iwamura, NGMN View on 5G Architecture, in: 2015 IEEE 81st Vehicular Technology Conference, VTC Spring, 2015, pp. 1–5.
[3] I. Budhiraja, S. Tyagi, S. Tanwar, N. Kumar, M. Guizani, CR-NOMA Based Interference Mitigation Scheme for 5G Femtocells Users, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6.
[4] A.A. Barakabitze, A. Ahmad, R. Mijumbi, A. Hines, 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges, Computer Networks 167 (2020), 106984.
[5] K. Samdanis, X. Costa-Perez, V. Sciancalepore, From network sharing to multi-tenancy: The 5G network slice broker, IEEE Communications Magazine 54 (2016) 32–39.
[6] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, H. Flinck, Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions, IEEE Communications Surveys & Tutorials 20 (2018) 2429–2453.

[7] R. Wen, G. Feng, J. Tang, T.Q.S. Quek, G. Wang, W. Tan, S. Qin, On Robustness of Network Slicing for Next-Generation Mobile Networks, IEEE Transactions on Communications 67 (2019) 430–444.

[8] I. Budhiraja, S. Tyagi, S. Tanwar, N. Kumar, J.J.P.C. Rodrigues, Tactile Internet for Smart Communities in 5G: An Insight for NOMA-Based Solutions, IEEE Transactions on Industrial Informatics 15 (2019) 3104–3112.

[9] R.R. Aditya, R.S. Rahul, G.M. Rishikesh, S.G. Rutuja, N.M. Parikshit, N.R. Poonam, Long-Term and Short-Term Traffic Forecasting Using Holt-Winters Method: A Comparability Approach with Comparable Data in Multiple Seasons, International Journal of Synthetic Emotions (IJSE) 8 (2017) 38–50.

[10] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, J. Rao, AI-Assisted Network-Slicing Based Next-Generation Wireless Networks, IEEE Open Journal of Vehicular Technology 1 (2020) 45–66.

[11] M. Vincenzi, E. Lopez-Aguilera, E. Garcia-Villegas, Maximizing Infrastructure Providers' Revenue Through Network Slicing in 5G, IEEE Access 7 (2019) 128283–128297.

[12] M.M. Beno, V.I. R, S.S. M, B.R. Rajakumar, Threshold prediction for segmenting tumour from brain MRI scans, International Journal of Imaging Systems and Technology 24 (2014) 129–137.

[13] M.E.H. Pedersen, A.J. Chipperfield, Simplifying Particle Swarm Optimization, Applied Soft Computing 10 (2010) 618–628.

[14] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Advances in Engineering Software 69 (2014) 46–61.

[15] G. Wang, G. Feng, S. Qin, R. Wen, S. Sun, Optimizing Network Slice Dimensioning via Resource Pricing, IEEE Access 7 (2019) 30331–30343.

[16] G. Wang, G. Feng, T.Q.S. Quek, S. Qin, R. Wen, W. Tan, Reconfiguration in Network Slicing—Optimizing the Profit and Performance, IEEE Transactions on Network and Service Management 16 (2019) 591–605.

[17] G.T. Reddy, M.P.K. Reddy, K. Lakshmanna, R. Kaluri, D.S. Rajput, G. Srivastava, T. Baker, Analysis of Dimensionality Reduction Techniques on Big Data, IEEE Access 8 (2020) 54776–54788.

[18] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, X. Costa-Pérez, A Machine Learning Approach to 5G Infrastructure Market Optimization, IEEE Transactions on Mobile Computing 19 (2020) 498–512.

[19] R. Li, Z. Zhao, Q. Sun, I. C, C. Yang, X. Chen, M. Zhao, H. Zhang, Deep Reinforcement Learning for Resource Management in Network Slicing, IEEE Access 6 (2018) 74429–74441.

[20] D. He, N. Kumar, J.-H. Lee, Privacy-preserving data aggregation scheme against internal attackers in smart grids, Wireless Networks 22 (2016) 491–502.

[21] R. Amin, S.K.H. Islam, G.P. Biswas, M.K. Khan, N. Kumar, An Efficient and Practical Smart Card Based Anonymity Preserving User Authentication Scheme for TMIS using Elliptic Curve Cryptography, Journal of Medical Systems 39 (2015) 180.

[22] Q. Pham, S. Mirjalili, N. Kumar, M. Alazab, W. Hwang, Whale Optimization Algorithm With Applications to Resource Allocation in Wireless Networks, IEEE Transactions on Vehicular Technology 69 (2020) 4285–4297.

[23] S. Challa, A.K. Das, V. Odelu, N. Kumar, S. Kumari, M.K. Khan, A.V. Vasilakos, An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks, Computers & Electrical Engineering 69 (2018) 534–554.

[24] R.S. Bali, N. Kumar, Secure clustering for efficient data dissemination in vehicular cyber–physical systems, Future Generation Computer Systems 56 (2016) 476–492.

[25] D. He, M. Ma, S. Zeadally, N. Kumar, K. Liang, Certificateless Public Key Authenticated Encryption With Keyword Search for Industrial Internet of Things, IEEE Transactions on Industrial Informatics 14 (2018) 3618–3627.

[26] A. Azab, R. Layton, M. Alazab, J. Oliver, Mining Malware to Detect Variants, in: 2014 Fifth Cybercrime and Trustworthy Computing ConferenceAuckland, New Zealand, 2014, pp. 44–53.

[27] M. Alazab, S. Khan, S.S.R. Krishnan, Q. Pham, M.P.K. Reddy, T.R. Gadekallu, A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid, IEEE Access 8 (2020) 85454–85463.

[28] T.R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P.K.R. Maddikunta, G. Srivastava, Deep neural networks to predict diabetic retinopathy, Journal of Ambient Intelligence and Humanized Computing (2020).

[29] D.A. Chekired, M.A. Togou, L. Khoukhi, A. Ksentini, 5G-Slicing-Enabled Scalable SDN Core Network: Toward an Ultra-Low Latency of Autonomous Driving Service, IEEE Journal on Selected Areas in Communications 37 (2019) 1769–1782.

[30] A.S.D. Alfoudi, S.H.S. Newaz, A. Otebolaku, G.M. Lee, R. Pereira, An Efficient Resource Management Mechanism for Network Slicing in a LTE Network, IEEE Access 7 (2019) 89441–89457.

[31] M.W.L. Moreira, J.J.P.C. Rodrigues, G.A.B. Marcondes, A.J.V. Neto, N. Kumar, I.D. L.T. Diez, A Preterm Birth Risk Prediction System for Mobile Health Applications Based on the Support Vector Machine Algorithm, in: 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1–5.

[32] Z. Liu, C. Cao, S. Ding, Z. Liu, T. Han, S. Liu, Towards Clinical Diagnosis: Automated Stroke Lesion Segmentation on Multi-Spectral MR Image Using Convolutional Neural Network, IEEE Access 6 (2018) 57006–57016.

[33] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, H. Zhang, Intelligent 5G: When Cellular Networks Meet Artificial Intelligence, 24, IEEE Wireless Communications, 2017, pp. 175–183.

[34] S. Bhattacharya, R.S. S, K.P. Maddikunta, R. Kaluri, S. Singh, R.T. Gadekallu, M. Alazab, U. Tariq, A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU, Electronics 9 (2020).

[35] J. Ni, L. Wu, X. Fan, S.X. Yang, Bioinspired Intelligent Algorithm and Its Applications for Mobile Robot Control: A Survey, Computational Intelligence and Neuroscience 2016 (2016), 3810903.

[36] A. Darwish, Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications, Future Computing and Informatics Journal 3 (2018) 231–246.

[37] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, H. Flinck, Coalitional Game for the Creation of Efficient Virtual Core Network Slices in 5G Mobile Systems, IEEE Journal on Selected Areas in Communications 36 (2018) 469–484.

[38] C. Song, M. Zhang, Y. Zhan, D. Wang, L. Guan, W. Liu, L. Zhang, S. Xu, Hierarchical Edge Cloud Enabling Network Slicing for 5G Optical Fronthaul, J. Opt. Commun. Netw. 11 (2019) B60–B70.

[39] X. Li, C. Guo, L. Gupta, R. Jain, Efficient and Secure 5G Core Network Slice Provisioning Based on VIKOR Approach, IEEE Access 7 (2019) 150517–150529.

[40] V. Sciancalepore, X. Costa-Perez, A. Banchs, RL-NSB: Reinforcement Learning-Based 5G Network Slice Broker, IEEE/ACM Transactions on Networking 27 (2019) 1543–1557.

[41] H. Wang, Y. Wu, G. Min, J. Xu, P. Tang, Data-driven dynamic resource scheduling for network slicing: A Deep reinforcement learning approach, Information Sciences 498 (2019) 106–116.

[42] Q. Wang, J. Alcaraz-Calero, R. Ricart-Sanchez, M.B. Weiss, A. Gavras, N. Nikaein, X. Vasilakos, B. Giacomo, G. Pietro, M. Roddy, M. Healy, P. Walsh, T. Truong, Z. Bozakov, K. Koutsopoulos, P. Neves, C. Patachia-Sultanoiu, M. Iordache, E. Oproiu, I.G.B. Yahia, C. Angelo, C. Zotti, G. Celozzi, D. Morris, R. Figueiredo, D. Lorenz, S. Spadaro, G. Agapiou, A. Aleixo, C. Lomba, Enable Advanced QoS-Aware Network Slicing in 5G Networks for Slice-Based Media Use Cases, IEEE Transactions on Broadcasting 65 (2019) 444–453.

[43] S.A. AlQahtani, W.A. Alhomiqani, A multi-stage analysis of network slicing architecture for 5G mobile networks, Telecommunication Systems 73 (2020) 205–221.

[44] A. Thantharate, R. Paropkari, V. Walunj, C. Beard, DeepSlice: A Deep Learning Approach towards an Efficient and Reliable Network Slicing in 5G Networks, in: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019, pp. 0762–0767.

[45] T.R. Gadekallu, S.P. R.M, P. M, C.L. Chowdhary, P.K.R. M, S. Hakak, W.Z. Khan, A deep neural networks based model for uninterrupted marine environment monitoring, Computer Communications 157 (2020) 64–75.

[46] K. Gurney, An Introduction to Neural Networks, Taylor & Francis, 1997.

[47] A. Khan, A. Zameer, T. Jamal, A. Raza, Deep Belief Networks Based Feature Generation and Regression for Predicting Wind Power, arXiv (2018) 31.

[48] S. Garg, K. Kaur, N. Kumar, J.J.P.C. Rodrigues, Hybrid Deep-Learning-Based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective, IEEE Transactions on Multimedia 21 (2019) 566–578.

[49] A. Jindal, G.S. Aujla, N. Kumar, R. Chaudhary, M.S. Obaidat, I. You, SeDaTiVe: SDN-Enabled Deep Learning Architecture for Network Traffic Control in Vehicular Cyber-Physical Systems, IEEE Network 32 (2018) 66–73.

[50] G. Brammya, S. Praveena, N.S. Ninu Preetha, R. Ramya, B.R. Rajakumar, D. Binu, Deer Hunting Optimization Algorithm: A New Nature-Inspired Meta-heuristic Paradigm, The Computer Journal (2019).

[51] Y. Zhou, G. Zhou, Y. Wang, G. Zhao, A Glowworm Swarm Optimization Algorithm Based Tribes, Applied Mathematics & Information Sciences 7 (2013) 537–541.

[52] B.R. Rajakumar, Impact of static and adaptive mutation techniques on the performance of Genetic Algorithm, Int. J. Hybrid Intell. Syst. 10 (2013) 11–22.

[53] B.R. Rajakumar, Lion Algorithm and Its Applications, in: M. Khosravy, N. Gupta, N. Patel, T. Senjyu (Eds.), Frontier Applications of Nature Inspired Computation, Springer, Singapore, 2020, pp. 100–118.

[54] S. Yu, K.K. Tan, B.L. Sng, S. Li, A.T.H. Sia, Lumbar Ultrasound Image Feature Extraction and Classification with Support Vector Machine, Ultrasound in Medicine & Biology 41 (2015) 2677–2689.

[55] Y. Chen, X. Hu, W. Fan, L. Shen, Z. Zhang, X. Liu, J. Du, H. Li, Y. Chen, H. Li, Fast density peak clustering for large scale data based on kNN, Knowledge-Based Systems 187 (2020), 104824.

[56] S. Rajadurai, M. Alazab, N. Kumar, T.R. Gadekallu, Latency evaluation of SDFGs on heterogeneous processors using timed automata, IEEE Access 8 (2020) 140171–140180.

[57] M.B. Mollah, M.A.K. Azad, A Vasilakos, Security and privacy challenges in mobile cloud computing: Survey and way ahead, Journal of Network and Computer Applications 84 (2017) 38–54.

[58] M.B. Mollah, M.A.K. Azad, A Vasilakos, Secure data sharing and searching at the edge of cloud-assisted internet of things, IEEE Cloud Computing 4 (1) (2017) 34–42.