Smart Parking Model based on Internet of Things (IoT) and TensorFlow

by

Md Omar Hasan
16301178

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2020

Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

Md Omar Hasan

Md Omar Hasan
16301178

Approval

The thesis titled “Smart Parking Model based on Internet of Things(IoT) and TensorFlow” submitted by

1. Md Omar Hasan (16301178)

Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 09, 2020.

Examining Committee:

Supervisor:
(Member)

Md. Motaharul Islam

Dr. Md. Motaharul Islam, PhD
Professor
Department of Computer Science and Engineering
United International University (UIU)
Former Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

Mahbubul Alam Majumdar, PhD
Professor and Chairperson(CSE), Interim Dean, School of Sciences
Department of Computer Science and Engineering
Brac University

Ethics Statement

I, hereby declare that this thesis is based on the findings of my research. All other materials used have been properly noted in the document. This work, neither in whole nor in part, has never been submitted by anyone to any other university or institution for the award of any degree.

Abstract

Our paper proposes a smart parking model to reduce a wastage of time by allocating the free parking spaces, ensures security and proposed a recommender system. We have targeted main three segments of our smart parking model. First, we have proposed a cloud based architecture. Through the architecture we shows how to allocate spaces of parking area among the users. We have used Google Cloud IoT core service to generate our cloud architecture. Secondly, we introduced the security portion of a parking system. License plate is an important aspect for those parking places where people reserve parking places for their comfortable journey. We used Deep Learning (DL) based architecture to detect and recognize the license plates of users. TensorFlow framework is used to detect and recognize the license plates for user authentication. Finally, we focused on building a recommender system to recommend a parking place where multiple parking areas are exist in one areas. The Reinforcement Learning (RL) algorithm is used to show how a recommender system can work. We have got 98% accuracy in terms of detecting and recognizing the license plates. Also, our recommender system performs 50% better than the existing algorithms. Overall, our proposed model is able to reduce the wastage of time by 50% which is a very good accuracy and ideal for practical deployment in our real life.

Keywords: Smart Parking, Recommender System, Deep Learning, Deep Reinforcement Learning, Reinforcement Learning, Cloud Computing, TensorFlow, Intelligent, IoT.

Dedication

I would like to dedicate this research work to my parents. Without their utmost support, I could not have achieved or came so far. Also, I want to dedicate my work to Dr. Motaharul Islam Sir who constantly pushed me all the way and Dr. Golam Rabiul Alam Sir who helped me a lot to generate ideas in my paper.

Acknowledgement

To begin with, all praise to the Great Allah for whom my thesis have been completed without any major interference.
Further, I would like thank our honorable supervisor Dr. Md. Motaharul Islam sir for his support, feedback and guidence. He encouraged me to conduct the research, give guidance to me and always were present to offer any help I could ask for.
Moreover, I also thank our respected BRAC University for providing us with the necessary facilities and resources during our research period.
Finally, special thanks to my parents; without their constant support and encouragement it might not have been possible. With their kind support and prayer I am now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$A2C$ Advantage Actor Critic

$A3C$ Asynchronous Advantage Actor Critic

AC Actor Critic

API Application Programming Interface

CRF conditional random field

DL deep learning

DNN deep neural network

DRL deep Reinforcement learning

FC fully connected

FCN fully convolutional network

FN false negative

FOV field-of-view

FP false positive

IoT Internet of Things

ML machine learning

NN neural network

RL Reinforcement Learning

Chapter 1

Introduction

In Bangladesh, number of car ownership is increasing day by day. Due to the narrow road structure, bad maintenance of traffic system, people do not have information and are not aware of finding a parking areas. In fact, the number of parking areas are not increased seeing the increment of car ownership. We tried to focus on these issues to be our first motivation of building our novel architecture of smart parking system. We believe our architecture can resolve these kinds of issues in future. Parking in unauthorized places creates a chaos and traffic congestion. This creates an irrelevant frustration among drivers or car owners how and where they park their car. Moreover, we have seen a wastage of time due to searching and navigating to the parking areas. With the blessing of IoT, Edge, Cloud, IoT, and ML algorithms can bring a good vibe of solution among these issues.

IoT is a concept where sensors, cloud, etc. everything is connected altogether. In parking areas sensors are used in mostly collecting the data [3, 4, 5, 6, 7]. Ultrasonic sensors are very powerful and popular to detect a place whether it is empty or not. A micro controller is a small chip device that can be used to connect with these sensors to collect data. Arduino or Raspberry Pi is popular known as a micro controller device. These are very easy to manage and use.

Cloud is the most reliable way to maintain and manage data. In fact, cloud gives us the security what we need for our data. In terms of Parking system maintaining data in cloud is important. Cloud provides different kinds of services where we can easily handle our data in day to day life. Through cloud services it is very easy to integrate the cloud with the sensor data as cloud itself is well manageable in terms of keeping those data [10, 11]. Also, cloud provides different Application Programming Interface (API) and for developers it is easy to integrate the cloud with smartphones. Therefore, through smartphone users can check parking status. Well there are different protocols used to transfer data. The use of CoAP and MQTT transfer data among cloud and devices. In paper [15] authors suggest to use CoAP protocol as low cost high performance to transfer data.

DL is the area of Artificial Intelligence. It changes the paradigm of researches to think in a differnt way. By using DL we can train our machine to develop the analyticial capabilities. In Image Processing, Computer Vision, Image analysis, Recognition of faces, etc. DL is very popular and successful term to apply. Different cloud services providers develop and provide DL related interface for developers to develop and make solutions easy and effective.

1.1 Recommender Systems

Popularity of various recommender systems is increasing day by day. People will take help from this recommender systems to take decision in their day to day life. Most importantly, it saves a lot of time and make our daily life easier. A recommender system can be designed in various ways. Content based recommender systems requires information about different types of items. For example, content based recommender system such as a movie genre, actor, director, year, etc. These extracted data is understood by using Natural Language Processing (NLP).

1.1.1 Techniques to build a recommender system

Collaborative Filtering (CF) is a popular method for developing a recommender systems[1,2,3]. CF takes previous historical data, because one perception that the CF adopts that user who like some items previous and can like the items now. Explicit and Implicit are the two types of approaches of the CF recommender systems. Explicit rating gives access to users to rate to any particular items. It is a direct feedback from the users. Serendipity is another advantage of CF. CF models helps users to discover new interests where the user may not interest but the model based on similar user preference suggest the item to users.

The term called data sparsity is considered as a positive element to improve the CF based models. The main reason is that if we increase number of data the model will improve gradually. But the questionable relationship exists between the number of data and user items. The authors in [4] briefly explained the disadvantages of using data sparsity in CF based models. another disadvantage of CF is that it only considers users or items only. It often makes it difficult to detect important patterns and recommendation quality drops because of it.

1.1.2 Reinforcement Learning

The popular and ongoing research field of Artificial Intelligence is RL. The major components are used in RL is called agent, policy, state, action and reward. The agent takes action in an environment e.g. atari game. The state changes after each actions of agent. To maximize the reward is the main goal of RL. For any wrong action, the agent will be penalized. For stochastic type environment, where the environment is unknown, the agent will take random actions to reach its goal state. Maximization of reward will provoke agent to take positive action to reach the goal.

1.1.3 RL in recommender systems

RL proves a great advantage where uncertain situations exist a lot. In a parking system, some of the uncertain terms are time, distance, security, etc. If we want to apply CF models here definitely the model will not perform good because the uncertain cases can not follow previous data. For example, for a car the calculated time between the source and destination can be changed at any time because of some uncertain cases such as car can stop suddenly, traffic congestion, etc. So, here if we apply RL approaches we can handle this uncertain case. To merge the concept of Deep learning can be an efficient approach to outperform our model better.

1.1.4 DL with RL combination

DL is so vast that it is now integrated with RL for computers to take decisions in more precisely. We called the term DRL. As we know that RL has many flaws to take decisions but due to DL computer or our machine try to analyse the activity or action take by it and try to make proper decision. The vision of Artifical Intelligence (AI) is now gradually developing day by day.

1.1.5 Architecture Selection

To select our architecture we need to identify our problems and approach for that problems. For RL based problems the architecture can be in different type. Our proposed architecture will be combined with neural network architecture.

1.1.6 Our Contributions

We did lot of contributions here:

- To allocate the parking spaces cloud based architecture is proposed
- A UI of Android application is proposed
- We implemented CNN and we got accuracy of 98%
- implement Q-learning algorithm to recommend parking places to users
- A navigation bot is proposed for future works to help users navigating to the parking place.
- evaluate Q-learning algorithm which outperform better than the other systems

We have structured our paper in different sections. We have reviewed different existing models of recommender systems in section 2. We try to focus on both pros and cons of each existing works and also discuss how recommender system is used in different domains. Our approach to collect and preprocess the data is discussed in section 3. Methodology is defined in section 4 where described our proposed architecture. Section 5 is about Algorithmic Analysis of our used algorithm. We evaluated our model in section 6. Conclusion and Future idea will be described in section 7.

1.2 Preliminary Idea

1.2.1 Elements of Reinforcement Learning

The Environment is the full area where the agent will perform to maximize rewards. There are 4 main elements of an environment of RL system and they are reward, value function, agent, policy, and a model.
The agent's way of learning is defined by the policy. The policy is the core of RL agent which determine the behavior of the agent. The policy may be a simple function or a lookup table where in some cases such as search process, it takes extensive computation. Policies may be stochastic or random possibilities for each action.

Reward signal refers the goal of RL problem. After every step, the agent receives a score which is called reward. The agent's main target is to maximize the reward to reach the goal. Reward signals define both good and bad. If the agent choose a wrong step in certain episode, it will receive negative reward. Reward signal can be stochastic in terms of state and action chosen by the agent.
To specify the good for the long run value function specifies this thing. It represents the total reward where the agent can achieve in future run. In a sense, value determines the long-term desirability of the states. The relation between rewards and values is quite similar. Without rewards there can not be any values also without values we can not determine the performance of choosing the action in a long term. The evironment actions provides rewards and the values is determined by the episodic tasks.
The final element of the RL is the model. Model is something to show the behaviour of the environment. We used models for experiencing the future situations of environment. Model-based methods are used where we generally made a model of the environment and agent behaviour will be based on our model. Model-free represents that the agent itself does the trial and error task to learn by itself and it does not depend on the model.

1.3 Research Methodology

This thesis aims to focus on different flaws on smart parking systems. We have deeply studied on existing previous works and tried to find the flaws of every system. Our primary focus was to improve different segments of this parking system. So, first we focused on how we can allocate the parking places among users. We used Google Cloud IoT Core and some related services to propose our architecture. After then we tried to improve the security portion of the license plate. We have made our own data set in terms of recognizing the plates. We also used MNIST dataset for english license plate recognition. Lastly our attempt was to make a recommender system to help commuters to find a nearby parking area.

1.4 Scope and Limitation

Due to shortage or time and some funding limitations we can not complete our full prototype for showcasing. We were almost completed phase but again the pandemic hits badly and our workflow becomes slow again. But we were optimistic to our goals and finally we can say we are almost at our goal place. There are also different advance tools that we need to use but due to some shortage of time we could not do that.

Chapter 2

Literature Review and Related Work

The need of recommender system is increased in different Internet of Things (IoT) architectures. In [1] authors described a smart parking system where they proposed a navigation based recommendation based on Deep Reinforcement Learning (DRL). Their primary motivation was to introduce a parking system where the time will saved in an efficient manner. Inspired by their previous work, authors in [2] introduced a recommender system using Cloud Computing and Q-Learning.
In [3] authors proposed a server-based recommendation system. The recommendation works based on the user distance from the destination spot. An efficient use of K-medoid clustering method and Conditional Random Fields is proposed by the authors. The clustering technique is a good approach to apply here.
A parking algorithm is proposed to recommend a parking area[4]. The authors used old page rank algorithm based on public information. The parking places are sorted and recommend the places for parking.
By using the minimum short distance for recommendation the authors in [2] uses the minimum short distance algorithms. Their proposed system includes booking, navigation and paying bills online. But some uncertain cases the proposed method may not be working properly.
The authors used K-medoid clustering and Conditional Random Fields to detect user parking [3]. Their approach is basically a server based system for parking recommendation. Their approach is unique but there is some lacking in clustering method.
In [4] Internet of Things (IoT) based contextualization techniques is used. It is used to reduce the complexity of data processing. The authors collected data from the sensors and the data will be contextualized. The server based system is beneficial for parking recommender system.
In [5] authors focued on mobile communication e.g. GPS, NFC, etc. to propose a parking recommender system. They proposed different types of periodical recommendations based on different scenarios. The periodical recommendation is effective to reduce the parking conflict.
Authors used GPS trajectories to find the parking places where the maximum number of passengers wait for the taxis. Basically they specified the fixed places and proposed a recommender system so that the taxi drivers picked up the passengers in a very less time to maximize the profit. The system is very effective to reduce the cost of waiting time for those taxi drivers and also the ride sharing services can easily adopt this strategy to increase the benefits of customer service.
In modern cities people use reservation techniques for parking their cars. Authors in

[7] use this strategy to propose a recommender system. They proposed a selection method to identify the optimal parking lot. The goal of their proposed architecture was to reduce the waiting time, cost and improve facilities.
Based on the exploiting information of user interaction collaborative filtering is used to recommend users.
Authors in [10] used wireless sensors to get the data of free parking spaces and using the Light-Emitting Diode (LED) will be helpful to find the right slot in the indoor parking area. Well using too many LEDs is expensive and in big parking areas is maybe sometimes trouble to find a place.
In [12] authors used ZigBee based wireless communication system and used GSM for the notifications of parking spaces. It seems a smart solution but there is a question arise on how fast the GSM process is. Moreover, they used IR sensors to find a parking places.
Based on a webpage showing authors in [13] illustrated how they distribute the parking areas. To check the parking status a notification messege will be sent to smartphones. Their approach is costly because they proposed two types of sensors in indoor and outdoor parking areas.
In [13] creators build up a site page to show the vacant spaces of the stopping region in a constant checking framework. A notice through message will spring up to shows that the stopping territory is full. They have utilized two sorts of sensors for outside and indoor stopping zone. Nonetheless, taking information of the parking spots is a decent methodology than other framework yet to refresh or oversee information to a page is an excruciating position and utilizing two kinds of sensors will be expensive.
A proposed model [14] utilize some IR sensors and an inserted regulator for ongoing information assortment where they build up a data set worker to store and update information of parking spots. They utilize a Direct Current (DC) engine at the passage door to permit access for the Identity Document (ID) holders. utilizing DC engine to permit the entrance is a valuable methodology for the security purposes however the model won't perform well when there are an excessive number of vehicles simultaneously.
In [15] the creator proposed an E-Parking framework where the creator utilizes Parking Meter (PM) with ultrasonic sensors for finding the free parking spot. A LED framework is utilized to show the stopping territory is saved or not. This framework is useful for sharing parking spots yet it won't work great to distribute the parking spots.

Chapter 3

Data

Data set Collection for License Plates

We need to identify the license plates. That is why we need huge dataset to identify the digits properly. the below figures show that we have used MNIST dataset the huge dataset for digits and characters.

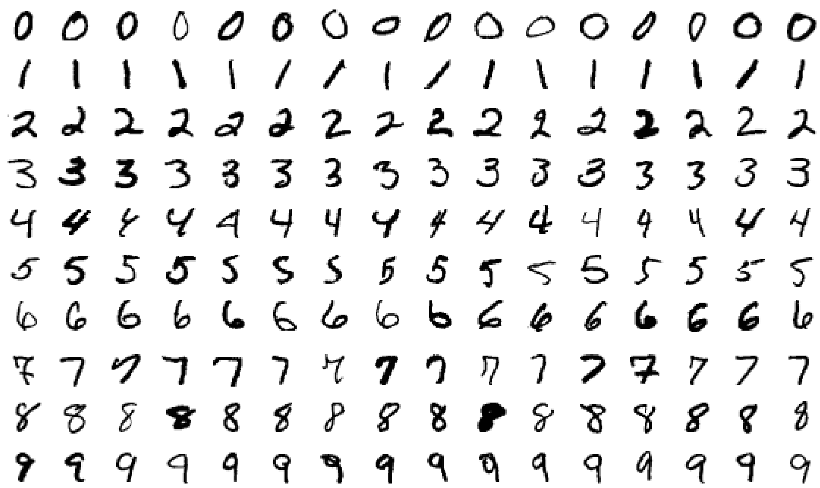

Figure 3.1: MNIST digit dataset

Bangla dataset for Bangla license plate

We train our network in Bangla dataset to detect bangla license plate which was a big contribution for recognizing bangla characters. We used Kaggle dataset for Bangla characters and captured real life data for this project. We have applied Data Augmentation to increase the size of the data.
Information expansion in information investigation are procedures used to expand the measure of information by including marginally adjusted duplicates of previously existing information or recently made engineered information from existing information. It lessens overfitting when preparing a machine learning.[1] It is clesely identified with oversampling in information investigation.
The below figures show the CSV and sample dataset used for bangla license plate detection.

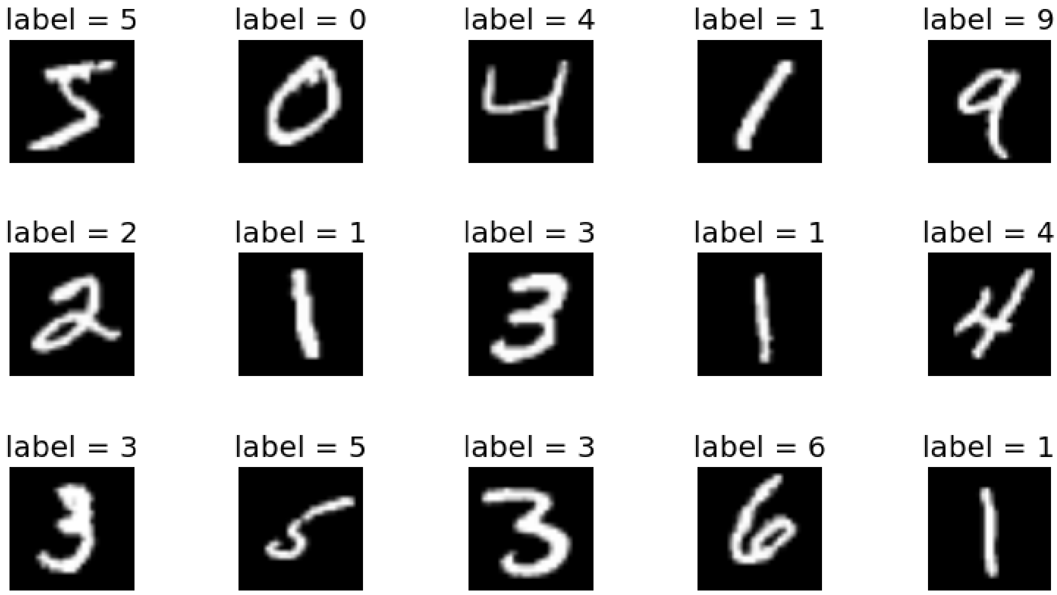

Figure 3.2: MNIST dataset for numbers

অক্ষর	অক্ষর	অক্ষর	অক্ষর	অক্ষর	অক্ষর	অক্ষর	অক্ষর
ব	ল	আ					
ক	থ	আ					
ই	ল	ই	শ				
ব	আ	ল	ই	ক	আ		
ম	ন	এ					
স	আ	প					
চ	ই	ঠ	ই				
হ	আ	ল	ক	আ			
প	আ	খ	আ	ল	ই		
ক	আ	র	ব	আ	র	ই	
ল	ই	খ	ত	এ			
ল	এ	খ	আ	প	ড়	আ	
ল	ন	ড	ন				
স	র	ক	আ	র			
হ	আ	ল	উ	ম			
ম	এ	ঘ	ন	আ	দ		
ব	আ	র	ই	ধ	আ	র	আ
ক	ল	এ	জ				
ব	ই	স	ত	আ	র		
প	র	থ	ম	এ			
স	ও	ন	আ	ল	ই		

Figure 3.3: Our own dataset for character recognition

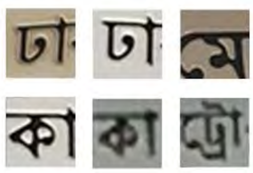

Figure 3.4: bangla dataset

Chapter 4

Algorithmic Analysis

The Q-learning agent

Q-learning agent will be designed in a MDP framework. States and Actions of agent is defined by S and A. The reward function is presented as:

$$R_c a(s,s') = E(r_{t+1}|s_t = s, s_{t+1} = s', ca_t = ca) \tag{4.1}$$

Transition Probability function as:

$$P_c a(s,s') = P_r(s_{t+1} = s'|s_t = s, ca_t = ca) \tag{4.2}$$

The Q function for both states and actions $Q(s,a)$ are:

$$Q(s_t, ca_t) = Q(s_t, ca_t) + \alpha * (r + \gamma * Q(s_{t+1}, ca_t) - Q(s_t, ca_t)) \tag{4.3}$$

Distance, time, cost and security are the main four parameters to train our agent. Parameters P defined as, $(P_i, ...P_n)$, we can define our following states of the agent:

$$S_{rcp} = (r_i c_i, p_i, ..., r_n c_n, p_n) \tag{4.4}$$

For goal oriented or ε-greedy perception, each positive and negative actions, agent will receive rewards. Positive sign of each numeric reward is used to define the goal oriented positive steps. Moreover, negative rewards for each steps, are defined as negative rewards. We assign positive 10 for every goal oriented positive actions and negative 10 for the other part. For each state s_t and action a_t, we can define our algorithm by the following ways. Alpha α and Gamma γ values for tweak the learning of agent. The above algorithm is online value function learning where we show how our agent will change states s_t by taking actions a_t. The agent will get both positive or negative rewards based on the actions taken.

Algorithm 1 Online based function learning

Initialization: value function table
REPEAT
$s_t = current\_state()$;
$a_t = action(s_t)$;
RECONFIGURATION(a_t);
Agent Observe_reward();
$s_{t+1} = current\_state()$;
$a_{t+1} = get\_action(s_{t+1})$;
$Q(s_t, a_t) = Q(s_t, a_t) + \alpha$ *(r + γ *$Q(s_{t+1}, a_t)$ - $Q(s_t, a_t)$);
$s_t = s_{t+1}$, $a_t = a_{t+1}$;
UNTIL convergance done

Algorithm 2 Recommeder Performance Controller

for Every time step **do**
  Calculate target seeking command $\mathbf{x}_{tsCmd}$
  for All map measurements from $\mathbf{x}_{Map}$ **do**
    Denormalize measurement
    Add margin of safety
    Calculate altitude difference $\Delta h_{ObsSafe_j}$ to aircraft
    if $\Delta h_{ObsSafe_j} > 0$ **then**
      Add measurement to set of critical measurements $\mathcal{M}_{crit}$
    end if
  end for
  for All measurements in $\mathcal{M}_{crit}$ **do**
    Calculate local obstacle avoidance vector
  end for
  Sum over all local avoidance vectors
  Transform to global coordinate frame to receive $\mathbf{x}_{oaCmd}$
  Calculate obstacle avoidance weight w_{oa} based on critical zone weight
  Calculate target seeking weight w_{ts} as $1 - w_{oa}$
  Calculate command vector $\mathbf{x}_{HSaCmd} = w_{oa}\mathbf{x}_{oaCmd} + w_{ts}\mathbf{x}_{tsCmd}$
end for

Chapter 5

Evaluation & Result Analysis

we passed on the structure in a business constructing free space and directed a customer consider suffering a month. The sending crossed two stories in a grounds constructing, and incorporated a varying arrangement of spaces including desk areas, work environments, gathering rooms, and open spaces, for example, lobbies. A couple of sensors are set up inside the spaces to prompt constant data through Edge Computing.

5.0.1 Simulation of data processing of Cloud

DynamoDB shows the parking status of the following table.

Table 5.1: Data Table

ID of Sensor	*SensorID*	*MsgID*	*Sts*	*TimeOfSts*	*Loc.*	*Vehicle*	*ParkingSpot*	*Other*
MKU12B1L1EP1S	S500	M1200	Emp	2/2/20 25.11.2020	UT66	SUVI	EST	Out
MKU12B1L1EP1S	S500	M1200	Emp	2/2/20 25.11.2020	UT66	SUVI	EST	On
MKU12B1L1S	S600	M1400	Emp	2/2/20 2.20.2020	GT56	–	WST	Out

Ultrasonic sensors gather info and sending to the cloud we get all information that we need. Edge server send the status of parking areas. Greengrass, AWS Kinesis and Data Analytics for preparation. There are different table entries in the following table. The name of entries are: ID of Sensor, SensorID, MsgID, Sts, TimeOfSts, Loc., VehicleType, ParkingSpotNo., OtherInfo. etc.
For example, the first column is the name of Sensor. The other columns are the name of territory name, status of the sensor, proper time, type of the vehicle, etc. the first entry is about Uttara area sector 12 where first column East side building number 1. The info table sends message of the sensor that's it.

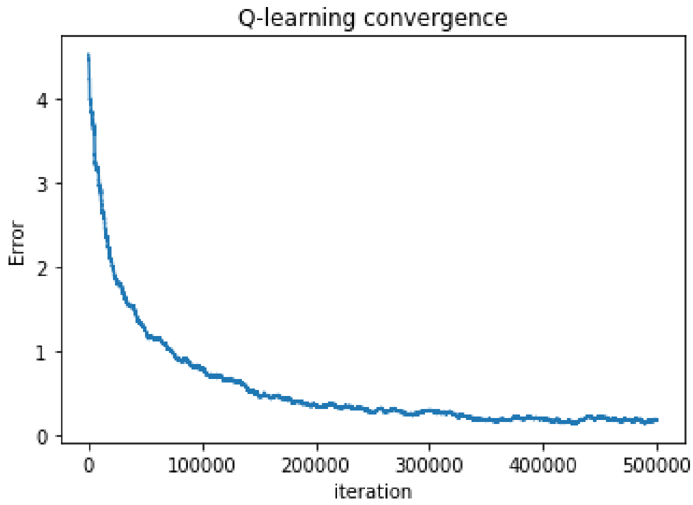

Figure 5.1: Convergence between error vs iteration of Q-learning with ε-greedy approach.

5.0.2 Q-learning Convergence with greedy approach

The Q-Learning calculation permits us to gauge the ideal Q work utilizing just directions from the MDP acquired by following some investigation strategy. Q-learning with epsilon-insatiable investigation does the accompanying update at time t:

5.0.3 epsilon-greedy policy vs soft max policy

The epsilon-avaricious strategy is an uncommon instance of epsilon arrangements. It picks the best activity with a likelihood and an arbitrary activity with likelihood. There are two issues with epsilon-avaricious. To start with, when it picks the irregular activities, it picks them consistently, including the activities we know are terrible. This restricts the presentation in preparing and this is terrible for creation conditions. In this way, when the organization is utilized to assess execution or control a framework, epsilon ought to be set to zero. On the other side, setting epsilon to zero makes per second issue: we are currently just abusing our insight. We quit investigating. On the off chance that the elements of the framework changes somewhat, our calculation can't adjust.

An answer for this issue is to choose irregular activities with probabilities relative to their present qualities. This is the thing that softmax strategies do.

The underneath diagram shows the presentation timestrap during roundabout errands to give disadvantage yields. The yellow line characterizes the superior and the red imprint is the benchmark or moderate execution than the current models, for example, bunching, bi-grouping and so forth The realtime estimations
are appeared in blue, and the reproduced estimations are appeared
in orange.

5.0.4 Agent Performance

An answer for this issue is to choose irregular activities with probabilities relative to their present qualities. This is the thing that softmax strategies do.

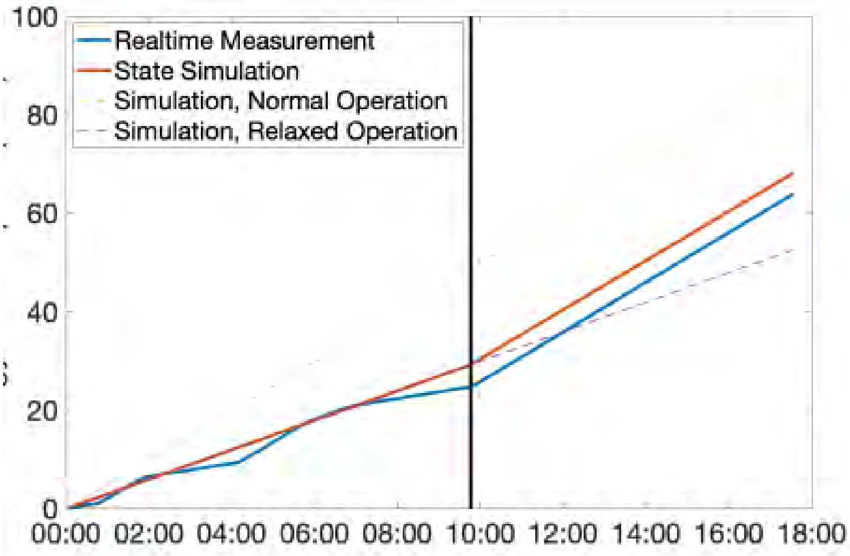

Figure 5.2: performance after episodic tasks of the agent bot during the traversal from one node to different connected nodes. Comparison of episodic tasks data with simulated normal operation, simulated relaxed operation, and simulation state.

The underneath diagram shows the presentation timestrap during roundabout errands to give disadvantage yields. The yellow line characterizes the superior and the red imprint is the benchmark or moderate execution than the current models, for example, bunching, bi-grouping and so forth The realtime estimations
are appeared in blue, and the reproduced estimations are appeared
in orange.

5.0.5 User case study performance

The performance of Q-learning agent is notable in terms of taking decisions. Well, sometimes it is time consuming as the graph appears that it takes some more time in terms of decision but still it is efficient in taking decisions. The parameters is learned very well by our agent that is why it performs good.

Table 5.2: Comparison Table

Criterias	***Agent***	***CF***
Areas(Parking)	20	20
Requests per min	10	10
GPS	Yes	No
Free Spaces	20	20
Approach to Solution	both	exploit only

The performance of Q-learning agent is notable in terms of taking decisions. Well, sometimes it is time consuming as the graph appears that it takes some more time in terms of decision but still it is efficient in taking decisions. The parameters is learned very well by our agent that is why it performs good.

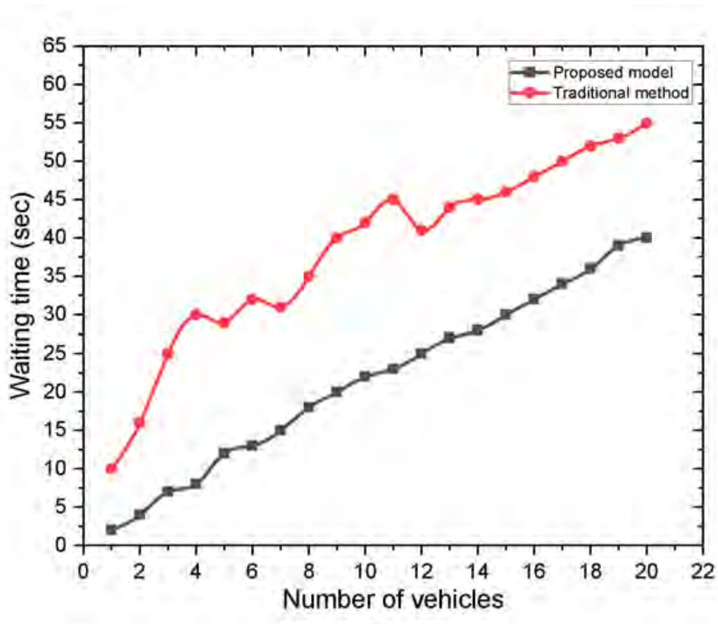

Figure 5.3: The waiting time of Q agent vs CF based recommendation

5.0.6 Accepted Recommendations

We calculated the accepted responses from our parameters calculations and then try to visualize the responses over time. The below graph shows the accepted and rejected responses.

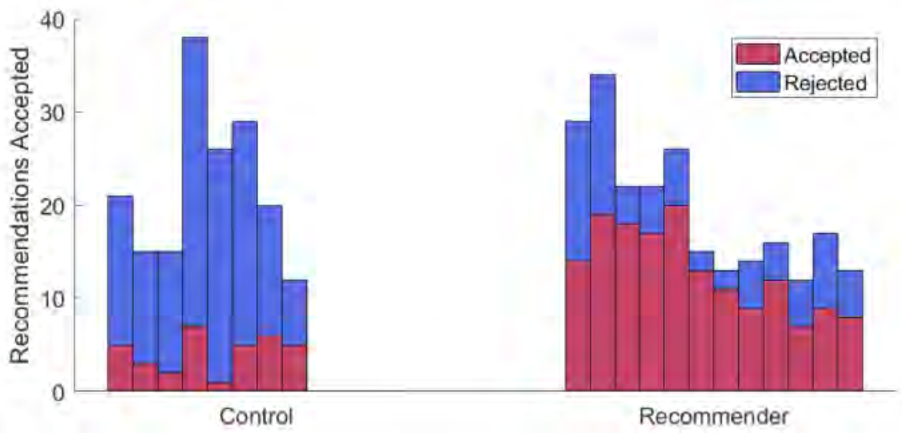

Figure 5.4: Accepted and Rejected Recommendations

Training Phase

We will talk about a portion of the measurements to assess our boundaries. Indeed, we proposed our custom test system like finding the way in the labyrinth for example ideal way finding the labyrinth climate. A portion of the significant measurement are: Score, cost, ε-eager qualities and ε-insatiable/arbitrary.
Score characterizes the presentation of the specialist, First 100 scenes the exhibition of way discovering labyrinth is poor as the operator isn't appropriately taken in the qual-

ities and its presentation is more awful than the irregular investigation done during preparing.
The expense is fundamentally alludes to the preparation cost of the calculation. After scene 10 the preparation stage begins and before that answer memory is develop by DQN. We utilized pre prepared organization, with the goal that our organization don't prepare neural organization during assessment.

Table 5.3: Evaluation Table

Metric	***Values***
Epochs	100
test steps	4e4
repeats	5
environment	custom
Algorithms	Q, DQN, A3C

The choice of picking the Q learning or Deep Q learning relies upon our usable boundaries. For test approach the Q execution is fine as our utilized questionable cases are low yet for bigger situation the calculation measurements will be bigger.

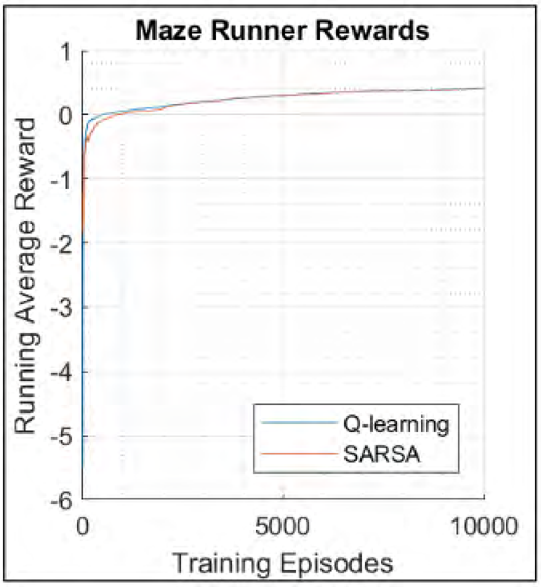

Figure 5.5: decision taking over episodic tasks

5.1 Research Overview

DL is now being used in every aspect from day to day problem to specific needs of people DL methods can be adapted. This is also seen in terms of the semantic image segmentation problem where many new DL algorithms based on previous ones for more accuracy, faster convergence and lower processing power are coming out. System implementers now have a huge array of options from these DL methodologies to

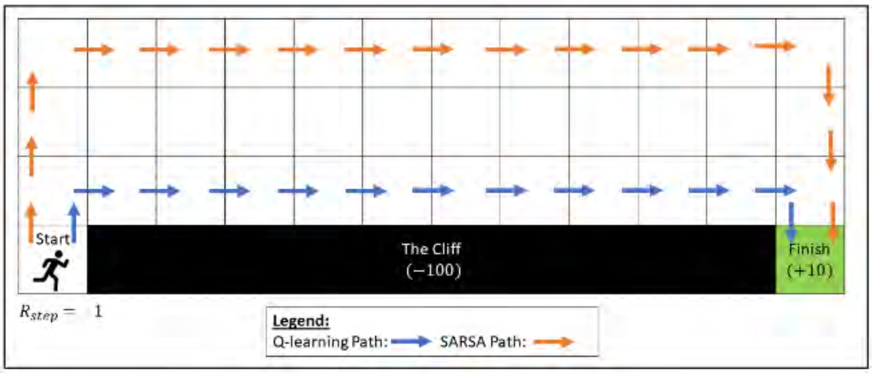

Figure 5.6: Path selected by Q learning agent and SARSA agent

use in their specific solutions.

DL now has come a long way to go much deeper tackling vanishing gradient problems and making much more complex, deeper NN's possible. However the major problems with NN's still remain, particularly CNN training becomes pretty much useless if there is no GPU or dedicated parallel processors available.

5.2 Recommendation and Future Work

Our proposed framework can allocate free parking spaces among users where through an Android UI application commuters able to see and check the parking status. This thing is very much essential for a parking system. Then, we try to ensure security for the parking system. Using license plate recognition system people can easily booked the parking place and get verified using the system. using DNN will ensure accuracy in terms of license plate detection and recognition of the users. Finally, we come to a solution of recommending a parking places. The number of parking areas can be high, but a recommender system can be essential for these kinds of scenarios where lots of requests come altogether and people do not know what and where to find a car. By using RL we can tackle the uncertain cases in a parking system and make an efficient parking recommender system.
Our future plan is to reduce the number of sensors. We try to catch real life scenario based parking spaces. We will utilize the security camera to identify the free parking places in a real life scenario. We will apply more advanced NN algorithms to detect license plates. To improve our RL algorithm we need to integrate the DL with our RL algorithms. Then the overall prototype will be a good demonstration for making a better parking system.

Bibliography

[1] M. O. Hasan, M. M. Islam and Y. Alsaawy, "Smart Parking Model based on Internet of Things (IoT) and TensorFlow," 2019 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, Malaysia, 2019, pp. 1-5, doi: 10.1109/ICSCC.2019.8843651.

[2] Hasan M.O., Ahmed K.R., Islam M.M. (2020) Parking Recommender System Using Q-Learning and Cloud Computing. In: Bhuiyan T., Rahman M., Ali M. (eds) Cyber Security and Computer Science. ICONCS 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 325. Springer, Cham.

[3] Koster A., Oliveira A., Volpato O., Delvequio V., Koch F. (2014) Recognition and Recommendation of Parking Places. In: Bazzan A., Pichara K. (eds) Advances in Artificial Intelligence – IBERAMIA 2014. IBERAMIA 2014. Lecture Notes in Computer Science, vol 8864. Springer, Cham.

[4] Dong, S., Chen, M., Peng, L., Li, H.: Parking rank: A novel method of parking lots sorting and recommendation based on public information. In: 2018 IEEE International Conference on Industrial Technology (ICIT), pp. 1381-1386. IEEE Press, Lyon (2018).

[5] Y. Li.Reinforcement learning applications.arXiv preprintarXiv:1908.06973, 2019.

[6] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Dawei Yin, Yihong Zhao, and Jiliang Tang. 2017. Deep Reinforcement Learning for List-wise Recommendations. arXiv preprint arXiv:1801.00209 (2017).

[7] Schafer J.B., Frankowski D., Herlocker J., Sen S. (2007) Collaborative Filtering Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg

[8] R. Chen, Q. Hua, Y. Chang, B. Wang, L. Zhang and X. Kong, "A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks," in IEEE Access, vol. 6, pp. 64301-64320, 2018, doi: 10.1109/ACCESS.2018.2877208.

[9] Y. Zhang, K. Meng, W. Kong, Z. Y. Dong and F. Qian, "Bayesian Hybrid Collaborative Filtering-Based Residential Electricity Plan Recommender System," in IEEE Transactions on Industrial Informatics, vol. 15, no. 8, pp. 4731-4741, Aug. 2019, doi: 10.1109/TII.2019.2917318.

[10] Hu B., Li Z., Chao W. (2012) Data Sparsity: A Key Disadvantage of User-Based Collaborative Filtering?. In: Sheng Q.Z., Wang G., Jensen C.S., Xu G. (eds) Web Technologies and Applications. APWeb 2012. Lecture Notes in Computer Science, vol 7235. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-29253-8_55

[11] M. Patil, V.N Bhonge, "Wireless sensor network and RFID for smart parking system", In-ternational Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 4, pp. 188-192.

[12] A. Kianpisheh et al., "Smart Parking System Architecture Using Ultrasonic Detector", In-tern'1 Journal of Software Engineering and its Applications, vol. 6, no. 3, pp. 51-58, July 2012.

[13] E. Karbab, D. Djenouri, S. Boulkaboul, A. Bagula, "Car park management with networked wireless sensors and active RFID", 2015 IEEE International Conference on Elec-tro/Information Technology (EIT), pp. 373-378, 2015, May.

[14] Z. Qadir, F. Al-Turjman, M. A. Khan and T. Nesimoglu, "ZIGBEE Based Time and Energy Efficient Smart Parking System Using IOT," 2018 18th Mediterranean Microwave Sympo-sium (MMS), Istanbul, 2018, pp. 295-298.

[15] 9. P. Chippalkatti, G. Kadam and V. Ichake, "I-SPARK: IoT Based Smart Parking System," 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, 2018, pp. 473-477.

[16] L. Mhatre and N. Rai, "Integration between wireless sensor and cloud," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 779-782.

[17] R. K. Dwivedi and R. Kumar, "Sensor Cloud: Integrating Wireless Sensor Networks with Cloud Computing," 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gorakhpur, 2018, pp. 1-6.

[18] B. M. Mahendra, S. Sonoli, N. Bhat, Raju and T. Raghu, "IoT based sensor enabled smart car parking for advanced driver assistance system," 2017 2nd IEEE International Confer-ence on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 2188-2193.

[19] P. Sadhukhan, "An IoT-based E-parking system for smart cities," 2017 International Con-ference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp. 1062-1066.

[20] F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 755-758.

[21] Islam, M., Khan, Z. and Alsaawy, Y. (2018). An Implementation of Harmonizing Internet of Things (IoT) in Cloud. Lecture Notes of the Institute for Computer Sciences, Social Infor-matics and Telecommunications Engineering, pp.3-12.

[22] TensorFlow Homepage, https://www.tensorflow.org/, [last accessed on 15 February,2019]

[23] Keras Homepage, https://keras.io/, [last accessed on 11 March, 2019]

[24] MIT Deep Learning Homepage, https://deeplearning.mit.edu, [last accessed on 10 January, 2019]

[25] Fang, J., Ma, A., Fan, H., Cai, M., Song, S.: Research on smart parking guidance and parking recommendation algorithm. In: 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 209-212. IEEE Press, Beijing (2017). doi:10.1109/ICSESS.2017.8342898

[26] Srisura, B., Wan, C., Sae-lim, D., Meechoosup, P., Mar Win, P.: User Preference Recommendation on Mobile Car Parking Application. In: 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 59-64. IEEE Press, Bamberg (2018). doi:10.1109/MobileCloud.2018.00017

[27] Yavari, A., Jayaraman, P. P., Georgakopoulos, D.: Contextualised service delivery in the Internet of Things: Parking recommender for smart cities. In: IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 454-459. IEEE Press, Reston, VA (2016). doi:10.1109/WF-IoT.2016.7845479

[28] Srisura, B., Avatchanakorn, V.: Periodical Mobile Recommendation Toward Parking Conflict Reduction. In: IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), pp. 397-402. IEEE Press, Tokyo, Japan (2019). doi:10.1109/IEA.2019.8715025

[29] Yuan, N. J., Zheng, Y., Zhang, L., Xie, X.: T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. In: IEEE Transactions on Knowledge and Data Engineering, vol. 25, pp. 2390-2403. IEEE Press (2013). doi:10.1109/TKDE.2012.153

[30] Fu, J., Chen, Z., Sun, R., Yang, B.: Reservation based optimal parking lot recommendation model in Internet of Vehicle environment. In: China Communications, vol. 11, pp. 38-48. IEEE Press, (2014). doi:10.1109/CC.2014.6969792

[31] Koster, A., Oliveira, A., Volpato, O., Delvequio, V., Koch, F.: Recognition and Recommendation of Parking Places. In: Bazzan A., Pichara K. (eds) Advances in Artificial Intelligence – IBERAMIA 2014. LNCS, vol 8864, pp. 675-685. Springer, Cham (2014).

[32] Shinde, S., Patil, A., Chavan, S., Deshmukh, S., Ingleshwar, S.: IoT based parking system using Google. In: International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 634-636. IEEE Press, Palldam (2017). doi:10.1109/I-SMAC.2017.8058256

[33] Rummery, G.A., Niranjan, M.: On-line Q-learning using connectionist systems. Department of Engineering, University of Cambridge (1994)

[34] Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM 40(3), 56–58 (1997)

[35] Chi, C.Y., Tsai, R.T.H., Lai, J.Y., Hsu, J.Y.J.: A reinforcement learning approach to emotion-based automatic playlist generation. In: 2010 International Conference on Technologies and Applications of Artificial Intelligence, pp. 60–65. IEEE (2010)

[36] Konstan, J.A., Riedl, J., Borchers, A., Herlocker, J.L.: Recommender systems: a grouplens perspective. In: Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08), pp. 60–64 (1998)

[37] Wu, T., Tsai, P., Hu, N., Chen, J.: Research and implementation of auto parking system based on ultrasonic sensors. In: International Conference on Advanced Materials for Science and Engineering (ICAMSE), pp. 643-645. IEEE Press, Tainan (2016). doi:10.1109/ICAMSE.2016.7840267

[38] Shao, Y., Chen, P., Cao, T.: A Grid Projection Method Based on Ultrasonic Sensor for Parking Space Detection. In: IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 3378-3381. IEEE Press, Valencia (2018). doi:10.1109/IGARSS.2018.8519022

[39] Sutton, R.S., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)

[40] Islam, M. M., Khan, Z., Alsaawy, Y.: An Implementation of Harmonizing Internet of Things (IoT) in Cloud. Lecture Notes of the Institute for Computer Sciences, Social Infor-matics and Telecommunications Engineering, pp.3-12. Springer, Cham (2018).

[41] Taghipour, N., Kardan, A., Ghidary, S.S.: Usage-based web recommendations: a reinforcement learning approach. In: Proceedings of the 2007 ACM conference on Recommender systems, pp. 113–120. ACM (2007)

[42] Bu, X., Rao, J., Xu, C.: Coordinated Self-Configuration of Virtual Machines and Appliances Using a Model-Free Learning Approach. In: IEEE Transactions on Parallel and Distributed Systems, vol. 24, pp. 681-690, IEEE (2013). doi:10.1109/TPDS.2012.174

[43] Oshii, K., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In: ISMIR, vol. 6, p. 7th (2006)

[44] Song, Y., Dixon, S., Pearce, M.: A survey of music recommendation systems and future perspectives. In: 9th International Symposium on Computer Music Modeling and Retrieval (2012)

[45] Shani, G., Heckerman, D., Brafman, R.I.: An MDP-based recommender system. J. Mach. Learn. Res. 6(Sep), 1265–1295 (2005)

[46] Suhr, J.K., Jung, H.G., Bae, K., Kim, J.: Automatic free parking space detection by using motion stereo-based 3D reconstruction. Machine Vision and Applicatons 21(2), 163–176 (2010)

[47] P. Wei, S. Xia, R. Chen, J. Qian, C. Li and X. Jiang, "A Deep-Reinforcement-Learning-Based Recommender System for Occupant-Driven Energy Optimization in Commercial Buildings," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6402-6413, July 2020, doi: 10.1109/JIOT.2020.2974848.

[48] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wier- stra, and M. Riedmiller, “Playing atari with deep reinforcement learn- ing,” arXiv preprint arXiv:1312.5602, 2013