ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SOCIAL SKI DRIVER BASED EFFICIENT PARKING DYNAMICS AND COMPUTATION USING DEEP REINFORCEMENT LEARNING IN VEHICULAR CLOUD

SHIVANAND C. HIREMATH*1, JAYASHREE D. MALLAPUR2

*1Department of Electronics and Communication, R.N.Shetty Polytechnic College, Belagavi, India

2Department of Electronics and Communication, Basaveshwar Engineering College, Bagalkot, India.

*1Corresponding Email address: shivch612@gmail.com1, jdmec@becbgk.edu2

ABSTRACT

Modern urbanization is experiencing an increased number of autonomous vehicles day by day leading to traffic congestion, parking problems, and overcrowding at the parking lots are one of the major concerns that the smart cities are facing nowadays and due to floating prices of inflammable gases recently there is a drastic escalation of electric motor vehicles (EMV) on the roads needs Parking with charging facility. This research explores and resolves the issues of dynamic provisioning of parking spaces for EMV with or without charging facility in three phases, firstly using cluster-based searching for vacant parking slots, secondly using deep reinforcement learning (DRL) based user parking request processing using cloud server and distributed fog nodes with the assistance of Vehicular Adhoc networks (VANET) model and finally using evolutionary optimization based social ski driven routing (SSD) algorithm is used to efficiently route the EMV to the vacant parking lots. The fitness function is newly devised considering multi-objective parameters such as traffic density,battery power As well as distance parameter and shortest routing decision are made, Hence by using proposed cluster-based parking system using deep reinforcement learning-based Social Ski driven (DRL-SSD) routing protocol achieves allocation of parking slots with a high probability of success rate of 95% with minimal traffic density 7.5 per lane, minimal delay of 7.55min and minimal fitness value of 15.88.

Keywords: *VANET; EMV; DRL-SSD;RL-SSD;DRL-PSO*

1. INTRODUCTION

Rapidly growing infrastructure and modern urbanization are transforming towards the development of smart cities to enhance living standards and Humans' advanced Information & Communication Technology (ICT) lifestyles, as well as the research community, werein favor of creating each tactile scheme remotely available via the internet. With the emergence of the Internet of Things (IoT) mobile applications are used for monitoring the quality of air, soil, weather forecasting, etc, and by using cloud environment monitoring of real-time traffic overcrowding, automated traffic control, intelligent parking scheme, etc, have evolved these days. As a consequence, one might relocate to cloud computing to reap the benefits of intellectual cloud-assisted facilities, as well as assert the dynamic nature of provisioning assets shift from basic servers to hosted cloud-dependent solutions to the ability of edge devices through a dedicated internet web or mobile application/service. Recently a lot of research works have been proposedtowards smart parking management systems but still lacks dynamicity in finding the vacant parking slot to park the vehicles in an overcrowded metro city has become one of the major challenging issues to resolve and it has to provision the nearest and quickest parking space with the least latency under real-time traffic is required.

To fulfill that demand a Vehicular cloud model, a collaboration of VANET and cloud model should be considered in smart city development schemes, this research work is an extended work of cluster-dependent parking provisioning mechanism [1] to find the available vacant spaces among different clusters in a certain communication range. The Vehicular Cloud infrastructure is comprised of a massive pool of physical servers which can deliver effectively limitless network resources, storage, as well as computing to the aforementioned services, and it has been encouraged by the notion of deep reinforcement learning (DRL) premised on effective service dispersal in cloud and fog environments [2] and also the authors' previous contribution of Reinforcement learning (RL) approach [10].

Previous research studies began with a vast pool of centralized cloud servers that devoured substantial computation, backup, as well as system resources. Owing to their geographic regions, the cloud service provider, as well as the end-user, might have had a poor network delay experience. To address such concerns, this research approach deals with the collaboration of VANET and cloud computing infrastructure consisting of fog nodes deployed as fog servers in the VANET model, fog nodes consisting of a server equipped with minimum computational capabilities.

In the first part of this work users, parking requests are served by a cluster-based parking system to search the vacant parking slot and parking requests are processed using the deep reinforcement learning (DRL) approach to havingthe least latency and for each EMV or normal vehicle, fitness values are evaluated considering multi-objective parameters such as Status of the battery, distance and traffic density for each User request. Hence all arriving parking requests are searched in all the nearby parking clusters in a certain communication range and a user parking service request for normal Parking or parking with a charging facility is sent for processing to the fog nodespresent at the closest vicinity of the requested vehicle's position, However fog nodes serve multiple services requests such as monitoring of traffic, Weather and emergency services, etc if the fog node is unable to handle the requests due to lack of computing resources or service overload then it forwards it to the centralized vehicular cloud to process the request. Once the system finds the vacant parking slot it will be informed the user to book the slot and in the second part of this work assist users with the shortest distance route to the parking destination using the nature-inspired evolutionary algorithm Social Ski driven routing (SSD) protocol.

The remainder of the article would be arranged as follows. Section II mentions the motivation for this project as well as summarises our contributions. Section III offers a summary of the latest existing literature depending on the motivation as well as the issue at hand. In Section IV, The proposed research work cluster-based parking system for EMV is illustrated for dynamic provisioning of vacant parking slots by distributing and processing requests through DRL-SSD among fog nodes and centralized vehicular cloud environment, and SSD routing is formulated in subsection. Section V discusses the performance evaluation of processing of parking service requests and its success rate using DRL-SSD and compared with RL-SSD for cloud and fog nodes followed by DRL-SSD based optimized routing is evaluated and compared with DRL-PSO and Section VI mentions the concluding remarks as well as future works.

2. MOTIVATIONANDCONTRIBUTION

As previously discussed, the user's parking request is indeed the service that will be implemented in both the fog node as well as the centralized vehicular cloud server to deliver the user's parking request with the least latency. It is first deployed in the nearest fog nodes to the EMV location to find the vacant parking slot at the earliest from the cluster-based parking if the fog node is overloaded with other services then it forwards the request to the main vehicular cloud server environment as fog nodes are used for multi-services such as emergency healthcare service, traffic prediction, etc. . Furthermore, in terms of meeting QoS, the demanded user cannot be allotted straightforwardly to the vehicular cloud because doing so may boost network latency. On the other hand, if the closest fog node to the vehicle does not have enough available resources to calculate the request at any given time, this same parking service request would be conveyed to a vehicular cloud. Predicated on this assertion, the handling of parking service requests among some of the closest fog nodes as well as vehicular cloud is becoming a daunting task for the service provider. Hence A novel cluster-based parking provisioning approach for smart city development schemes is shown in Figure 1

The above cluster-based parking scenario motivates us to explore the issue of finding vacant parking slots with or without charging facilities among the clusters present near the well-known landmarks in metro cities, further if vacant parking slots are available in multiple clusters by applying a meta-heuristic social ski driven routing algorithm to select theefficient shortest route among the clusters having empty slots the user is suggested to book and the nearest and shortest route for the parking lot is assisted through google map and this parking service request computation process will be handled by a futuristic deep reinforcement learning (DRL) approach.

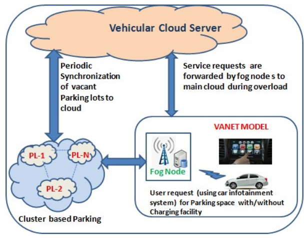

Figure 1: Cluster-based Parking

2.1. Contributions

In the context outlined above, our research contributions to this work could be presented as follows:

- The user's parking request for searching parking spaces in different clusters for parking with charging and parking without charging facility is processed among the nearest fog nodes and vehicular cloud server a problem has been formulated, which offers a comprehensive goal for this paper.
- To attain the aforementioned goal, an ingenious DRL- SSD reliant strategy is being used, which forwards parking service requests from EMV to closest fog nodes as well as a vehicular cloud server for identification of vacant parking lots in clusters and thus is compared with the reinforcement learning (RL) approach to obtain better efficiency.
- The performance of the proposed model is then evaluated for the percentage of the Success rate of allocation of parking spaces and then applying an optimized DRL-SSD based shortest routing algorithm to reach the parking destination, DRL-SSD performance is further evaluated using a newly devised fitness function and compared with another stochastic optimization-based algorithm DRL-PSO for QoS parameters such as analysis of traffic density, delay, and fitness function values for EMV.

3. RELATEDWORKS

Throughout this section, a descriptive extensive research on the process of delivering parking service requests in fog nodes as well as cloud computing environments has been offered. Reinforcement Learning (RL), a branch of AI, has been thoroughly researched in so many fields of research, such as deep reinforcemelInt predicated Intelligent transportation systems[4], Deeplearning-basedautomationofsmarthomes [5],Smartdomesticarchitectures[6],Smartaccounting [7],DRLbasedframework for Auto driving systems are discussed[8][9] Reinforcementlearning-basedimplementationdiscussespersonalizeddelivery ofservicesinthe cloudenvironmentandfognodes[6],AmodeledElectricVehicleRoutingProblem,andits mathematical model was proposed for charging Electricvehicles(EV),B. Chatterjee et al. [10] suggested a meta-heuristic methodology for selecting features called Social Ski Driver (SSD) optimization, which could be used as a pre-processing tool for minimizing the dimensionality as well as remove unnecessary characteristics in machine learning and maybe data mining algorithms.

D. Niyato et al. [11] presented an offloading methodology to calculate the workloads of end devices from the perspective of numerous cloud service providers. The basic objective of the thesis is to strengthen the quality of service (QoS). Using the same environment as well as the Vehicular cloud network as an implementation [12], the authors of [13] developed a scheme for computing & system resources for cloud as well as fog node resources that uses dynamic orchestration.

M Karthi et al. [14] discuss reservation-dependent Smart parking accessibility by constantly evaluating availability in a single parking lot at a chosen place by proficiently calculating the count of free slots utilizing electronic sensors as well as Arduino kits to interact with the cloud system. Regrettably, it will not take into account contacting numerous parking lots inside a given target place for availability. The real issue of traffic congestion appears unlikely that parking spaces will be available. Sowmiya M et al. [15] discuss an intelligent parking scheme predicated on vehicular cloud computing but also recommend a scene in which all vehicles communicate with RSUs, which would then be synced with the cloud, as well as an android app affirms the free parking slots in such a parking lot. Even so, this model only works for solitary parking lots & appears to lack the dynamic nature of supplying the availability of parking resources before the hand in a place with greater probability. S Valipour et al.[16] illustrate a Field programmable gate arrays (FPGA) premised intelligent parking managerial system comprises a camera, database management, as well as user interaction. The camera has been used to acquire available parking slots, and also the pictures were also handled utilizing deep neural convolution network systems to determine vacant spaces. The above technology can reduce system financing as well as maintenance expenses, but it does not offer parking for real-time traffic scenarios. E.Katz et al. [17] illustrate in-depth an approach predicated on image processing for recognizing vacant parking spaces; it utilizes cameras at diverse viewpoints which are already attached to the car and also calculate the count of treads to assist the driver to park with just a speed sensor; however, this idea is hard to execute rationally in real life, culminating in exorbitant costs. Rishi Gupta et al. [18] mention the lack of supply of parking spaces as well as establish a system utilizing ultrasonic sensors & rapid reaction codes for backing up associated with reserving information at the entrance of parking spots depending on distance quantification. This seems to be constrained to establishing recourses at one place and system shortfalls in seeking solutions throughout traffic overcrowding demand expansions to communicate as well as lookup numerous parking spaces to create adaptability to manage parking issues.

Joshi J et al. [19] integrate a parking scheme depending on radio-based road congestion prognostication via incoming parking requests for vacant slots; this system utilizes infrared identification to recognize unutilized slots. Also, it employs a trip modeling approach as well as seasonal information to anticipate traffic flows, and then it implements digital keys for parking confirmation to optimize the effectiveness of the parking scheme. It employs a Smart Trip Modelling Method, and also because the count of vehicles on the road was indeed growing by the day, it is unreasonable to expect dynamic traffic at any spot in metro cities. Surekha et al. [20] utilizes Dijkstra's methodology predicated on least expense by assessing the shortest route but also recommends optimal way with least time to attain between demanded user place as well as parking place and enhances the waiting period for real-time traffic crowding, & describes primarily only about offering shortest way among user as well as parking spot when available, but it's crucial to increase the work as well as seek optimal solutions for obtaining a parking spot with such an elevated likelihood of winning via cloud-assisted procurement of aimed clusters in real-time dynamic traffic flows. Dongshu Wang et al. [21] provides a concise summary of the Particle swarm optimization algorithm discussed in detail for searching the space and due to its strong distribution ability and excellent robustness, it can converge quickly to optimization and hybridizes easily with other algorithms for performance analysis. Tharwat et al. [22] announced Social Ski driven (SSD), an evolved optimization technique able to detect near-optimal values of SVM classifier parameters. T.N. Pham et al. [23] suggested a cloud-based parking strategy to enhance effectiveness by lowering the count of consumers who fail to discover a free parking space, lowering the cost of moving to the parking spot, but does not focus on managing parking requests. Pablo et al. [24] recommended an effective charging control system for Electric Vehicles with charging scheduling as well as integrates a communication approach that relies on VANETs by contemplating geographical-routing guidelines for centralized prognostication of charging slot booking and deems traffic levels for electric vehicles for each trip there at chosen CS creating a traffic-aware CS selection scheme and yet lacks dynamic managing of requestsand Bandi et.al[25] proposed methodology uses the Artificial Neural Networks and uses Mask R-CNN a segmentation model based on Neural Network Architecture using Deep learning for finding the vacant space by annotating the data using captured images from the parking lots but lacks the dynamic allocations of parking slots with high success rates. Zhang et.al[26] use a meta-heuristics-based Ant colony optimization (ACO) approach for routing and computation of energy for EV that attempts to reduce the consumption of energy of EV, Nevertheless, the method did not enlist the additional factors like partial and time window constraints.

4. PROPOSEDSYSTEM

In crowded cities of metropolitan areas and modernization in automobile industries leading to electric motorvehiclesontheroadsrequireschargeinginfrastructuresonepaperbacksolutionforthemisprovidingparkingwitha chargingfacility and similarly, numerous automated vehicles having arrived from those other areas encounter obstacles parking their automobiles, which might also lessen their likelihood of heading out due to parking restrictions. Another remarkable problem would be when they arrive at their destination yet do not find an appropriate parking place, which might also result in a serious matter of staying in heavy traffic as well as searching for a parking spot. One quick-witted alternative to meet such a demand is to implement cluster-dependent vehicular cloud-assisted dynamic parking slot procurement throughout every famous landmark in a major metropolis with a betterprobabilityofsuccessfulallocation.

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

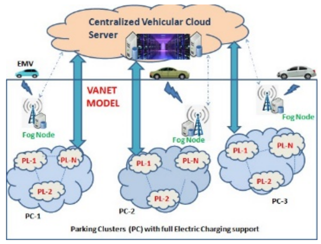

Figure 2: Proposed Cluster-based Parking System

The proposed research work shown in Figure-2 is composed of a collaboration of a centralized vehicular cloud comprised of computing devices as well as a database server. that stores the total number of occupied and vacant slots of each cluster, A Parking cluster (PC) is a group of parking lots Where parking cluster one (PC-1) contains 3 parking lots (PL), hence PL- 1,PL-2, and PL-3 be the parking lots each consisting of100 parking slots equipped with electric charging facility and also provides normal parking facility if the user opts for parking without charging facility if maximum vacant slots are available, Always priority is given for user requesting for parking with charging facility. A total of 300 slots per cluster for parking are available for users Similarly considering PC- 2 and PC-3 each with 300 slots a total of 900 parking slots. Parking clusters are distributed around landmark places within one square kilometer area considered and assumed as the smart city development authorities to manage and maintain all parking clusters by sending available vacant slots from every cluster after every transaction of allocation and deallocation of vacant slots periodically from each parking lots of every cluster to a centralized vehicular cloud server and the nearest fog nodes as well. Whenever the user initiates the parking service request, the nearest fog node to the Electric motor vehicle (EMV) accepts the request computes the service of parking request with charging and without charging facility using a cluster-based system if the fog node is running out of resources to handle the request itthen forwards the request to centralized vehicular cloud server leading to two types of latencies (a) Computational latency (b) Communication latency the performance of these latencies are analyzed using deep reinforcement learning (DRL) method and further once the parking service request is processed booking of the vacant slot will be done and using meta-heuristic DRL based social ski driven routing (DRL-SSD) algorithm a shortest route to the parking lot by assisting with the help of google map to the user interface.

4.1. System flow

A revolutionary approach towards futuristic transportationconsisting of dynamically provisioning the parking space withchargingorwithoutchargingfacilityParticularlyinmetrocities the most challenging issue is to park the vehicles atcrowdedlandmarks,ToovercomesuchanissueaClusterbased parking system is shown in Figure-3, The EMV userinitiates the Parking service request when they arrive nearbytheirdestination,theparkingservicerequestarediffusedtonearbyfognodeswhichstoresandkeepupdatesitselfevery vehicle IN/OUT transaction from parking lots at nearbyparking clusters to it, If the fog nodes are free to serve then ithandles the request else redirects it to the centralized vehicularcloud environment and if fog nodes finds the availability ofvacantslotsayEx:PC1providestheavailabilityofvacantslot then it directly ask the user to book a vacant slot withconfirming parking with or without charging facility, Usersrequesting parking with charging facility has the top priorityoveruserrequestingforjustparking,elsethesystemwillfurther keep checking in PC2 if not then in PC3 and looprepeats, once the vacant parking slot is reserved then systemwill assist the user to parking slot by applying Social Skidriven (SSD) shortest routing algorithm through google map.The parkingservicerequestishandledbydeepreinforcementlearningmechanismbydistributingtheparkingrequeststofognodesanda centralizedvehicularcloudenvironment.

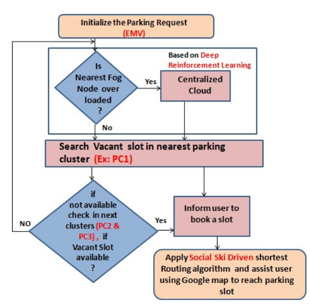

Figure 3: System flow of Cluster-based Parking System

4.2. Modeling Service

ThefognodeoffersasetofNnumberofservices,denotedby $\delta = \delta_1, \delta_2, \delta_3, ..., \delta_n$where each service is representedas $\delta_k, 0 < k \leq N$ Nearest fog node is responsible for acceptinguser Parking service requests followed by the distributionamong fog nodes and the main vehicular cloud server.

A cluster-dependent parking scheme takes into account the sensitivity of a service δk,which would be signified by ζk, 0 < ζk1. It's also presumed that the fog node determines the sensitivity of the services. A higher k-value indicates a more sensitive service. The investigation of the consumer parking request service, for example, seems to have a greater sensitive value than some other services such as video surveillance.

4.3. User Parking service request

A fog node provides a collection of services to every user attached toa cluster-based parking system in a vehicular cloudsystem.Let,P=P_1, P_2, P_3, ..., $P_n$betheset of allconnected users requesting parking. Each user is denotedby P_i, 0 <in. The Boolean variable $P_i\delta_k$ indicates if the user P_i is availing of the parking requestservice. Mathematically,

$$\mathrm{P_i}\,(\delta_k) = \begin{cases} 1, if\ user\ P_i\ is\ availing \\ \quad parking\ request \\ 0, Ot\square erwise \end{cases} \quad (1)$$

Distance-based priority: A requested consumer would be regarded to be at a specific distance from the fog node within the fog node's maximum range. Each consumer, as well as the fog node, has been assigned a location parameter. The Euclidean technique could be used to evaluate a user's distance, as shown below.

$$D_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (2)$$

Where (xi, Yi) seems to be the user Pi's location coordinate as well as (x, y) is indeed the neighboring fog node's location coordinate.That is obvious that perhaps the user's distance would be less than the peak value of the fog node, so the distance-dependent preference could be estimated as follows, predicated on the maximum range as Dmax well as the user's distance Di

$$\zeta_i^d = \frac{D_i}{D_{max}}, \forall P_i \in D_{max} \quad (3)$$

These are abundantly utilized in healthcare as well as forensic picture analysis for recognizing people in video clips, assessing medical images for ailment prognostication, and so on.

4.4. Computation using Deep Reinforcement Learning (DRL) Approach

Deep reinforcement learning (DRL) is indeed a subdiscipline of machine learning as well as neural networks in which smart machines could indeed learn from their activities in the same way that humans do. The actuality that an agent is praised or punished depending on their activities seems to be innate in this sort of machine learning. The features are extricated layer by layer. This research offers DRL-dependent fog node computing with a social ski-driven routing protocol, which leverages the Deep reinforcement learning methodology in handling EMV user service requests. Keras' sequential structure is primarily comprised of 3 layers: i) an input layer with nodes identical to the count of states, ii) a hidden layer with twice the count of nodes in the input layer, as well as iii) an output layer with the identical count of nodes as the input layer. In the hidden layer, let γ be the input passed to 1 − a sigmoid function. This Sigmoid function provides the output in the range of (0, 1). As a consequence, in the output layer, A linear activation function is being utilized, and the input values are unaltered. The Markov Decision Process has been used to numerically simulate the user parking request. As seen below, this is depicted as four states: state space, action space, state transition probability function, as well as a reward function.

4.5. A comparison between RL as well as Deep RL techniques

In the RL strategy, the notion of deep reinforcement learning (DRL) has been implemented in the agent. In contrast to the Q-learning technique, in which a Q function is being used to ascertain the motion of the agent, the agent anticipates an activity by employing the prognostic value of all activities, as seen in Figure 4. As a consequence, rather than the state as well as a specific activity as input, the present state could be provided to the agent, and also the agent identifies which action to accept utilizing the deep learning approach. To put deep reinforcement learning into practice The Keras framework was being used. Framework Keras: That is a Python deep learning API constructed on TensorFlow 2.0 as well as composed of famous open-source software libraries and a huge count of Python-dependent tools for evaluating complicated mathematical expressions. These tools are predominantly utilized in machine learning applications including artificial neural networks. The libraries have grown large enough to manage the vast bulk of the assessment & experiments associated with both scientific research as well as software platforms. On top of these frameworks/libraries, Keras [27] provides an interface for easy and fast prototyping of reinforced neural network-based applications that assist complexity as well as recurrent networks. This is predominantly configured to conserve intelligence's scalability as well as adaptability [28].To developthe Kerasframework,Pythonastheprogramminglanguageisusedforbetterconformabilitywith TensorFlow.DuringtheimplementationofDRL,theframeworkhelpedustofocuson the construction of the sequence model, which eventuallygrasps what action should be taken from each state.

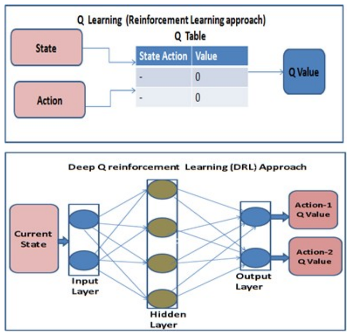

Figure 4: Comparision Between RL as well as Deep RL techniques

4.6. Computation using Deep Reinforcement Learning (DRL)Approach

Deep reinforcement learning (DRL) is indeed a subdiscipline of machine learning as well as neural networks in which smart machines could indeed learn from their activities in the same way that humans do. The actuality that an agent is praised or punished depending on their activities seems to be innate in this sort of machine learning. The features are extricated layer by layer. This research offers DRL-dependent fog node computing with a social ski-driven routing protocol, which leverages the Deep reinforcement learning methodology in handling EMV user service requests.Keras' sequential structure is primarily comprised of 3 layers: i) an input layer with nodes identical to the count of states, ii) a hidden layer with twice the count of nodes in the input layer, as well as iii) an output layer with the identical count of nodes as the input layer. In the hiddenlayer, let γ be the input passed to[1]–a sigmoid function.ThisSigmoidfunctionprovidestheoutputintherangeof(0,1). As a consequence, in the output layer, A linear activation function is being utilized, and the input values are unaltered. The Markov Decision Process has been used to numerically simulate the user parking request. As seen below, this is depicted as four states: state space, action space, state transition probability function, as well as a reward function.

State-space: State-space can be defined as per figure -4, Let$P^* = P_1^*, P_2^*, P_3^*,$ ...bethestatespaceandstateof user parking service request $\delta_i$from user $P_i$is denoted by $P_i^*$is a vector state ofall the user parking requests and let the$P_i^*{}_{,j}$indicatestheassignmentofuserparkingservice request to the fog node and centralized cloud Mathematically is given

By

$$P^*_{i,j} = \begin{cases} (0,0), \text{Request is not assigned} \\ (0,1), \text{Cloud is chosen to} \\ \qquad \text{process the request} \\ (1,0), \text{Fog node is chosen to} \\ \qquad \text{process the request} \\ (1,0), \text{Invalid State} \end{cases} \tag{4}$$

Let F P ∗ and C P ∗ be two boolean variables, representing the parking service request allocated to the fog node as well as cloud context, correspondingly. If the parking service gets assigned to a specific environment, the number 1 is being used; otherwise, the value 0 will be used. The preceding criterion must be met for a legitimate allocation.

$$F(P^*_{i,j}) + C(P^*_{i,j}) = 1 \tag{5}$$

The two fake states were added to the existing state space in addition to the current state space, oneofthemisnotassignedtoanycomputingenvironmentand the otherisassignedtooneofthe computing environments, Further, all the states of parkingservice requests were allowed to either fog nodes or centralizedcloudcomputingenvironment.

- *Actionstate:*The action space encompasses all of the probable motions of the agent out of its present state. The symbol φ specifies the agent's activity space. The present activity in the action space has been indicated by φ*. In the current implementation, an action might be seen as going to a new state. In this strategy, the agent travels to the bulk of other states. To restrict the action space, numerous restrictions were imposed, including the agent being forbidden to travel from the first fake state to the final dummy state.

- *Rewards:* The agent obtains a reward value from the environment by executing a precise activity in the present state. The reward value designates the action's value in the present state. As per the situation, a positive reward value drives the agent to perform actions. Nevertheless, the agent's mission seems wholly dependent on the conditions as well as the request. In this work, the agent's focus is to eliminate the overall reward value accumulated throughout diverse learning occurrences. The reward therein proposed calculation has been split into two categories: reward for managing the parking service request there at the fog node as well as the reward for managing the parking service request at the centralized cloud infrastructure. Identical to the work [6,] the reward evaluation in this research involves assessing (a) communication delay as well as (b) computational delay. Even though there are adequate resources to implement & process the whole parking service request in a centralized cloud system, the propound infusion takes full advantage of the VANET cloud environment.Moreover, it raises the acceptance rate of the fog node environment without violating QoS or service deadlines. The centralized cloud gets employed only when computational resources are necessary.

Furthermore, the reward of a state may be computed by aggregating the reward values of all parking service requests. The agent must fulfill the minimum constraints while optimizing the clustered reward value:

$$D_i < D_{max}, \forall P_i \in U \tag{6}$$

This confirms that to avail of service from the nearby fog node, the user must be within the maximum communicationRange of thefog noderepresented asD_{max}.

4.7. Social Ski driven (SSD) routing algorithm

The social ski driver methodology was motivated by a plethora of metamorphic approaches employed in determining the optimum values for support vector machines [22], as well as delayed acceptance hill-climbing for characteristic selecting [3]. The purpose of SSD in this case is to search the space for repetitions or simulation cycles that are close to the best.As follows, the SSD calculates the best optimum solution: EMV position: The EMV's present location was determined as an objective function within this geographic region. Prior highest-ranking position acquired: The estimated fitness value for every EMV gets contrasted to prior finest data records & kept as the EMV's first location. Mean global best solution: Ultimately, the global best solution was gained by selecting the mean of every fitness value. Desired Users within the identical proximity will be given the very same distance-dependent preference.

The SSD [22] algorithm assists in gradually updating the position to obtain an optimal position. Furthermore, the locatiton of EMV gets upgraded via applying the velocity of the consumer V 0 at the 0th iteration, which is expressed as,

$$\begin{cases} qSin(\alpha 1)(P_i^0 - C_i^0) + \\ \mathrm{Sin}(\alpha 1)(G_i^0 - C_i^0);\ \mathrm{if}\ \alpha 2 \le 0.5 \\ qCos(\alpha 1)(P_i^0 - C_i^0) + \\ \mathrm{Cos}(\alpha 1)(G_i^0 - C_i^0);\ \mathrm{if}\ \alpha 2 > 0.5 \end{cases} \tag{7}$$

where q symbolizes the parameter balancing exploration as well as exploitation, α1, α2 symbolizes random numbers uniformly distributed in a range [0,1]. P 0 represents the best solution ofthe ith user at 0th iteration, C0represents the current solution of ith user and G0 denotes the global solution for the whole population of EMV. Battery state of charge for EMV typically calculated utilizing Columb's counting approach, which assesses the condition of the battery's charging & discharging current via incorporating the values across time would be represented by

$$B_{soc} = B_{soc}(t-1) + \frac{i(t)}{dt}\alpha t \tag{8}$$

Once the requested user gets the vacant parking slot using cluster dependent parking system the fitness function for the shortest route is evaluated using the Social Ski Driven routing algorithm based on the multi-objective parameters such as distance, traffic density, and battery status of the EMV. Because of the constrained driving distances of EMV, each meter of motion or idling of the automobile wastes power during traffic jams, therefore estimating the shortest path with the least amount of traffic levels is becoming a crucial issue. We created a fitness function that quantifies fitness values and therefore is given by

$$F = \frac{1}{2}\sum_{i=1}^{n} T_{den} + \frac{1}{2n} B_{soc}(t-1) + D_{sd} \tag{9}$$

Where BSoC is the current battery status of EMV, Tden is the traffic Density and Dsd indicates the shortest distance between the requested location of the Electric Motor Vehicle as well as the Parking Lot. The pseudo-code of Social ski driven routing algorithm is as follows:

Algorithm1: SocialSkiDriverroutingAlgorithm

Result: OptimalsolutionforShortestRouteinitialization;
Users'Current**Position**and**Velocity**;
whilethe *Stoppingcriterionis notmetforallEMV***do**GetthebatterystatusofEMVusingEq(8);**Estimate**thefitnessvalues;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EvaluatepastBestsolutionsandMeanGlobalsolutions ;
Computethe OptimalSolutionforthe shortestrouteusingEq(9);
AdjustVelocitiesofEMVtoreachthe destinationusing**optimalsolution**;
end

5. PERFORMANCEEVALUATION

The proposed Deep reinforcement learning-dependent Social Ski Driver (DRL-SSD) routing strategy for handling parking service requests to fog nodes with centralized cloud environments is being tested under diverse scenarios. This is in contrast to reinforcement learning coupled with SSD (RL-SSD), which employs a Q-learning strategy with restricted parameters.

The parking service requests are classified into two types:

(a) CPU demand (b) Memory demand. The memory demand for parking service requests ranges from 1GB to 2GB. On the other hand, the amount of CPU resource available in fog nodes and centralized cloud ranges between 16 to 32 and 64 to 128, respectively. The quantity of memory resources available in fog nodes and centralized clouds ranges between 30GB to 50GB and 50GB to 100GB, respectively.

5.1. Simulation Results

5.1.1. Diffusion of Parking requests in Fog nodes and Cloud

The proposed DRL-SSD routing strategy distributes parking service requests proficiently across neighboring fog nodes as well as the centralized cloud environment. Figure-5 and Figure-6 exhibit the comparison of the proposed DRL-SSD with theRL-SSD approach and represents the percentage of parking requests which are dispersed to nearby fog nodes as well as the cloud platform.

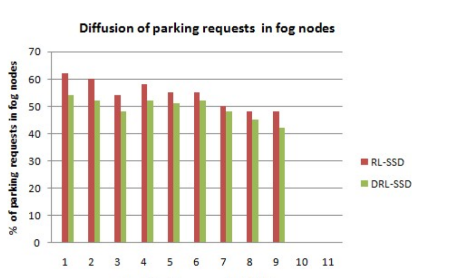

Figure 5: Diffusion of parking requests in fog nodes

Figure-5's X-axis depicts the count of Parking service requests, which ranges from 100 to 1000. The Y-axis, on the other hand, reflects the degree of dissemination of parking service requests within fog nodes. When contrasting the DRL-SSD strategy to the RL-SSD approach, it is revealed that a lower percentage of parking service requests have been sent to fog nodes. Once the count of requests reaches 900, the DRL-SSD technique diffuses roughly 42 percent of the parking service requests to fog nodes, contrasted to around 48 percent utilizing the RL-SSD strategy.

Alternatively, Figure-6 depicted that approximately 68% and 62% of parking service requests are diffuse using DRL-SSD and RL-SSD strategy, correspondingly when 100 Parking service requests showed up towards fog environment. Once the count of requests for parking service reaches 900, DRL- SSD strategy diffuses about 78% of the Parking service requests to the centralized cloud, whereas RL-SSD diffuses only approximately 70% to the cloud environment.

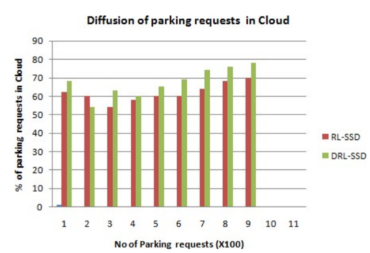

Figure 6: Diffusion of parking requests in the Cloud environment

It's critical to keep track of resource demand dispersion among fog nodes as well as centralized cloud systems while disseminating the parking service request. Figures 7 & 8 show the percentage distribution of CPU demand as well as memory resource demand, accordingly.

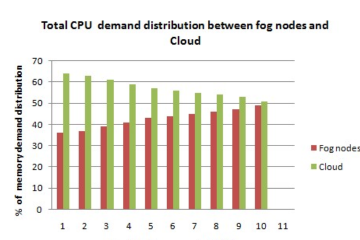

Figure 7: Total % of CPU demand distribution

Figure 7 shows the count of parking service requests as well as the percentage of resource requirement distribution across fog nodes & centralized cloud, correspondingly. The count of requests for parking services varies between 100 and 1000. It has been found that, while distributing resource demand, DRL-SSD prefers the cloud to accommodate the resource need of parking service requests. If there are 100 parking service requests, the cloud infrastructure fulfills roughly 64% of the CPU requirements, while local fog nodes gratify the remainder 36%. When the count of parking service demands reaches 1000, although, the distribution gap narrows. This might be because, as the count of parking requests grows, the centralized cloud assigns a higher portion of available CPU resources to parking service demands, as depicted in Figure-7.

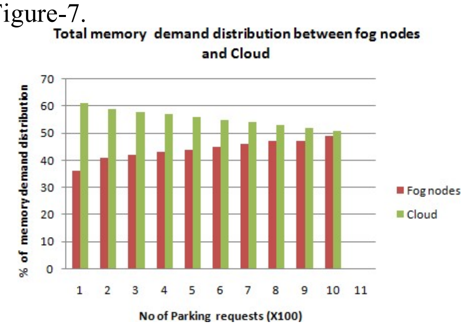

Figure 8: Total % of Memory demand distribution

Figure-8 illustrates an identical trend: when the count of parking service requests rises, the gap in the proportion of memory requirement distribution in fog nodes as well as centralized cloud shrinks. When there are 100 parking service requests, the centralized cloud fulfills roughly 64 percent of the memory need, while neighboring fog nodes satisfy the remaining 36 percent. When DRL-SSD receives 1000 parking service requests, it needs roughly 51% & 49% of the memory requirement, which would be met by the centralized cloud environment as well as fog node, correspondingly. Depending on the findings in Figures 6 & 7, it could be inferred that centralized cloud environments were given more precedence over fog nodes systems when it comes to meeting resource demands as well as managing parking service requests.

5.1.2. Success Rate (%) of Parking Allocatio

The proposed DRL-SSD has been further assessed by contrasting it to the RL-SSD utilizing success rates as a performance measure. Through investigating the state and action space, an agent might learn to proficiently disperse the parking service request a specific proportion of the time. In a cluster-dependent parking system, the success rate was measured against a parking service request from 300 EMV. Figure-9 depicts the count of parking requests allocated as well as the count of iterations. The count of iterations ranges from 1000 - to 10000 (along the X-axis). The success rate of parking allotment in percentage was expressed by the value on the Y-axis. Figure 5 illustrates that when the technique is run for 1000 iterations, the projected success rate for both the RL-SSD as well as DRL-SSD approaches is 33% & 31%, correspondingly. Once the count of repetitions reaches 2000, though, the situation becomes more complex. When the count of iterations gets raised to 10000, the rate of success of the proposed DRL-SSD strategy outperforms RL-SSD by 1%.

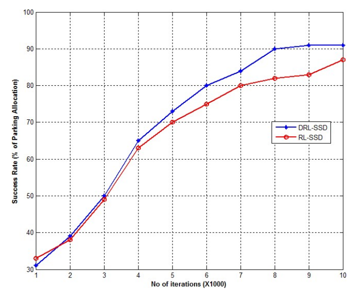

Figure 9: Success rate with 300 EMV

When the count of repetitions exceeds 9000, the success rate of the DRL-SSD strategy appears to be practically saturated. As a consequence, when the count of iterations is between 9000 and 10000, there is no substantial enhancement in the DRL-SSD success rate. Moreover,a significant increase has been observed with RL-SSD but still, it is less compared to DRL-SSD.

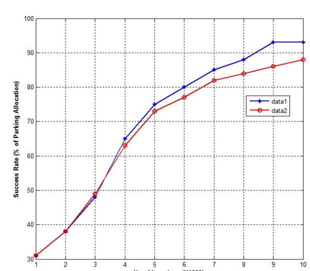

Figure 10: Success rate with 400 EMV

Increasing the number of vehicles to 400, as shown in Figure-10, Nevertheless, In the case of the DRL-SSD approach, the success rate of parking allocation gradually gets improvement after 3000 iterations against 400 parking requests, after 9000 iterations round it achieves successful parking allocation rate of 93% compared to 88% in RL-SSD approach.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Further Increasing the number of EMV to 500, as shown in Figure 11, However, In the case of the DRL-SSD approach, Initially

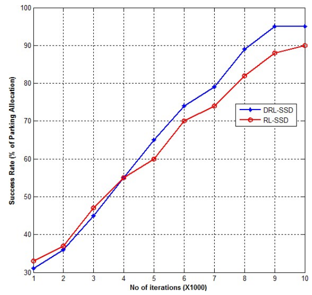

Figure 11: Success rate with 500 EMV

RL-SSD approach performs significantly well over the DRL-SSD approach, the success rate of parking allocation of DRL-SSD gradually gets improved as the number of iterations increases and it conquers the RL-SSD approach after 4000 iterations against 500 parking requests, after 10000 iteration rounds it achieves successful parking allocation rate of 95% compared to 90% in RL-SSD approach.

5.1.3. Performance Evaluation of SSD algorithm

Once the Allocation of the parking service request is done With the help of the DRL method by diffusing requests in nearby fog nodes and cloud environment after obtaining the available vacant parking slot using a cluster-based parking system. An efficient routing algorithm is required to route to the parking destination, hence we have proposed a social ski driven routing algorithm compared with a population dependent stochastic optimization approach termed Particle Swarm Optimization (PSO) for evaluating the performance efficiency, DRL-PSO is a kind of searching technique based on swarm most widely used for routing in particular dimensional search. Hence PSO is considered with DRL here for performance analysis of Traffic density, Delay, and Fitness valuebased on multi-objective parameters with the DRL-SSD algorithm in this research work.

The fitness values were computed utilizing Eq:(9), taking into account the multi-objective attributes of battery capacity, traffic density, as well as distance. The fitness values assessed by DRL-PSO and proposed DRL-SSD were 18.235 as well as 16.453, correspondingly. By raising the count of simulation cycles to 100, the fitness values are even further enhanced. The DRL-PSO, as well as proposed DRL-SSD optimal fitness values, seem to be 16.23 & 15.88, correspondingly. As a consequence, the DRL- PSO optimization tactic outperforms the assessed fitness values for diverse simulation cycles concerning traffic density, battery power status, as well as latency. Figure-13 depicts the assessment of methodologies with traffic density.

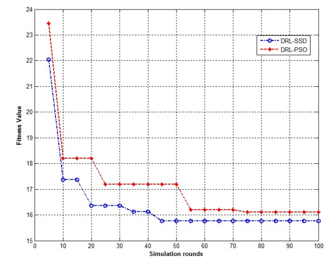

Figure 12: Fitness values after multiple iterations

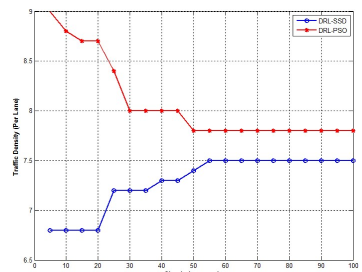

Figure 13: Analysis of Traffic Density

When simulation rounds were expanded to 100 to get the optimal traffic density per lane determined by DRL-PSO is 7.8 while the proposed DRL-SSD is 7.5, indicating that the proposed DRL-SSD methodology picked the route with the least amount of traffic per lane.

Figure 14 illustrates that for the first 20 simulation cycles, the latency in calculating the optimum paths by DRL-PSO as well as proposed DRL-SSD is 8.455 minutes & 8.55 minutes, respectively. As the simulation rounds boost to 100, the latency diminishes, and also the latency estimated by DRL-PSO as well as proposed DRL-SSD seems to be 7.65 minutes and 7.55 minutes, respectively. As an outcome, the proposed methodology demonstrated enhanced effectiveness with little delay by optimizing the efficiency of EMV battery to reach vacant Parking slots to avail Parking with a charging facility at theearliest.

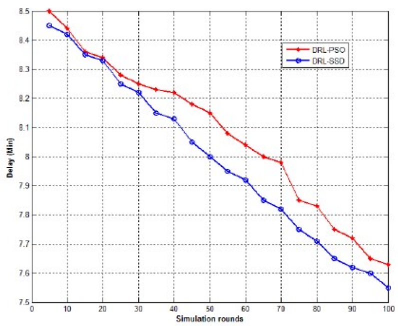

Figure 14: Analysis of Delay (per Lane)

6. CONCLUSION

In this paper, a novel DRL-SSD-dependent dynamic handling of parking service requests using cluster dependent parking system by diffusing in fog nodes as well as cloud computing environments has been adopted. Both nearby fog nodes, as well as the vehicular cloud server, execute every parking service request. This would be accomplished by distributing the count of incoming parking service requests to fog nodes & vehicular cloud environments. The DRL approach is being used, in which the agent has the power to decide what percentage of parking service requests should be offloaded to the vehicular cloud. The learning agent's foremost target is to diffuse a large count of parking requests towards the fog nodes while preserving the preferred service quality and lessening delay. The Keras framework is being utilized to implement and model the proposed DRL-SSD. The performance of DRL-SSD was compared with RL-SSD and computed in terms of its success rate of parking allocation using cluster-based parking system, percentage of parking service requests deployed in centralized vehicular cloud and fog nodes and the resource (CPU and Memory) demand dispersed between fog and cloud server, further once the vacant slot is booked the DRL-SSD based routing algorithm finds the shortest route and assists the user through google map, the performance evaluation of DRL- SSD is compared with DRL-PSO, it is observed that the devised fitness function evaluates considering multi-objective parameters such as traffic density, status of EMV battery power and distance parameter based on which a routing decision are made, Hence by using proposed Deep learning-based Social Ski driven (DRL-SSD) routing protocol and cluster-based parking system achieves allocation of parking slots with high probability of success rate of 95% with minimal traffic density7.5 per lane, minimal delay of 7.55min and minimal fitness value of 15.88. In the future, when enhancing the existing system, bandwidth demand will be regarded as another crucial factor.

REFERENCES

[1] S. C. Hiremath, and J. D. Mallapur, “Optimization of Parking Dynamics in Smart City using Cloud Networks”, *In 2019 IEEE Bombay Section Signature Conference (IBSSC),* 2019, pp. 1-6

[2] C. K. Dehury, and S. N. Srirama, “An efficient service dispersal mechanism for fog and cloud computing using deep reinforcement learning”, *In 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020, pp. 589-598.

[3] C. K. Dehury and S. N. Srirama, “Personalized Service Delivery using Reinforcement Learning in Fog and Cloud Environment”, *In 21st International Conference on Information Integration and Web-based Applications Services (iiWAS2019)*, 2019, pp. 524–531.

[4] Y. He, N. Zhao, and H. Yin, “Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach”, *IEEE Transactions on Vehicular Technology*, Vol. 67, No. 1, 2008, pp. 44–55.

[5] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. S. Oh, “Semi supervised deep reinforcement learning in support of iot and smart city services”, *IEEE Internet of Things Journal,* Vol. 5, No. 2, 2017, pp. 624–635.

[6] J. Zhu, F. Lauri, A. Koukam, and V. Hilaire, “Managing ventilation systems for improving user comfort in smart buildings using rein-forcement learning agents”, *In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2014),* 2014, pp. 149–155.

[7] Z. Jiang, D. Xu, and J. Liang, “A deep reinforcement learning framework for the financial portfolio management problem”, *arXiv preprint arXiv:1706.10059,* [Online]. Available: https://arxiv.org/pdf/1706.10059.pdf, (2017)

[8] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforce- ment learning framework for autonomous driving”, *Electronic Imaging*, Vol. 2017, No. 19, 2017, pp. 70–76.

[9] X. Liang, T. Wang, L. Yang, and E. Xing, Cirl, “Controllable imitative reinforcement learning

for vision-based self-driving", *In European Conference on Computer Vision (ECCV)*, 2018.

[10] B. Chatterjee, T. Bhattacharyya, K. K. Ghosh, P. K. Singh, Z. W. Geem and R. Sarkar, "Late Acceptance Hill Climbing Based Social Ski Driver Algorithm for Feature Selection", *IEEE Access*, Vol. 8, 2020, pp. 75393- 75408, doi: 10.1109/ACCESS.2020.2988157.

[11] D. Niyato and A. Asheralieva, "Hierarchical Game-Theoretic and Rein- forcement Learning Framework for Computational Offloading in UAV Enabled Mobile Edge Computing Networks with Multiple Service Providers", *IEEE Internet of Things Journal*, 2019, pp. 1-1.

[12] Y. He, N. Zhao, and H. Yin, "Integrated Networking, Caching, and Computing for Connected Vehicles: A Deep Reinforcement Learning Approach", *IEEE Transactions on Vehicular Technology*, Vol. 67, No. 1, 2018, pp. 44–55.

[13] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Reinforcement learning for resource provisioning in the vehicular cloud", *IEEE Wireless Communications*, Vol. 23, No. 4, 2016, pp. 128–135.

[14] M. Karthi, and P. Harris, "Smart parking with reservation in cloud based environment", *In 2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM),* 2016, pp. 164-167.

[15] R. Aswini, and T. Archana, "Automatic Car Parking System Using Raspberry-Pi with Cloud Storage Environment", *In 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019, pp. 1-5.

[16] S. Valipour, M. Siam, E. Stroulia, and M. Jagersand, "Parking-stall vacancy indicator system, based on deep convolutional neural networks", *In 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 2016, pp. 655-660.

[17] G. Krasner, and E. Katz, "Automatic parking identification and vehicle guidance with road awareness", *In 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE),*2016, pp. 1-5.

[18] R. Gupta, S. Pradhan, A. Haridas, and D. C. Karia, "Cloud based smart parking system", *In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT),* 2018, pp. 341-345.

[19] J. Joshi, M. Harsha, U. Kurulkar, A. Reddy, D. Akhil, B.N. Reddy and V. Sahithi, "Intelligent parking allotment system based on traffic prediction", *In 2016 International Conference on Computer and Communication Engineering (ICCCE)*, 2016, pp. 321-326.

[20] A. Zajam, and S. Dholay, "Detecting efficient parking space using smart parking", *In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2018, pp. 1-7.

[21] D. Wang, D. Tan and L. Liu, "Particle swarm optimization algorithm: an overview", *Soft Computing*, Vol. 22, No. 2, 2018, pp. 387-408.

[22] A. Tharwat, T. and Gabel, "Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm", *Neural Computing and Applications*, Vol. 32, No. 11, 2020, pp. 6925-6938.

[23] T. N. Pham, M. F. Tsai, D. B. Nguyen, C. R. Dow, and D. J. Deng, "A cloud-based smart-parking system based on Internet-of-Things technologies", *IEEE access,* Vol. 3, 2015, pp. 1581-1591.

[24] P. B. Bautista, L. L. Cárdenas, L. U. Aguiar, and M. A. Igartua, "A traffic-aware electric vehicle charging management system for smart cities", *Vehicular Communications*, Vol. 20, 2019, pp. 100188.

[25] B. N. Rao, P. Krishna and V. Yarlagadda, "Mitigation of photovoltaic solar system harmonics using shunt active power filter", *In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC),* 2020, pp. 566-571.

[26] S. Zhang, Y. Gajpal, S. S. Appadoo and M. M. S. Abdulkader, "Electric vehicle routing problem with recharging stations for minimizing energy consumption", *International Journal of Production Economics*, Vol. 203, 2018, pp. 404-413.

[27] Keras documentation, https://keras.io/, 2019, [Online; accessed 21- November-2019].

[28] G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, A. Lopez Garcia, I. Heredia and L. Hluchý, "Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey", *Artificial Intelligence Review,* Vol. 52, No. 1, 2019, pp. 77-124.