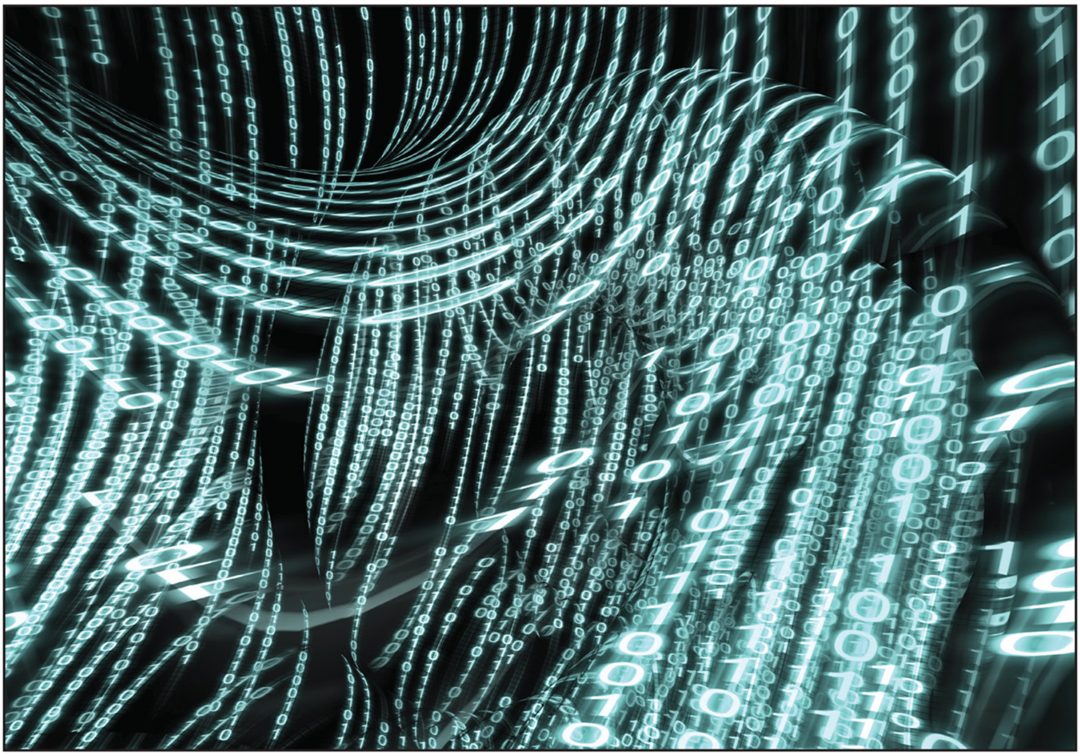
INTERNATIONAL JOURNAL OF

Software Science and Computational Intelligence



IGI PUBLISHING

Publisher of Peer-Reviewed, Timely, and Innovative Research Since 1988

www.igi-global.com

International Journal of Software Science and Computational Intelligence (IJSSCI)

Volume 12, Issue 4
October - December 2020

Brij Gupta - National Institute of Technology, Kurukshetra, India
Andrew W.H. Ip - University of Saskatchewan, Canada

9781799806127

International Journal of Software Science and Computational Intelligence

Volume 12 • Issue 4 • October-December 2020 • ISSN: 1942-9045 • eISSN: 1942-9037

IGI PUBLISHING
AN IMPRINT OF IGI GLOBAL
WWW.IGI-GLOBAL.COM

Call for Articles

International Journal of Software Science and Computational Intelligence

Volume 12 • Issue 4 • October-December 2020 • ISSN: 1942-9045 • eISSN: 1942-9037

MISSION

The **International Journal of Software Science and Computational Intelligence (IJSSCI)** is a transdisciplinary, archived, and rigorously refereed journal that publishes and disseminates cutting-edge research findings and technological developments in the emerging fields of software science and computational intelligence, as well as their engineering applications. The latest developments in computer science, theoretical software engineering, cognitive science, cognitive informatics, and intelligence science, and the crystallization of accumulated knowledge by the fertilization of these areas, have led to the emergence of a transdisciplinary and convergence field known as software and intelligence sciences.

Software Science is a discipline of enquiries that studies, theoretical framework of software as instructive and behavioral information, which can be embodied and executed by generic computers in order to create expected system behaviors and machine intelligence. The discipline of software science studies the common objects in the abstract world such as software, information, data, concepts, knowledge, instructions, executable behaviors, and their processing by natural and machine intelligence. From this view, software engineering is the engineering discipline that applies software science theories and methodologies to efficiently, economically, and reliably organize and develop large-scale software systems.

Computational Intelligence is a human enquiry of both natural and machine intelligence at the reduction levels of biology, physiology, psychology, cognition, computation, logic, and philosophy from the bottom up. Intelligence science is naturally interdisciplinary in vertical and based on reductionism and is transdisciplinary in horizontal based on holism. Intelligence science studies theories and models of the brain at all levels, and the relationship between the concrete physiological brain and the abstract soft mind. Intelligence science is a new frontier with the fertilization of biology, psychology, neuroscience, cognitive science, philosophy, information science, computer science, anthropology, and linguistics.

COVERAGE AND MAJOR TOPICS

The topics of interest in this journal include, but are not limited to:

Automatic software code generation technologies • Autonomic/autonomous systems • Autonomous agent systems • Autonomous machine learning systems • Cognitive complexity of software • Cognitive computers • Cognitive machines that think and feel • Future generation computers • Granular computing • Hybrid man-machine systems • Hyper programming • Intelligent software engineering • Novel computing methods • Novel intelligence simulation systems • Novel memory devices • Soft computing • Concept algebra for knowledge modeling • Denotational vs. analytic mathematics • Formal description of cognitive processes • Formal description of the brain • Formal inference processes • Fuzzy/rough sets • Mathematical foundations of software • Mathematical models of machine intelligence • Mathematical models of natural intelligence • Mathematical models of the brain and mind • Mathematical structures for software modeling • Process algebra for behavioral modeling • Real-Time Process Algebra (RTPA) • System algebra for complex system modeling • Universal mathematic models of software • Visual semantic algebra • Autonomous computing • Cognitive informatics • Cognitive informatics foundations of the brain • Cognitive mechanisms of the brain and mind • Formal language theories • Functional models of the brain • Instructive information foundations of software • Intelligent behavioral foundations of software • Knowledge representation methodologies • Logical models of the brain • Neural informatics • Non-language centered programming • Software simulations of the brain • Software vs. brain processes • Theories for computational intelligence • Transdisciplinary theories shared by software and intelligence science

ALL INQUIRIES REGARDING IJSSCI SHOULD BE DIRECTED TO THE ATTENTION OF:

Brij Gupta, Editor-in-Chief • IJSSCI@igi-global.com

ALL MANUSCRIPT SUBMISSIONS TO IJSSCI SHOULD BE SENT THROUGH THE ONLINE SUBMISSION SYSTEM:

http://www.igi-global.com/authorseditors/titlesubmission/newproject.aspx

IDEAS FOR SPECIAL THEME ISSUES MAY BE SUBMITTED TO THE EDITOR(S)-IN-CHIEF

PLEASE RECOMMEND THIS PUBLICATION TO YOUR LIBRARIAN

For a convenient easy-to-use library recommendation form, please visit:
http://www.igi-global.com/IJSSCI

Table of Contents

International Journal of Software Science and Computational Intelligence

Volume 12 • Issue 4 • October-December-2020 • ISSN: 1942-9045 • eISSN: 1942-9037

Special Issue on Smart Driver Assistance Systems and Autonomous Driving

Guest Editorial Preface

The *International Journal of Software Science and Computational Intelligence* is indexed or listed in the following: ACM Digital Library; Bacon's Media Directory; Cabell's Directories; DBLP; GetCited; Google Scholar; INSPEC; JournalTOCs; MediaFinder; The Standard Periodical Directory; Ulrich's Periodicals Directory

Guest Editorial Preface

Special Issue on Smart Driver Assistance Systems and Autonomous Driving

Larbi Guezouli, LaSTIC Laboratory, Department of Computer Science, University of Batna 2, Batna, Algeria

The National Highway Traffic Safety Association was created in 1970. Its main role was to effect safety standards on vehicles. Since its creation, manufacturers have been competing to seriously think about creating systems that will help drivers drive safely and without accidents. The role that driver assistance systems can play in reducing road accidents and make driving safer cannot be overstated.

Today, the obligation to use data-mining in driver assistance systems seems to be a reality. The use of data mining affects many areas of research, where a huge amount of information needs to be manipulated.

During the last decade, several research studies on this issue have been launched to find optimal solutions and many approaches have been proposed.

To help the driver in his driving task, we will deal with real-time information processing in order to facilitate the prediction of the dangers related to the different driving parameters. For this, several questions will be asked:

- How can we manage the vast amount of information captured by different devices placed on the car?
- How do we filter pertinent information from captured information?

This special issue on: "*Smart Driver Assistance Systems and Autonomous Driving*" of the *International Journal of Software Science and Computational Intelligence* (IJSSCI) focuses on information processing in driver assistance systems. It contains five original and well-reviewed papers.

The topics covered in this special issue include: Driver assistance systems; scene understanding; software architectures for autonomous vehicles; driver-vehicle interaction and assisted driving; computational intelligence; image processing; information retrieval; knowledge representation; machine learning; neural computing; signal processing; information and communication technologies for development.

In the first paper entitled "*The Optimal Path Finding Algorithm Based on Reinforcement Learning*", *Ganesh Khekare, Pushpneel Verma, Urvashi Dhanre, Seema Raut and Shahrukh Sheikh* propose an algorithm which takes into consideration multiple objectives and provides an optimal route solution. It is based on reinforcement learning and capable of deciding the optimal route on its own.

Meriem Benadda and *Ghalem Belalem* propose the service "HAaaS" in "*Improving Road Safety for the Drivers Taken Malaise and Sleepiness Behind the Wheel Using Vehicular Cloud Computing and Body Area Networks*". In their paper, they present a new Vehicular Cloud Computing service, named "HAaaS", based on BANs to detect, monitor and manage driver's malaise and provides a cooperation support for the driver rescue. The aim is to reduce the number of accidents, material and human damage as well as the time and fuel lost in traffic jams.

Rinat Galiautdinov considers the possibility of applying modern IT technologies to implement information processing algorithms in UAV motion control system in his paper "*Information processing systems in UAV based on Bayesian filtering in conditions of uncertainty*". Filtration of coordinates and motion parameters of objects under a priori uncertainty is carried out using nonlinear adaptive filters: Kalman and Bayesian filters. The author considers numerical methods for digital implementation of nonlinear filters based on the convolution of functions, the possibilities of neural networks and fuzzy logic for solving the problems of tracking UAV objects.

In his paper "*Experience-based approach for cognitive vehicle research*", *Hironori Hiraishi* discusses an experience-based approach to cognitive vehicle research. He analyzed the driving data he collected, and he developed models using the Cognitive Qualitative Analysis and Modeling tool (QCAM) that he has developed.

The last paper is presented by *Houcine Matallah, Ghalem Belalem* and *Karim Bouamrane* and entitled "*Evaluation of NoSQL Databases - MongoDB, Cassandra, HBase, Redis, Couchbase, OrientDB*". In this paper, authors develop a comparative study about the performance of six solutions NoSQL, very employed by the important companies in the IT sector: MongoDB, Cassandra, HBase, Redis, Couchbase and OrientDB. They provide some answers to choose the appropriate NoSQL system for the type of data used and the type of processing performed on that data.

We would like to thank all the authors for having submitted their research results for publication in this special issue. We toughly believe that the papers published here reveal progress in the topic of smart driver assistance systems and autonomous driving. Also, we are very grateful to the collaboration of the reviewers which influence on the final quality by their expertise. We sincerely thank Prof. Brij Gupta and Prof. Andrew Wai Hung Ip, Editors-in-Chief of International Journal of Software Science and Computational Intelligence (IJSSCI), for giving us the opportunity to prepare this special issue and their advice throughout this project. The support of the publisher with technical issues was also very welcomed and assured good progress.

We hope that reading these high-quality papers will inspire you to make your own submissions to future issues.

Larbi Guezouli
Guest Editor
IJSSCI

International Journal of Software Science and Computational Intelligence
Volume 12 • Issue 4 • October-December 2020

The Optimal Path Finding Algorithm Based on Reinforcement Learning

Ganesh Khekare, G H Raisoni College of Engineering, Nagpur, India
https://orcid.org/0000-0002-1687-4699

Pushpneel Verma, Bhagwant University, Ajmer, India

Urvashi Dhanre, Government College of Engineering, Nagpur, India
https://orcid.org/0000-0002-3989-7770

Seema Raut, G H Raisoni Institute of Engineering and Technology, Nagpur, India
https://orcid.org/0000-0002-4081-5306

Shahrukh Sheikh, G H Raisoni College of Engineering, Nagpur, India
https://orcid.org/0000-0002-2877-6279

ABSTRACT

Urbanization has been extensively increased in the last decade. In proportion, the number of vehicles throughout the world is increasing broadly. The detailed survey of available optimal path algorithms is done in this article, and to ease the overall traveling process, a dynamic algorithm is proposed. The proposed algorithm takes into consideration multiple objectives like dynamic traffic density, distance, history data, etc. and provides an optimal route solution. It is hinged on reinforcement learning and capable of deciding the optimal route on its own. A comparative analysis of the proposed algorithm is done with a genetic algorithm, particle swarm optimization algorithm, and the artificial neural networks algorithm. Through simulation results, it is proved that the proposed algorithm has better efficiency, decision making, and stability. It will ease the driver's headache and make the journey more comfortable with traffic less short distance routes that will minimize overall travel time making a positive impact on traffic jams, accidents, fuel consumption, and pollution.

KEYWORDS

Dynamic Algorithm, Dynamic Decision Making, Intelligent Transportation System, Multiple Objectives, Reinforcement Learning, Shortest Path Algorithm, Traffic Density

INTRODUCTION

Smart city concept has been evolved in recent years. 2007 was the first year where the trend has been changed. People situated in urban areas increased as compared to rural. As per the United Nations report in 2018, approximately fifty-five percent of people in the world live in urban areas (Glaeser,

DOI: 10.4018/IJSSCI.2020100101

2020). By the end of 2050 sixty-eight percent, people are expected to be urbanized. Urbanization rapidly increasing in Asia and Africa region as compared to others, with India and China making the main difference. The trend of urbanization affected all the things right from transportation, smart cities, health facilities, automation, etc. The number of vehicles around the world is also increasing briskly. Currently, 1.2 billion vehicles are running on the roads and by the year 2035, it would become 2 billion. An increase in vehicles (Guidoni et al., 2020) creates problems like congestion, traffic jams, accidents, parking issues, pollution, economical investment in infrastructure building, etc.

An intelligent traffic system (Zhang et al., 2019) comes with intelligent algorithms for optimal pathfinding. In case of emergency situations drivers can't take decisions accurately (Jhiduk, 2014) but with the help of intelligent algorithms they can avoid situations like accidents, jams, roadblock, etc. that's why the novel algorithm proposed in this article can intelligently provide the optimal path considering multiple parameters. The route planning of the intelligent path algorithm is based on the current traffic scenario by considering history data, cost, etc. It is based on the continuous feedback taken from various resources and the optimal path intelligently determining method. It is partitioned into two ways, local route planning, and global route planning (Zhang et al., 2018). Global route planning works on the concept of optimization and survey providing method using local data in case of a well-known data house to know the possible areas and optimal routes (Zhang et al., 2018). As route provided by the global route generator is not the final one as it does not consider multiple objects during calculations. So the local route generator must be based on local traffic data (Zhang et al., 2019). The local and global route generator algorithms are correlated. In some cases, local information is gathered with the help of sensors and scenario is analyzed (Zhang et al., 2019). The static algorithms like A* algorithm are the quickest ones to provide an intelligent optimal route in state space. But static algorithm will not serve the purpose as traffic scenario is dynamically changing. So we need to switch on other technological algorithms. Reinforcement Learning is a vital innovation of upcoming Machine Learning technology. Reinforcement learning provides the facility with the machine to take the decisions the way humans do (Ren, 2017). The main advantage of using reinforcement learning is to provide long term solutions by considering the current state scenarios. It takes into consideration the environmental updating and can learn from the errors made in the past (Zhang et al., 2019). The optimal route for a traveler can be provided by considering multiple parameters like minimum distance, weight, total journey time, various performance parameters, etc. In this article, the novel dynamic algorithm for optimal pathfinding has been proposed which is based on reinforcement learning with multi objectives parameters. By the end of the year, 2050 the number of vehicles will become double i.e. 2.5 billion. In the metro cities like Bangalore in India, people spend more than half travel time in jams waiting for traffic roads to be clear. Sometimes even google map doesn't show any road is blocked due to repairing work, accidents, jams, etc. The proposed algorithm provides an optimal path to the driver within a fraction of second so that traveling can be done with ease, comfort, in minimum time and with the shortest distance. It also overcomes the problem that arises due to sudden dynamic changes in traffic. It will reduce traffic jams, accidents, fuel consumption perhaps pollution.

The remaining paper is presented as follows. In chapter II detailed survey of available optimal path algorithms has been done. In chapter III system design and the proposed algorithm are explained. Chapter IV devoted to the experimental setup. In chapter V results are discussed. In chapter VI conclusion has been made.

LITERATURE REVIEW

In (Miller, 2009) designed a model to collect information from real-time traffic scenarios with advanced tools like detectors, video cameras, nanosensors, VANET (Khekare & Sakhare, 2013), V2I, acoustic signal (Borkar et al., 2016), etc. The presented algorithm which provides the shortest path based on Pre-computed data (Khekare, 2014). The help of mobile nodes and already available infrastructure is taken to collect information. The proposed model shows the location of a vehicle but unable to identify the overall traffic in the area.

There are limitations with loop detector or video cameras:

- It can identify the exact location of the vehicle but unable to identify overall traffic on the road;
- To provide the accuracy of the location of the vehicle, require a huge number of detectors. Distance between two detectors is inversely proportional to the accuracy of location finding. To install so many detectors overall cost increases;
- Maintenance of detectors after installation. If some detectors not maintained/working properly, directly affect the reliability;
- If the speed of the vehicle is more than the detector's detecting capacity, then that vehicle cannot be traced;
- In bad weather conditions detector is unable to detect objects properly;
- Power consumption/supply.

So it always better to collect data through GPS rather than relying on detectors or cameras.

The algorithm for finding the congestion-free path must be superior so that it must provide the best path within the fraction of the second. Most congested cities like Bangalore in India has so congested traffic that the average person spends an extra 243 hours every year just because of unwanted traffic. To avoid the latency period algorithm must be perfect. Unwanted latency may affect the various parameters. In a Vehicle to Infrastructure pattern, data is updated continuously so the algorithm is to be implemented which will provide the optimized path concerning the current traffic scenario. To calculate the best route between source and destination in a directed graph, algorithms are divided into three different groups: Static Group, Dynamic Group, Pre Computed Group.

The static group of the algorithm include Johnson's Algorithm and Floyd Warshall's group, both compute the shortest path from each node present in graph to every other node available in the network, pairwise. Floyd-Warshall's Algorithm calculates the best route between source and destination in O(N3). Unlike Johnson's Algorithm which rearranges the cost of edges from time to time which provides the exact weight of each edge and if negative weight exists, cancels it by using Bellman-Ford's Algorithms (Ehrgott et al., 2012), and after that making use of Dijkstra's Algorithm (Dijkstra, 1959) from each node to find out congestion-free shortest path. Dijkstra's Algorithm has running time developed with minimum priority queue is O(NlgN + C), resulting in overall running time for Johnson's Algorithm of O(N2lgN + NC), which is giving better result than Floyd-Warshall's. Comparison of various algorithm concerning running time is as shown in Table 1.

In (Amar et al., 2016) various behavioral driven systems also required an efficient algorithm for finding the optimal path. It becomes important to find out optimal considering huge traffic in urban areas. It becomes the mainstream of research. Initially, Dijkstra's algorithm is widely used to find out the optimal path from source to destination. This algorithm works on the simple concept, it

Table 1. Shortest route algorithms comparison according to running time

	Naïve (Johnson	The Dynamic All-Pairs Shortest Path	Pre-Computed Constantly Updating	Pre-Computed Constant Query	Hybrid Pre-Computed
Pre computation	NA	NA	0(N2C!)	0(N2C!)	0(N2C!)
Update Edge	0(N2 log N + NC)	0(N2 log3N)	0(1)	0(N2mlogm)	0(N2m)
Retrieve Fasted Path	0(1)	0(1)	0(N2 m)	0(1)	0(m)

finds all the possible routes from source to destination and calculates their cost and provide the final path with the minimum cost. Based on the Dijkstra algorithm various advanced algorithm has been implemented. Lots of problem definitions come up aligned with finding the shortest path algorithm. One of the problems is associated with the Vehicle Routing Problem. Here (Solomon, 1987) weight from source to destination is calculated considering the time factor and final path is provided to the traveler and this algorithm is called Vehicle Routing Problem (Kim et al.,2005) with Time bounds. It is a multi-objective problem (DeviPriya, 2020) as it reduces the total weight of the path as traffic increases. The driver-centric algorithm is needed to find the minimum cost problem.

To provide the shortest path various algorithms have been implemented that may be dynamic (Hellinga et al., 1999) or static (Dijkstra, 1959). Nowadays various parameters came into existence with lots of urbanization like security, safety, easiness, etc. Each one of these parameters is considered as separate objective function and the final path is provided by using the multi-objective function. In (Blue et al., 1997) proposed an algorithm based on two objectives route-finding method. The two objectives are minimum time and travel complexity. Travel complexity calculated based on lane shifting, route changing, movements like merging and weaving. The straight road has a complexity indicator as zero whereas the U-turn is denoted by complexity index half. The final path here is provided by performing some iteration and calculations between travel complexity and time. The same kind of approach is applied to a bicycle where both objectives are considered separately. In (Raith et al., 2014), Raith et al. considered toll fee and time to travel as two separate objectives to provide a solid solution and named it as Multi-objective Traffic Assignment. Equilibrium concept is used in this algorithm to provide a balance between time and toll fee considering user requirements. So far we have discussed bi-objective problems or algorithms (Gupta, 2019).

The research articles discussed so far creates a separate objective function as per the requirement or demand. For the multi-objective function, for finding the optimized path feasibility may decrease. One solution for that to preprocess various small objective functions separately into a generalized function. For multi objective function this method will work very well. In the end, by combining various generalized functions, and the ultimate final generalized function provides the perfect solution. This method is known as multi objective-based vehicle routing.

In (Chen et al., 2010) proposed a stabilized weighted method where the weighted sum of duration value and toll fee value is calculated. Solution provided in this research article mostly look like bi-objective function. The weight and toll fee to each road is provided and labeled as the Dijkstra algorithm. The route with minimum weight is provided as the final path.

Various researchers focused on the behaviors and experience of the driver during the entire travel time. In (Ben-Akiva et al., 1991) a method is proposed that formulates the instinct behaviors of the traveler. In this driver encapsulates various constraints like source, destination, and driver's behavior, fees, and travel period. These constraints are processed to provide the ultimate driver's behavior. In (Adler et al. 1993) proposed a method to calculate driver's attitude and its effect on road scenario. In (Feng et al., 2010), an analysis model is proposed where the driver's behavior is identified and concerning behavior path selection is decided based on Bayesian theory and the conclusion making theory. While selecting the final path preference of various parameters like weight, speed, etc. is decided by the driver's behavior. Preference is decided one per path. In (Arentze, 2013) designed a car looks like a mobile robot that can rotate in a bidirectional manner in open space. The algorithm designed shows better results than the ReedsShepp algorithm. The whole space of the shortest path is divided into several segments and compared it with the various number of elements.

In (Desaulniers et al., 1995) driver decides the preferences and concerns that the model works. This is called an adjustable concretize trip data model which helps the ATIS model to act concerning the driver's requirement. Here also Bayesian-based algorithm is used and repeated sampling is performed. The centralized server handles multiple queries at a time and efforts are made to the ATIS system as per driver's history data of road selections. In (Yang et al., 2014) proposed a model that calculates the difference between expected times required for the vehicle to reach from source to destination and actual time taken by the vehicle to reach due to the driver's decisions. Analyze the driver's behavior by the changes made by the driver in the path during travel. Passengers are not informed in advance about these predictions and concerning predictions made by the driver results generated. In this system, the ATIS model uses the data and made the decision and if in run time traffic changes traveler is aware of this but not the system. In (Ben-Elia et al., 2010) as per the tolerance ratio of the passengers' various factors like safety, cost, and easiness are taken into consideration in addition to driver's behavior. By taken into consideration these all aspects the final decision is made. The most important is to provide accurate data to the traveler in a quick time so that decision making becomes easier. For decision making various simulation tools are available (Illenberger et al., 2011). Risk factors values are provided by travelers. With the capacity of travelers to take risk decision is made.

Some ATIS systems decided by taking overall history data not focusing on the behaviors of a particular driver. Individual preferences are not given that much weightage. The passenger who provides source and destination is provided with the path with minimum risk factor (Wahle et al., 2002). One of the major issues with the ATIS system is that it provides the same solution to multiple users (Wahle et al., 2000) based on their geographical area based on the past decisions taken in the same area, but driver's current behaviors are sometimes neglected. This affects the overall performance of the system. Concentration is the most needed factor. Though some researchers tried to use the altruistic method (Chen, 2008) for path providing to overcome this problem (Hoefer, 2009), it degrades the overall efficiency of the system (Caragiannis et al. 2010). If travelers are not focusing on minimizing their distance of travel (Chen et al., 2011) in that case altruistic method will serve the purpose. The meaning of the word "behavior" is taken as satisfying the parameters decided by the travelers (Desai et al., 2013). The satisfaction of these parameters reflects the success ratio of the path provided is correct or not. ATIS system performance will be improved if behavior-based planning is considered. If traveler's decisions are taken by the proxy servers then it would be the best thing available in the market to serve the said purpose.

Traffic forecasting (Kubek, 2016) is also one of the main areas researchers are focusing on. Forecasting is needed to avoid jams. Information needed to users about traffic scenarios, delivery of items in the city (Barcelo, 2010). The data is gathered and processed and helpful information is provided to people (Herrera, 2010). Traffic forecasting is a subpart of Intelligent Transportation Systems (Treiber, 2012), provides information related to current traffic scenarios, travel time from source to destination, distance, etc. The decisions are made considering the current traffic scenario (Vlahogianni, 2013). Quick results are expected for the best decision making (Gentili, 2012). Decisions provided must be reliable so that user gets the assured (Hu, 2009). Providing traffic forecasting accurately is still a big challenge (Chan et al., 2012). Mostly due to the installation and maintenance of detectors (Zhang, 2014). The survey of various researchers on traffic forecasting shows that the proper method still not available. Dynamic changes in the traffic are not handled by these methods.

In (Yixiaohuang et al.,2016) proposed a system which observed the real-time speed data in fifteen minutes span daily for providing traffic jam solution. Similarly, in (Franceschetti et al., 2013) make use of comparatively fewer speed levels to recognize traffic scenarios in the city. Although in (Kim, 2020) researchers use sharp speed levels to identify the traffic the whole day (Ehmke et al., 2016). So by considering the speed of the vehicles and analyzing it at different levels, they used to recognize traffic in the city. But it has a limitation, it is not necessary if the road is free still drivers may use slow speed depending on the mood and vice versa is possible in some cases.

A large number of algorithms are available to provide the shortest path considering time as a major factor. Some of the well-known algorithms are as follows,

Dijkstra algorithm with the Bidirectional approach (Qian, 2016) finds the path both way till target is not achieved, starting from a source in the forward direction and the backward direction from a destination. The bi-directional technique is useful as it can be used with any other algorithm (Chmiel, 2019). Highway hierarchies (HHs) method (Sanders, 2006), considers a limited number of roads while calculating the shortest path. Whenever the user asks for the shortest path only these roads are considered for calculating the shortest path. This method is used for commercial purposes. It is somewhat similar to the heuristics approach. This algorithm is better as it provides the optimal route in a fraction of seconds. The updated version (Sanders, 2005) of Highway hierarchies uses weighted functions. The cost from each node is already calculated and kept in the form of data. This method is two times better than the traditional Highway hierarchies' algorithm.

Transit-based routing (Naghawi & Wolshon, 2010) algorithm focuses on neighbors' nodes of the source through lookups data and hence very fewer options are available to provide the shortest path. Reach-based routing (Gutman, 2004) algorithms are one step ahead of the Highway hierarchies (HHs) method, takes dynamic into considerations while calculating the optimal path. Due to dynamic nature a large amount of data is generated and the cost of processing this data increases, hence this algorithm is not used for bigger areas. Divider based multimodal algorithm (Schulz, 1999), first create the overlay graph and converts it into several subparts, where the best route in the original graph is the same as the best route in subgraphs (Bast et al., 2007). The same algorithm is also applied to various levels (Holzer, 2009). Where traffic changes dynamically the reconstruction of the whole graph is not required every time. But the time required to process data is much more in this method, space consumption is high, so this method cannot be applied to large data.

Edge-based algorithm (Lauther, 2004) calculates the weight of each edge and store the data of all nodes in some selected supernodes. Supernodes are denoted with the angular set. This algorithm is further extended to geometric sets (Wagner & Willhalm, 2003). Still optimized route calculations for every pair hosts is required.

Landmark A* algorithm (Goldberg & Harrelson, 2004) takes the already calculated weights of every vertex, constantly spread all over the graph. The algorithm provides the optimal path in a quick time and provides a comfortable time to travelers for decision making. It saves the data of all generated vertices in the provided memory.

Precomputed cluster distances (PCD), breaks complete graph into smaller subsets (Maue, 2010). Prune finding is possible with the PCD method, as it both upper as well as the lower level for weights. The efficiency of this method is quite better than the Prune search method and requires less memory for processing. In (Schultes & Sanders, 2007) proposed a dynamic algorithm with a better type of information preprocessing for finding the optimal path in large geographical areas, knows as Dynamic Highway-Node Routing (DHN). If any run-time changes occur it uses preprocessed data and provides the solution within a fraction of second. It is based on the normalization of the Separator-based multilevel algorithm for the highway-vertex routing and provides an optimal path in very little time. This algorithm fails to provide both types of settings dynamic and server-side.

The next level of routing methods are implemented upon linear and dynamic programming and called as an approximation and exact procedures. In (Dinitz & Itzhak, 2017) discussed a hybrid method having a base of the Dijkstra algorithm and Bellman–Ford, which improves the running cost of the Bellman–Ford algorithm for dispersing plan and negative weight of routes. In (Delling et al., 2011) discussed the optimal route method which can decide on real-time queries and provides a cost-efficient path in a very few times. In (Buchhold, 2018) proposed a system based on the equilibrium of traffic shades in the city. Equilibrium traffic analysis of each road in this city is done. These approaches make real-time things possible and relevant decision is made.

In (Angelelli et al., 2016) discussed the linear programming model which provides continuous path guidance for a proactive route guide that neutralizes traffic as well as journey inconveniences. In this approach, the driver is provided the road other than the shortest to avoid unnecessary jams. In (Rahmani, 2017) proposes the system which manages the balance of both short paths as well as comfortability. In (Lee, 2017) Gini coefficient is proposed to calculate the optimal path with reliability and less depression. In (Zeng & Wang, 2018) discusses the fuel-efficient traveling from the source to target. Dynamic programming is used to achieve the target. In this system, a decision is made on various objectives like time, cost, quality of roads, etc.

In (Sever et al., 2018) designed a hybrid approximate dynamic programming (ADP) where the optimal route is provided dynamically look ahead method and finding the approximate value. In (Hu et al., 2016) stated the ripple spreading algorithm (RSA) for the route optimization method. As compared to Dijkstra's algorithm which top-down approach, the bottom-up approach RSA algorithm gives better results with good efficiency. In (Hu et al., 2017) the RSA method provides dynamic path selection with route trajectory. In (Idri et al., 2017) considering time factor and multiobjective environment, a new optimal route algorithm is designed.

In (Sarbazi-Azad, et al., 2017) single pair and all pair problems are resolved by using GPU computing (Premkamal, 2020) power. All possible pairs (Patle et al., 2018) are obtained and after processing the final path is provided in a microsecond. The use of dynamic concepts (Yang et al., 2018) for providing optimal route solution has been increased in recent years (Zhao, 2018). But they consider the meaning of dynamic concerning mobile vehicles only.

Now the world is moving towards artificial intelligence and machine learning (Dib et al. 2017). Reinforcement learning is the main part of it. Various intelligent reinforcement algorithms came into existence in the last decade which can take decide on its own as we do. The instant–similar–based hybrid genetic algorithm (Pal, 2019) is described in for providing an efficient and reliable optimal route through speculative path infrastructure with correlation linking. The linking between edges and vertices is done by using a correlational coefficient (Marinakis, 2017). The genetic algorithm (Desuo et al., 2019) is designed with the meta-heuristics type and the neighborhood node is searched for the optimal route situation. In a hybrid particle swarm optimization algorithm is proposed that searches for neighboring vertices for the desired output. Constraints setting is done in this algorithm. In (Huang et al., 2017) particle swarm optimization pathfinding algorithm is presented based on Pareto optimal solution considering multiple objectives during the simulation. In (Paumard, 2020) designed a time-delay artificial neural network for providing optimal route with time constraints. The neural network identifies the vertices despite the gap during analysis (Jiang, 2018). Predictions (Khekare & Janardhan, 2017) of the future traffic scenario are done. Through literature review, it is cleared that there is a requirement of the shortest path algorithm that provides the optimal path considering multiple objectives, which also considers history data to make a decision on dynamic changing traffic scenarios and which is capable of taking the right decisions on its own through reinforcement learning.

In this paper comparison of the Genetic Algorithm, Artificial Neural Network and Particle Swarm Optimization are done with the proposed optimal route algorithm with reinforcement learning (ORAWRF) is done.

METHODOLOGY

Approach

Abjure the intelligent traffic system denoted by the directed Graph DG= (N, C), where a node $n \in N$ which shows the particular spot on the road and a cost $c \in C$ shows the specific distance on the road connecting two nodes. A route R on a directed graph DG= (N, C) is stated as follows:

$$R(x, y) = \{nx, \ldots, ny\} \;\Big|\; x \leq k \leq y, (nk, nk+1) \in C$$

The nodes on the road help to provide the congestion-free shortest path. As the number of nodes increases, accuracy increases, but it requires high bandwidth to transmit the messages of the best route. The cost between two nodes indicates the time required to reach from one node to another. Cost represents the minimum time required for a vehicle to travel from source vertex to destination vertex. Considering current traffic scenarios, to provide the congestion-free shortest path, three aspects must be taken into consideration:

- Current traffic scenario;
- Shortest distance from the source to destination;
- Processed history data;
- Multi-objectives identification;
- Dynamic decision making.

Considering these objectives improves the overall result. Current traffic scenarios can be gathered through various techniques already discussed in the literature review like detectors, video cameras, GPS, VANET, stochastic signals, V2I, etc. GPS is most preferable with reinforcement learning. History data is taken into consideration to avoid for accurate results. The shortest path is provided such that the vehicle would reach the destination with the shortest time and ease.

Proposed Method

Assume traveler M, wish to go from source node S to destination node D. Consider the congestion-free shortest path is selected from various possible paths denoted by set P. So one of the possible paths is denoted as P(S, D). For each $p \in P(S, D)$). A set of attributes $Ap(S, D)$. Ap shows the cost, $\delta p(S, D)$, between source to destination along with path p, Travel Duration $\tau_p(S, D)$ during path p, security indicator $\sigma p(S, D)$, traffic density indicator κ_p and relief indicator $\varphi_p(S, D)$,. Every path p is calculated asset of various sub-paths $Vp = \{v_1, v_2, \ldots., v_n\}$, where v_1 indicates the first i.e. source node and v_n indicates the last i.e. destination node. For every subpath $v_i \in Vp$. $A^{vi}{}_p$ is a set of attributes the same as path p. $\delta^{vi}{}_p$ is the distance to cover along on path p, Travel Duration $\tau^{vi}{}_p$, security indicator $\sigma^{vi}{}_p$, relief indicator $\varphi^{vi}{}_p$, and traffic density indicator $\kappa^{vi}{}_p$. The Travel duration $\tau^{vi}{}_p$, and relief indicator $\sigma^{vi}{}_p$ are depended on the current traffic scenario. It is calculated and denoted by the $E(\aleph)$ which gets the data from the wireless sensor network $\aleph$. A wireless sensor model consists of various sources through which data is gathered. During transportation M's dumping is a set of transportation updates $TX = \{tx_1, tx_2, \ldots, tx_x\}$. For each transportation update txi and path $p \in P$, a weight function is denoted as $\xi^{txi}{}_p = f(Vp)$. Processed history data with reinforcement learning is denoted with W.

The weight of the journey from *Source* to *Destination* is shown by ξp, for path selected p, the travel way taken tx, and the shortest congestion-free shortest path provided by using this algorithm is by considering various factors like environment, security, density, relief to the user, etc. These various parameters are shown by using $\Gamma M(S,D) = \{\gamma_1, \gamma_2, \ldots, \gamma_p\}$, γ_i indicates cost that passenger M allots to a provided attribute.

The passenger M's final optimal route then provided as follows:

$$pOpt = \min_{\forall\, p \in P} Y\left(S, D, p, Ap, \Gamma M\right)$$

whereas Υ indicates the method through which the best optimal path p_{Opt} is obtained.

This formula provides the best results with comparison with the available reinforcement algorithms.

EXPERIMENTAL SETUP

In this paper comparison of the Genetic Algorithm, Particle Swarm Optimization and Artificial Neural Network are done with the proposed optimal route algorithm with reinforcement learning (ORAWRF). The iterations are performed on the MATLAB platform. Focus is given on the avoidance of obstacles or road which are blocked or under construction. The various objectives which are taken into consideration while performing simulation are as follows:

- Overall convergence speed of the algorithm;
- Current traffic scenario;
- History weighted function;
- The optimal path probability P;
- The distance from the source to destination using the mentioned four algorithms;
- An algorithm running time.

For performing simulation, various parameters are set as shown in Table 2.

Table 2. Parameters setup

Algorithm Name	Parameter Name	Value
ORAWRL Algorithm	Learning Factor	0.7
	Discount Factor	1
	Greedy Strategy	0.5
	History weight factor	0.6
Genetic Algorithm	Population Space M	100
	Cross Probability P1	0.6
	Mutation Probability P2	0.01
Artificial Neural Network Algorithm	Network Layers	4
	Number of input vertices	8
	Number of hidden layer vertices	2
PSO Algorithm	Particle swarm sample	80
	Learning factor	1.5
	Learning Factor	2.0
	Inertia cost parameter	0.9

RESULTS AND DISCUSSION

All four algorithms are applied for the same source and destination point. In the experimental results, X-axis shows the number of iterations performed whereas the Y-axis represents the shortest route distance. Around 200 iterations are performed for each algorithm during the simulation. The rate of convergence of these algorithms is as shown in graphs 1 to 5.

From the simulation results Figures 1-5, it is clear that at the beginning in every algorithm lots of fluctuation is there but as the number of iterations increases stability comes and at the end, fine rate of convergence is achieved. As seen from graphs, proposed algorithm required fewer iterations' to achieve the target as compared to the Genetic Algorithm and Particle Swarm Optimization. In PSO it takes fewer iterations to achieve the target as compared to the proposed system, but route length provided is much higher. So in all aspects, our proposed system provides better results and stability as compared to other algorithms.

To increase the precision of the results the experimental simulation is extended based on the number of vehicles, route distance and number of turns considering the historical data required to reach from source to destination. Various scales of networks are used for performing the simulation. The same number of nodes and obstacles are kept during simulations for each algorithm. Figures 6 and 7 shows the simulation results.

Figure 1. Rate of convergence for ORAWRL

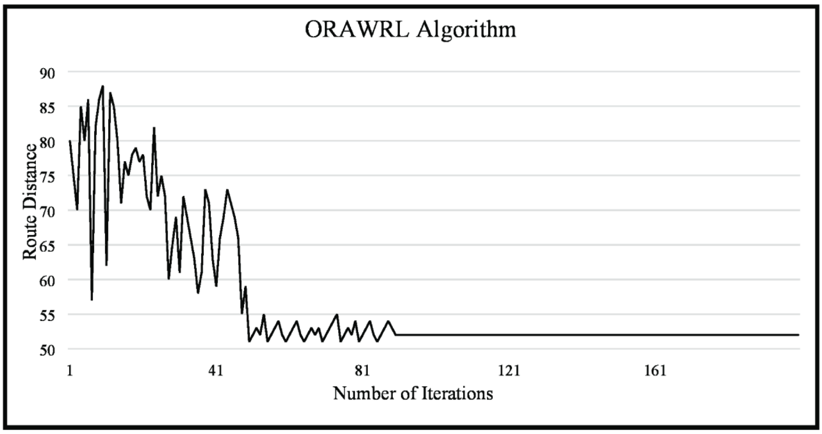


Figure 2. Rate of convergence for genetic algorithm

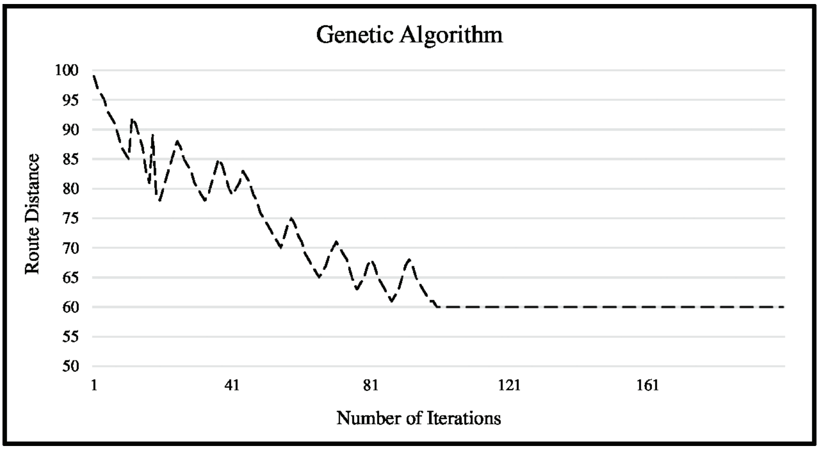


Figure 3. Rate of convergence of artificial neural networks algorithm

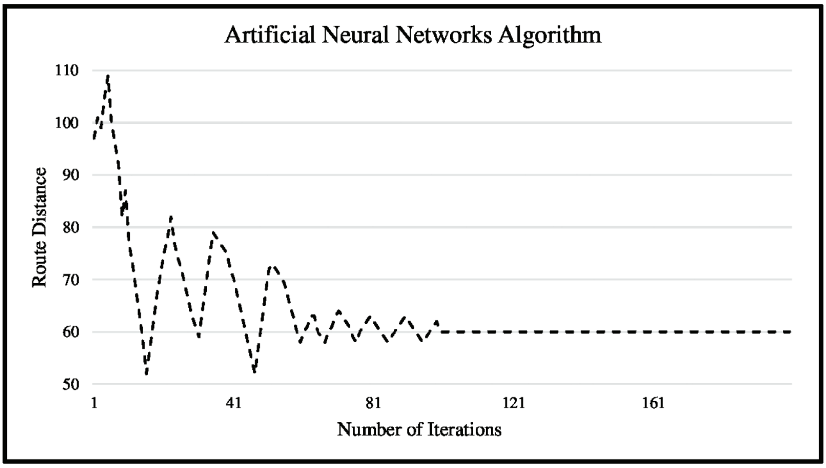


Figure 4. Particle swarm optimization algorithm

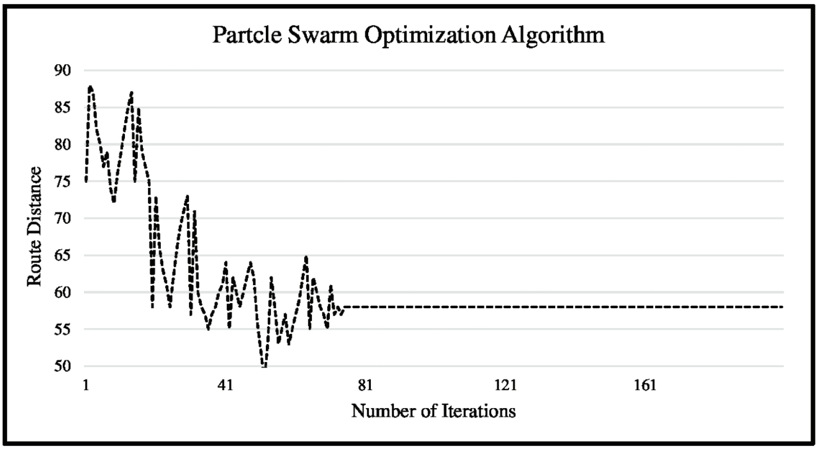


Figure 5. Comparison of all 4 algorithms based on route length

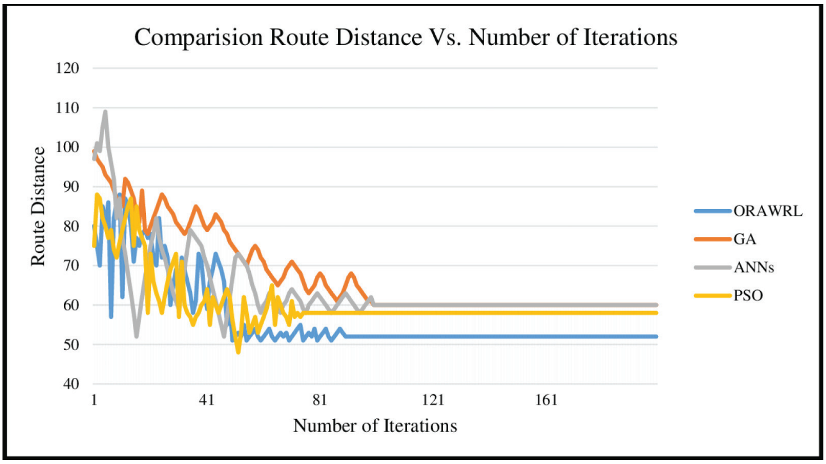
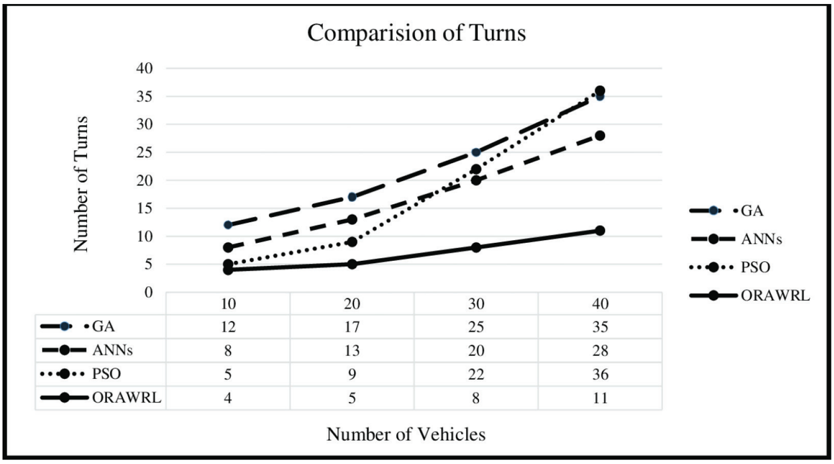


Figure 6 shows a comparison of 4 algorithms on the basic numbers of vehicles vs the number of vehicles at variable scenario sizes. The size of the simulation space is N*N and obstacle probability proportion is still 0.5 and the graph is generated from the data obtained by performing 30 iterations. It is cleared from the graph number of turns taken by proposed graphs in comparatively less. It provides stability and provides an optimal route with minimum weight.

Figure 7 shows the abilities of algorithms to provide an optimal route. Here also 30 sample iterations are performed during the simulation. It cleared from results that when the number of vehicles is less difference between four algorithms is not much. But for the greater number of vehicles, the difference in using the proposed algorithm is making an impact.

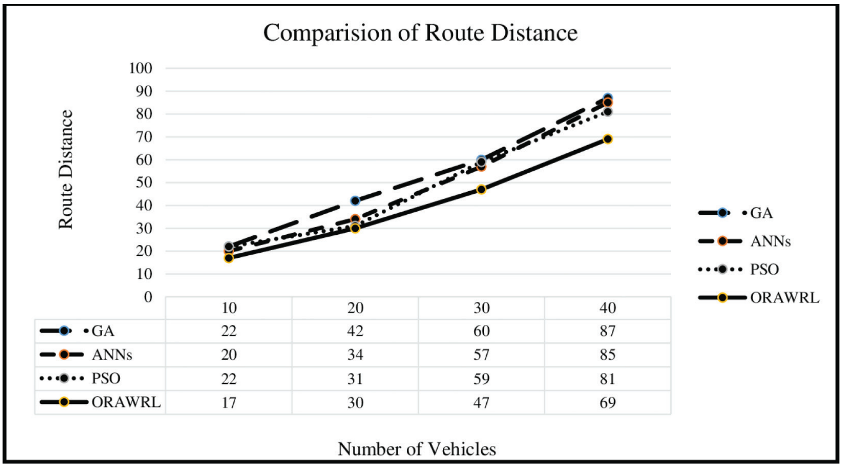
Table 3 shows the simulation results of four algorithms to find out optimal path probability. The proposed algorithm has shown the best results. The best route provided by the proposed algorithm is 47.9 meters and the number of turns required is four as per simulation scenarios. The running time and the optimal path probability of the proposed algorithm is much smaller as compared to the other three algorithms.

Figure 6. Comparison of turns



	10	20	30	40
GA	12	17	25	35
ANNs	8	13	20	28
PSO	5	9	22	36
ORAWRL	4	5	8	11

Figure 7. Comparison of route distance



	10	20	30	40
GA	22	42	60	87
ANNs	20	34	57	85
PSO	22	31	59	81
ORAWRL	17	30	47	69

Table 3. Comparison of simulation results

Algorithms	Planned Route	Best Route	Average Number of Turns	Running Time	Optimal Path Probability
Genetic Algorithm	81.2	47.9	11	29	0.81
Artificial Neural Network	78.6	47.9	8	34	0.93
Particle Swarm Optimization	79.4	47.9	6	29	0.94
ORAWRL	72.1	47.9	4	19	0.46

Though the proposed algorithm is best as compared with the other three algorithms, only one limitation is there that particle swarm optimization technique provides the perfect result with fewer iterations as compared to the proposed algorithm as shown in Figure 5. As the number of iterations increases, the time required to provide the stabilized result is also increased. This limitation will remain as a future scope.

CONCLUSION

A comparative survey analysis of shortest path algorithms is done in this article and based on that a new algorithm called optimal route algorithm with reinforcement learning (ORAWRL) is implemented. This algorithm is helpful for scenarios like jams, accidents, roadblocks, etc. It can analyze dynamically changing traffic scenarios and provide optimal route concerning real-time scenarios. A comparison of the proposed algorithm is done with a genetic algorithm, artificial neural network and particle swarm optimization. From results, it is clear that when the number of vehicles is less, the difference between the four algorithms is not much. But when traffic increases, the difference in using the proposed algorithm is making an impact. The proposed algorithm got better results in terms of route distance, the number of turns, running time, optimal path finding probability. It provides intelligent decision making in terms of cost-efficient route planning. Through simulation results, it is proved that the proposed algorithm has better efficiency, decision making, and stability. It eases the driver's headache and makes traveling comfortable.

REFERENCES

Adler, J. L., Recker, W. W., & McNally, M. G. (1993). A conflict model and interactive simulator (FASTCARS) for predicting enroute driver behavior in response to real-time traffic condition information. *Transportation*, *20*(2), 83–106. doi:10.1007/BF01307054

Amar, H. M., Drawil, N. M., & Basir, O. A. (2016). Traveler Centric Trip Planning: A Behavioral-Driven System. *IEEE Transactions on Intelligent Transportation Systems*, *17*(6), 1521–1537. doi:10.1109/TITS.2015.2499220

Angelelli, E., Arsik, I., Morandi, V., Savelsbergh, M., & Speranza, M. (2016). Proactive route guidance to avoid congestion. *Transp Res B Methodol*, *94*, 1–21. doi:10.1016/j.trb.2016.08.015

Arentze, T. (2013). Adaptive personalized travel information systems: A Bayesian method to learn users' personal preferences in multimodal transport networks. *IEEE Transactions on Intelligent Transportation Systems*, *14*(4), 1957–1966. doi:10.1109/TITS.2013.2270358

Barceló, J., Montero, L., Marqués, L., & Carmona, C. (2010). Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. *Transportation Research Record: Journal of the Transportation Research Board*, *2175*(1), 19–27. doi:10.3141/2175-03

Bast, H., Funke, S., Matijevic, D., Sanders, P., & Schultes, D. (2007). *In transit to constant time shortest-path queries in road networks*. Academic Press.

Ben-Akiva, M., De Palma, A., & Isam, K. (1991). Dynamic network models and driver information systems. Transp. Res. *GEN*, *25*(5), 251–266.

Ben-Elia, E., & Shiftan, Y. (2010). Which road do I take? A learning-based model of route-choice behavior with real-time information. Transp. Res. *Policy & Practice*, *44*(4), 249–264.

Blue, V. J., Adler, J. L., & List, G. F. (1997). Real-time multiple-objective path search for in-vehicle route guidance systems. *Transportation Research Record: Journal of the Transportation Research Board*, *1588*(1), 10–17. doi:10.3141/1588-02

Borkar, P., Malik, L. G., & Sarode, M. V. (2016). Acoustic Signal based Optimal Route Selection Problem: Performance Comparison of Multi-Attribute Decision Making methods. *Transactions on Internet and Information Systems (Seoul)*, *10*(2), 647–669.

Buchhold, V., Sanders, P., & Wagner, D. (2018). Real-time traffic assignment using fast queries in customizable contraction hierarchies. *17th international symposium on experimental algorithms*, 27:1–27:15. doi:10.4230/LIPIcs.SEA.2018.27

Caragiannis, I., Kaklamanis, C., Kanellopoulos, P., Kyropoulou, M., & Papaioannou, E. (2010). The impact of altruism on the efficiency of atomic congestion games. Proc Trustworthly Global Comput. doi:10.1007/978-3-642-15640-3_12

Chan, K. Y., Dillon, T. S., Singh, J., & Chang, E. (2012). Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing using a hybrid exponential smoothing and levenberg-marquardt algorithm. *IEEE Transactions on Intelligent Transportation Systems*, *13*(2), 644–654. doi:10.1109/TITS.2011.2174051

Chen, A., Oh, J. S., Park, D., & Recker, W. (2010). Solving the bicriteria traffic equilibrium problem with variable demand and nonlinear path costs. *Applied Mathematics and Computation*, *217*(7), 3020–3031. doi:10.1016/j.amc.2010.08.035

Chen, P. A., De Keijzer, B., Kempe, D., & Schäfer, G. (2011). The robust price of anarchy of altruistic games. Proc Internet Netw. Econ. doi:10.1007/978-3-642-25510-6_33

Chen, P. A., & Kempe, D. (2008). Altruism, selfishness, and spite in traffic routing. Proc. 9th ACM Conf. Electron. Commerce, 140–149. doi:10.1145/1386790.1386816

Chmiel, W., Skalna, I., & Jędrusik, S. (2019). Intelligent route planning system based on interval computing. *Multimedia Tools and Applications*, *78*(4), 4693–4721. doi:10.1007/s11042-018-6714-x

Delling, D., Goldberg, A. V., Pajor, T., & Werneck, R. F. (2011). Customizable route planning. In P. M. Pardalos & S. Rebennack (Eds.), *Experimental algorithms* (pp. 376–387). Springer. doi:10.1007/978-3-642-20662-7_32

Desai, P., Loke, S. W., Desai, A., & Singh, J. (2013). CARAVAN: Congestion avoidance and route allocation using virtual agent negotiation. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1197–1207. doi:10.1109/TITS.2013.2256420

Desaulniers, G., & Soumis, F. (1995). An efficient algorithm to find a shortest path for a car-like robot. *IEEE Transactions on Robotics and Automation*, *11*(6), 819–828. doi:10.1109/70.478429

Desuó, L., Bessani, M., Fanucchi, R., Gross, T., & Maciel, C. D. (2019). A Multi-objective Swarm Intelligence Approach for Field Crews Patrol Optimization in Power Distribution Systems Restoration. IEEE Latin America Transactions, 17(2), 338-346. doi:10.1109/TLA.2019.8863181

DeviPriya, K., & Lingamgunta, S. (2020). Multi Factor Two-way Hash-Based Authentication in Cloud Computing. *International Journal of Cloud Applications and Computing*, *10*(2).

Dib, O., Manier, M. A., Moalic, L., & Caminada, A. (2017). Combining VNS with genetic algorithm to solve the one-to-one routing issue in road networks. *Computers & Operations Research*, *78*, 420–430. doi:10.1016/j.cor.2015.11.010

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. doi:10.1007/BF01386390

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. doi:10.1007/BF01386390

Dinitz, Y., & Itzhak, R. (2017). Hybrid Bellman–Ford–Dijkstra algorithm. *J Discrete Algo*, *42*, 35–44. doi:10.1016/j.jda.2017.01.001

Ehmke, J., Campbell, A., & Thomas, B. (2016b). Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research*, *251*(2), 478–494. doi:10.1016/j.ejor.2015.11.034

Ehrgott, M., Wang, J. Y., Raith, A., & Houtte, C. (2012). A bi-objective cyclist route choice model. Transp. Res. *Policy & Practice*, *46*(4), 652–663.

Feng, G., & Mingzhe, W. (2010). Route choice behavior model with guidance information. *Journal of Transportation Systems Engineering and Information Technology*, *10*(6), 64–69. doi:10.1016/S1570-6672(09)60072-6

Franceschetti, A., Honhon, D., Van Woensel, T., Bektaş, T., & Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, *56*, 265–293. doi:10.1016/j.trb.2013.08.008

Gentili, M., & Mirchandani, P. B. (2012). Locating sensors on traffic networks: Models, challenges and research opportunities. *Transportation Research Part C, Emerging Technologies*, *24*, 227–255. doi:10.1016/j.trc.2012.01.004

Ghiduk, A. S. (2014). Automatic generation of basis test paths using variable length genetic algorithm. *Information Processing Letters*, *114*(6), 304–316. doi:10.1016/j.ipl.2014.01.009

Glaeser, E. L. (2020). Urbanization and Its Discontents. *Eastern Economic Journal*, *46*(2), 191–218. doi:10.1057/s41302-020-00167-3

Goldberg, A. V., & Harrelson, C. (2004). *Computing the shortest path: A* search meets graph theory.* Tech. Rep. MSR-TR-2004-24 Microsoft Research, Vancouver, Canada.

Guidoni, D. L., Maia, G., Souza, F. S. H., Villas, L. A., & Loureiro, A. A. F. (2020). Vehicular Traffic Management Based on Traffic Engineering for Vehicular Ad Hoc Networks. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 45167–45183. doi:10.1109/ACCESS.2020.2978700

Gupta, B. B. (2019). *Modern Principles, Practices, and Algorithms for Cloud Security*. IGI Global.

Gutman, R. (2004). Reach-based routing: a new approach to shortest path algorithms optimized for road networks. *Proceedings 6th workshop on algorithm engineering and experiments (ALENEX)*, *SIAM*, 100–111.

Hellinga, B., & Fu, L. (1999). Route selection considering travel time variability. *Proc. 6th World Congr. Intell. Transp. Syst.*, 1–8.

Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., & Bayen, A. M. (2010). Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C, Emerging Technologies*, *18*(4), 568–583. doi:10.1016/j.trc.2009.10.006

Hoefer, M., & Skopalik, A. (2009). Altruism in atomic congestion games. Proc Algorithms-ESA.

Holzer, M., Schulz, F., & Wagner, D. (2009). Engineering multilevel overlay graphs for shortest-path queries. *J Exp Algorithmics, 13*, 5:2.5–5:2.26.

Hu, Sh. R., Peeta, S., & Chun-Hsiao, C. (2009). Identification of vehicle sensor locations for link-based network traffic applications. *Transportation Research Part B: Methodological*, *43*(8-9), 873–894. doi:10.1016/j.trb.2009.02.008

Hu, X. B., Wang, M., Leeson, M. S., Di Paolo, E. A., & Liu, H. (2016). Deterministic agent-based path optimization by mimicking the spreading of ripples. *Evol Comput, 24*(2), 319–346. a00156<ALIGNMENT.qj></ALIGNMENT>10.1162/EVCO

Hu, X. B., Zhang, M. K., Zhang, Q., & Liao, J. Q. (2017). Co-evolutionary path optimization by Ripple-Spreading algorithm. *Transportation Research Part B: Methodological*, *106*, 411–432. doi:10.1016/j.trb.2017.06.007

Huang, W., Yan, C., Wang, J., & Wang, W. (2017). A time-delay neural network for solving time-dependent shortest path problem. *Neural Networks*, *90*, 21–28. doi:10.1016/j.neunet.2017.03.002 PMID:28364676

Huang, Y., Zhao, L., Woensel, T. V., & Gross, J. P. (2016, Oct.). Time-Dependent Vehicle Routing Problem with Path Flexibility. *Transportation Research Part B: Methodological*.

Idri, A., Oukarfi, M., Boulmakoul, A., Zeitouni, K., & Masri, A. (2017). A new time-dependent shortest path algorithm for multimodal transportation network. *Procedia Computer Science*, *109*, 692–697. doi:10.1016/j.procs.2017.05.379

Illenberger, J., Flötteröd, G., & Nagel, K. (2011). A model of risk-sensitive route-choice behavior and the potential benefit of route guidance. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 384–389. doi:10.1109/TITS.2011.2105266

Jiang, F. (2018). Deep Learning based Multi-channel intelligent attack detection for Data Security. IEEE Transactions on Sustainable Computing, 1-1. doi:10.1109/TSUSC.2018.2793284

Khekare, G. S. (2014). Design of emergency system for intelligent traffic system using VANET. *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 1-7. doi:10.1109/ICICES.2014.7033910

Khekare, G. S., & Sakhare, A. V. (2013). A smart city framework for intelligent traffic system using VANET. International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 302-305. doi:10.1109/iMac4s.2013.6526427

Khekare, S., & Janardhan, S. (2017). Stability Analysis of a Vector Host Epidemic Model. *Asian Journal of Mathematics & Computer Research*, *21*(3), 98–109.

Kim, J. W., & Yang, J. H. (2020). Understanding Metrics of Vehicle Control Take-Over Requests in Simulated Automated Vehicles. *International Journal of Automotive Technology*, *21*(3), 757–770. doi:10.1007/s12239-020-0074-z

Kim, S., Lewis, M. E., & White, C. C. (2005). Optimal vehicle routing with real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems*, *6*(2), 178–188. doi:10.1109/TITS.2005.848362

Kubek, D., Wiecek, P., & Chwastek, K. (2016). The impact of short term traffic forecasting on the effectiveness of vehicles routes planning in urban areas. *Transportation Research Procedia*, *18*, 172–179. doi:10.1016/j.trpro.2016.12.024

Lauther, U. (2004). An extremely fast, exact algorithm for finding shortest paths in static networks with geographical background. Geoinformation und mobilit¨at – von der forschung zur praktischen anwendung, 22, 219–230.

Lee, S. B., Lee, S. M., & Lee, K. Y. (2017). A Gini coefficient based evaluation on the reliability of travel time forecasting. *Journal of King Saud University - Engineering and Science*. Advance online publication. doi:10.1016/J.JKSUES.2017.12.001

Marinakis, Y., Migdalas, A., & Sifaleras, A. (2017). A hybrid particle swarm optimization – variable neighborhood search algorithm for constrained shortest path problems. *European Journal of Operational Research*, *261*(3), 819–834. doi:10.1016/j.ejor.2017.03.031

Maue, J., Sanders, P., & Matijevic, D. (2010). Goal–directed shortest-path queries using precomputed cluster distances. *J Exp Algorithmics, 14*, 2:3.2–2:3.27.

Miller, J. (2009). Dynamically computing fastest paths for intelligent transportation systems. IEEE Intelligent Transportation Systems Magazine, 20-26. doi:10.1109/MITS.2009.932717

Naghawi, H., & Wolshon, B. (2010). Transit-based emergency evacuation simulation modeling. *Journal of Transportation Safety & Security*, *2*(2), 184–201. doi:10.1080/19439962.2010.488316

Pal, D., Shah, D., & Bhagat, S. K. (2019). Design of a Genetic Algorithm Based Shortest Path Routing with different Mutation Techniques. *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 1-5. doi:10.1109/ICECCT.2019.8869243

Patle, B., Pandey, A., Jagadeesh, A., & Parhi, D. (2018). *Path planning in uncertain environment by using firefly algorithm*. Defence Technology. doi:10.1016/j.dt.2018.06.004

Paumard, M., Picard, D., & Tabia, H. (2020). Deepzzle: Solving Visual Jigsaw Puzzles With Deep Learning and Shortest Path Optimization. *IEEE Transactions on Image Processing*, *29*, 3569–3581. doi:10.1109/TIP.2019.2963378 PMID:31944956

Premkamal, P. K., Pasupuleti, S. K., & Alphonse, P. J. A. (2020). Efficient Escrow-free CP-ABE with Constant Size Ciphertext and Secret Key for Big Data Storage in Cloud. *International Journal of Cloud Applications and Computing*, *10*(1), 28–45. doi:10.4018/IJCAC.2020010103

Qian, J., & Eglese, R. (2016). Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research*, *248*(3), 840–848. doi:10.1016/j.ejor.2015.09.009

Rahmani, M., Koutsopoulos, H. N., & Jenelius, E. (2017). Travel time estimation from sparse floating car data with consistent path inference: A fixed point approach. *Transportation Research Part C, Emerging Technologies*, *85*, 628–643. doi:10.1016/j.trc.2017.10.012

Raith, A., Wang, J. Y., Ehrgott, M., & Mitchell, S. A. (2014). Solving multi-objective traffic assignment. *Annals of Operations Research*, *222*(1), 483–516. doi:10.1007/s10479-012-1284-1

Ren, Z. Y. (2017). A summary of research on autonomous obstacle path planning for intelligent vehicles. *Softw. Guide*, *16*(10), 209–212.

Sanders, P., & Schultes, D. (2005). Highway hierarchies hasten exact shortest path queries. In *Proceedings of the 13th annual European conference on algorithms, ESA'05*. Springer-Verlag. doi:10.1007/11561071_51

Sanders, P. D. S. (2006). Engineering highway hierarchies. In *Proceedings of the 14th conference on annual European symposium, ESA'06*. Springer-Verlag.

Sarbazi-Azad, H., Chapuis, G., Djidjev, H., Lavenier, D., & Andonov, R. (2017). GPU-accelerated shortest paths computations for planar graphs. *Advances in GPU Research and Practice*, 273–306. <ALIGNMENT.qj></ALIGNMENT>10.1016/B978-0-12-803738-6.00011-2

Schultes, D., & Sanders, P. (2007). Dynamic highway-node routing. In C. Demetrescu (Ed.), *Experimental algorithms, lecture notes in computer science* (Vol. 4525, pp. 66–79). Springer.

Schulz, F., Wagner, D., & Weihe, K. (1999). Dijkstra's algorithm on-line: an empirical case study from public railroad transport. In *Proceedings of the 3rd international workshop on algorithm engineering, WAE'99*. Springer-Verlag. doi:10.1007/3-540-48318-7_11

Sever, D., Zhao, L., Dellaert, N., Demir, E., Van, W. T., & De, K. T. (2018). The dynamic shortest path problem with time-dependent stochastic disruptions. *Transportation Research Part C, Emerging Technologies*, *92*, 42–57. doi:10.1016/j.trc.2018.04.018

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265. doi:10.1287/opre.35.2.254

Treiber, M., & Kesting, A. (2012). Validation of traffic flow models with respect to the spatiotemporal evolution of congested traffic patterns. *Transportation Research Part C, Emerging Technologies*, *21*(1), 31–41. doi:10.1016/j.trc.2011.09.002

Vlahogianni, E. I., & Karlaftis, M. G. (2013). Testing and comparing neural network and statistical approaches for predicting transportation time series. *Transportation Research Record: Journal of the Transportation Research Board*, *2399*(1), 2399. doi:10.3141/2399-02

Wagner, D., & Willhalm, T. (2003). Geometric speed-up techniques for finding shortest paths in large sparse graphs. *Proceedings of the 13th annual European conference on algorithms, ESA'03*, 776–787. doi:10.1007/978-3-540-39658-1_69

Wahle, J., Bazzan, A. L. C., Klugl, F., & Schreckenberg, M. (2000). Decision dynamics in a traffic scenario. *Physica A*, *287*(3/4), 669–681. doi:10.1016/S0378-4371(00)00510-0

Wahle, J., Bazzan, A. L. C., Klügl, F., & Schreckenberg, M. (2002). The impact of real-time information in a two-route scenario using agent-based simulation. *Transp. Res. C. Emerging Technol.*, *10*(5/6), 399–417. doi:10.1016/S0968-090X(02)00031-1

Yang, C. H., Tsai, M. H., Kang, S. C., & Hung C. Y. (2018). UAV path planning method for digital terrain model reconstruction – A debris fan example. *Autom Construc, 93*, 214–230. .AUTCON.2018.05.02410.1016/J

Yang, J., & Jiang, G. (2014). Development of an enhanced route choice model based on cumulative prospect theory. Transp. Res. C. *Emerging Technol.*, *47*(2), 168–178. doi:10.1016/j.trc.2014.06.009

Zeng, X., & Wang, J. (2018). Globally energy-optimal speed planning for road vehicles on a given route. *Transportation Research Part C, Emerging Technologies*, *93*, 148–160. doi:10.1016/j.trc.2018.05.027

Zhang, D. G., Chen, C., Cui, Y. Y., & Zhang, T. (2019). New method of energy efficient subcarrier allocation based on evolutionary game theory. *Mobile Networks and Applications*, *24*(10), 1930–1945.

Zhang, D. G., Cui, Y.-Y., Gao, J.-X., Liu, X.-H., Zhang, T., & Tang, Y.-M. (2019). Novel reliable routing method for engineering of Internet of vehicles based on graph theory. *IEEE Access: Practical Innovations, Open Solutions*, *36*(1), 226–247.

Zhang, D. G., Liu, S., Zhang, T., Cui, Y. Y., & Liu, X. H. (2018). Novel dynamic source routing protocol (DSR) based on genetic algorithm-bacterial foraging optimization (GA-BFO). *International Journal of Communication Systems*, *31*(18), 1–20. doi:10.1002/dac.3824

Zhang, D. G., Zhang, T., Cui, Y. Y., Liu, X., Mao, G., & Ge, H. (2019). New multihop clustering algorithm for vehicular Ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, *20*(4), 1517–1530. doi:10.1109/TITS.2018.2853165

Zhang, D. G., Zhang, T., Dong, Y., Liu, X., Cui, Y., & Zhao, D. (2018). Novel optimized link state routing protocol based on quantum genetic strategy for mobile learning. *Journal of Network and Computer Applications*, *122*, 37–49. doi:10.1016/j.jnca.2018.07.018

Zhang, D. G., Zhang, T., Zhao, D. X., Gao, J. X., & Liu, X. H. (2019). Novel approach of distributed & adaptive trust metrics for MANET. *Wireless Networks*, *25*(6), 3587–3603. doi:10.1007/s11276-019-01955-2

Zhang, Y., Zhang, Y., & Haghani, A. (2014). A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. *Transportation Research Part C, Emerging Technologies*, *43*, 554–573. doi:10.1016/j.trc.2013.11.011

Zhao, Y., Zheng, Z., & Liu, Y. (2018). Survey on computational-intelligence-based UAV path planning. *Knowledge-Based Systems*, *158*, 54–64. doi:10.1016/j.knosys.2018.05.033

International Journal of Software Science and Computational Intelligence
Volume 12 • Issue 4 • October-December 2020

Improving Road Safety for Driver Malaise and Sleepiness Behind the Wheel Using Vehicular Cloud Computing and Body Area Networks

Meriem Benadda, University Oran 1, Algeria
Ghalem Belalem, University Oran 1, Algeria
https://orcid.org/0000-0002-9694-7586

ABSTRACT

Malaise and sleepiness behind the wheel are considered to be the leading causes of fatal highway accidents. With the body area networks (BANs), a continuous health monitoring of a driver can be performed without any constraint on his/her normal daily life activities. Many of the systems proposed in the literature are intended to prevent traffic accidents but without treating this kind of cause because difficult to highlight in an accident. This paper proposes "HAaaS," a new vehicular cloud computing service based on BANs to detect, monitor, and manage driver malaise and provide a cooperation support for the driver rescue. The objective is to reduce the number of accidents, the material and human damage as the time and fuel lost in traffic jams. The proposed service has been validated by simulating real-world highway scenarios extracted from Oran city in Algeria. The results show that the service is efficient at a significant rate.

KEYWORDS

Body Area Networks, Cooperation, Health Monitoring, Intelligent Transportation Systems, Malaise Behind the Wheel, VANets, Vehicular Cloud Computing, World Health Organization

INTRODUCTION

Since their appearance, VANets (Vehicular Area Networks) caught the attention of research communities, major automakers and governments because of their potential applications and their specific characteristics. The VANets research results began with vehicle awareness of collision avoidance to Internet access and then expanded to vehicular multimedia communications. In addition, the high computing, communication, and storage resources of the vehicle are a fertile ground for deploying these applications in the near future. Nevertheless, the resources on board vehicles are for the most part underutilized. The introduction of the new hybrid technology known as Vehicular Cloud Computing (VCC) (Whaiduzzaman, Sookhak, Gani, & Buyya, 2014) has had a great impact

DOI: 10.4018/IJSSCI.2020100102

on Intelligent Transportation Systems (ITS) by utilizing underused vehicle resources such as GPS (Global Positioning System), storage, Internet, and computing power to make instant decisions and share information on the Cloud. At the same time, the proliferation of the automotive industry has made it possible to design more and more equipped vehicles even for everyday use vehicles. These latter include relatively more communication systems such as embedded computing devices, storage and computing power, GPS, etc. to provide an ITS. Thereby, by combining VCC technology and these new vehicles, we can obtain a state-of-the-art system that can solve most of the road safety and driver well-being problems.

Motivator Elements

Traffic accidents are considered to be one of the leading causes of death in the world. Every year more than 1.25 million people die in road traffic accidents and 20 to 50 million more are injured, sometimes even become disabled as a result of their injuries, according to the World Health Organization (World Health Organization [WHO], 2020). These accidents result in considerable economic losses for those who are victims, their families and the countries as a whole. In 2017, and in the United States only, there was a loss of $ 166 billion (160 in 2014) with 8.8 billion lost hours(6.9 in 2014) and lost 3.3 billion gallons (3.1 in 2014) of wasted gasoline due to traffic congestion (Schrank, Eisele, Lomax, & Bak, 2019). More seriously, these numbers will increase by about 65% over the next 20 years if there is no new commitment to prevention according to projections made within the World Health Organization (WHO, 2020).

Context and Problematics

Before beginning, should be noted that a malaise can be materialized by a crisis, syncope or a falling asleep and according to the French highway companies association ASFA (Association des Sociétés Françaises d'Autoroutes [ASFA], 2018), the *drowsiness-sleepiness* factor was considered, between 2014 - 2018, as the number one cause of fatal accidents on a highway with 27% of the total number of fatalities (ASFA, 2018). These accidents occurred in 52% of cases during the day, mainly in the early morning (between 4am and 6am), as well as in the afternoon (between 3pm and 4pm), as indicated in the latest safety report of the association. Also, according to the French road prevention association (Association Prévention Routière [APR], 2016), the *malaise drowsiness* factor appears in 9% of deaths on the entire road network. Finally, a study conducted by Professor Pierre Phillip and ASFA (ASFA, 2018) on a sample of French drivers, in 2017, showed that 47% of the drivers questioned suffered sleepiness during the night and 28% had at least one episode of severe sleepiness at the wheel with an obligation to stop and that 11% had at least one almost accident, with exit of road or uncontrolled crossing of line.

Considering the overall statistics concerning malaise and sleepiness behind the wheel, and knowing that it is the number one cause of fatal accidents and that it is difficult to highlight it in an accident and that there are no works, until now, which interest on giving a technological solution to deal with this cause of accidents, that this paper proposes a service called HAaaS "Health-Assistance as a Service" to detect, monitor and manage a driver's malaise and sleepiness. This service strives to make a positive difference in the field of road safety. By monitoring the data of the vehicle and the driver in real time through the linkage of various sensors in the vehicle and around the driver (Vehicular sensors and BANs), it could effectively allow people with chronic disease to take vehicle safely, curb traffic accidents, restrain material and human damages, reduce the overall costs caused by congestion (hospital costs, fuel, lost hours,..), give a fast medical help and improve the operation safety of vehicles.

For the remainder of the paper, this latter is organized as follows: Section two provides a state of the art of literature in this area and an overview of the proposed systems in the field of vehicular Cloud Computing. Section three details the proposed service for the management of driver's malaise.

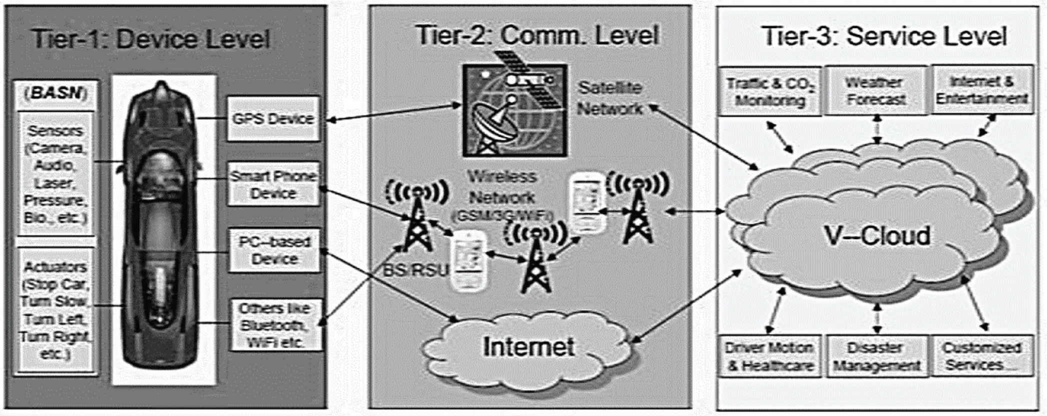
Section four presents two scenarios of the happening of a malaise, one using the proposed service HAaaS and the other without its use. Finally, section five discusses and concludes this paper.

BACKGROUND

In the last few years, many works have been consecrated to providing services in a Vehicular Cloud environment. At the same time, a lot of works have been done in the field of prevention and road safety. However, many few of them have been dedicated to providing services in the field of health, and especially in health monitoring and well-being of the driver in a VCC environment. Indeed, the only work founded that combines health monitoring, BANs, and VCC is (Wang, Cho, Lee, & Ma, 2011). However, their work has no implementation or algorithms; it remains still at the proposal stage.

In (Wang et al., 2011), the authors propose their new three-tier Vehicular Cloud architecture called V-Cloud (Figure 1) which includes the device level, the communication level, and the service level. The authors introduce the concept of Cloud Computing enabled by real-time car services with special emphasis on personalized services. They use various devices from sensors, actuators, GPS, Smartphones and PC-based devices (Personal Computer based devices). Also, they employ the Body Area Sensor Network technology in order to monitor the driver's health physical information.

Figure 1. Three tier V-Cloud architecture (Wang et al., 2011)

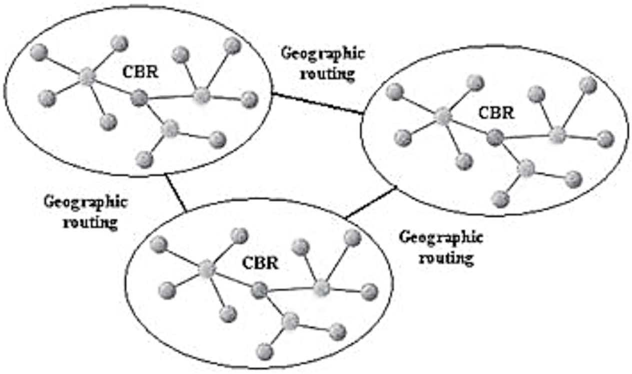


The authors mention the huge amount of financial and human costs caused by traffic congestion. They estimate that each vehicle must cooperate with other vehicles or RSI (Road Side Infrastructure) in a vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) manner to share and use external resources more efficiently. In fact, the concept of Cloud Computing is presented from an economic perspective and the key idea is to rent software and platform services rather than buying them. Also, they present some personalized and innovative services in real time for future vehicular Cloud applications such as driver health monitoring and car sharing resources services, etc. to facilitate the interaction between drivers and the outside world of the automobile.

Always in the Vehicular Cloud environment, but this time the emphasis is not put on health. The authors in (Abuelela & Olariu, 2010) describe how underutilized computing equipment, networks, and storage can be used efficiently to be shared between drivers or rent to others clients on the internet by setting up what they call a Vehicular Cloud Computing or VC2. The improvement of three existing services in VANet is proposed. These services are Computing as a Service, Network as a Service and Storage as a Service. For the first service, they discuss several instances of a VC2 that aggregates the calculation capabilities of parked cars. For the second service, they propose that every driver with internet connectivity, who is willing to share this resource, will announce this information to all the vehicles around him on the road. For the third service, the authors propose to use the several terabytes of storage which are attached to computers on cars in many Cloud applications.

Also in VCC, but this time it is the cooperation aspect that is highlighted, an extension of NaaS (Network as a Service) and SaaS (Storage as a Service) is proposed (Mousannif, Khalil, & Al Moatassime, 2011). The authors introduce a new service called *Cooperation as a Service* or CaaS (Figure 2) in which a hybrid publish/subscribe mechanism is used. The objective of authors is to provide sets of free services and without any additional infrastructure for vehicles/drivers that are willing to co-operate enjoying V2V communications.

Figure 2. The proposed network structure (Mousannif, Khalil, & Al Moatassime, 2011)



In (Bitam & Mellouk, 2012), a new model of Cloud Computing called ITS-Cloud applied to intelligent transportation system is proposed in order to improve road safety, transport productivity, travel reliability, informed travel choices, environmental protection, and resilience of traffic. They present a three layers architecture (Figure 3) which includes end-users, communication, and Cloud layers. To validate their approach, the authors perform a simulation study using the Bees Life Algorithm (BLA) in order to deal with load balancing as an NP-complete problem.

In road safety, the author in (Bektache, 2014) describes the design, the modeling and the simulation of a prediction approach called FCAA (Forecasting Collision Avoidance Approach), which is based on the detection, prediction, and avoidance of the collision (Figure 4). He proposes a new communication architecture based on the kinematic modeling of vehicle type and the filter method of Kalman for the trajectories estimation. The proposed approach has been validated by simulating real-world intersections scenarios extracted from the Annaba city in Algeria. The results show that the FCAA approach is enabled at a significant rate.

In (Sam, Evangelin, & Cyril Raj, 2015), the authors propose a Hybrid VANet based driver alert system. An alert given to the drivers, ahead of time, gives a better chance for the driver's act in a way as to avoid accidents. In their system, the authors integrate a pedestrian body unit in a VANet (Figure 5), this unit sends signals to the vehicular nodes in the VANet, and these signals are given as input to the alert system which will notify the drivers about surrounding pedestrians, and in turn, give him more reaction time. The system was simulated and a laboratory demonstration was also done. It all proved that the system was able to reduce the chances of accident drastically with the alert system.

In (Alshareef & Grigoras, 2018) the authors propose a mobile cloud service in the medical emergency field which can be considered as a form of help that can be complementary to the existing emergency systems. The service aims to provide a directory of trusted and qualified medical

Figure 3. ITS-Cloud architecture (Bitam & Mellouk, 2012)

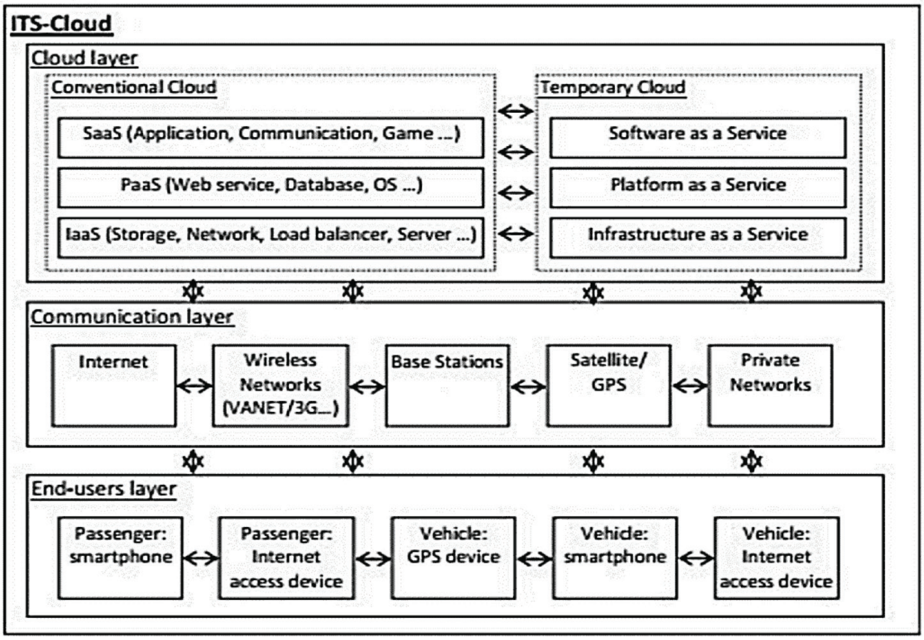
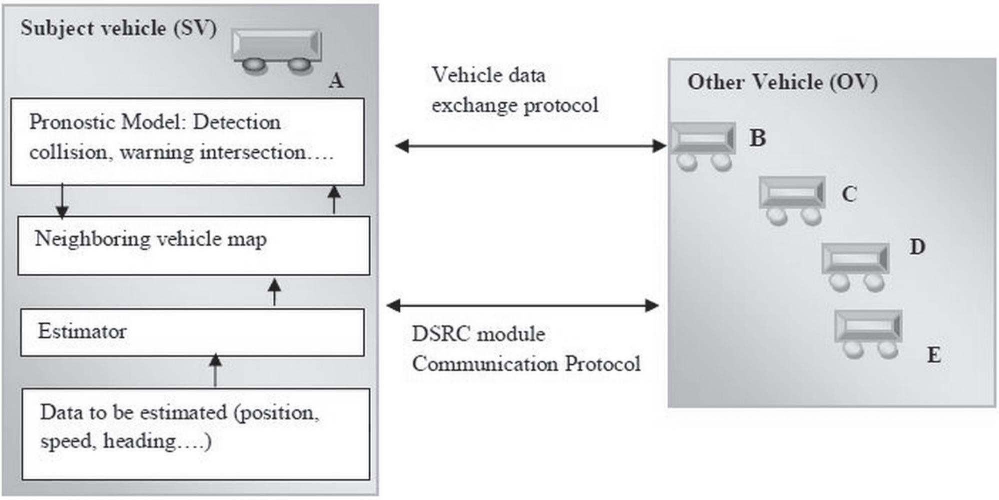
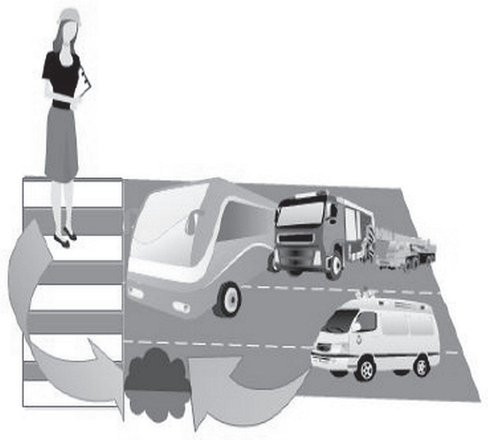


Figure 4. Architecture of the proposed communication model (Bektache, 2014)



professionals to a person in a medical emergency case and requesting help. The only point in common with the work proposed in this paper is that the persons who can give help are in close proximity to person requesting help and that the objective to give a fast healthcare service is one of the several objectives defined in this paper. However, their service (Alshareef & Grigoras, 2018) concerns medical professionals and the context is different. Indeed, in this paper the proposed service detects the malaise and sends, automatically, a helping message unlike to their service in which the medical emergency case is not detected and the help message must be sent by a person (the person requesting

Figure 5. The proposed approach model (Sam, Evangelin, & Cyril Raj, 2015)



help or someone around him/her). Adding to that, their system is not used for driver's malaise and traffic accidents prevention and does not use wireless sensors.

In (Abatal, Khallouki, & Bahaj, 2018) the authors present a smart interconnected healthcare system based on cloud. Their proposed system can connect and share health information to authorized health professionals where and when needed. The data that can be collected and analyzed in real-time helps the health professionals shift from treating patients on visit at a time to proactively focusing on prevention and wellness. Their architecture is composed of user interfaces, medical database representing the medical data, the medical records which are stored in the cloud and web services which allows the communication with different parts of their system. Although, their proposition allows making the community healthier, their system does not allow the detection and the management of a malaise while driving.

In (Shen et al., 2018) the authors present a smart safety monitoring system for vehicles, in order to satisfy drivers' driving, and health and safety needs. Their system collects and analyses driver's behavior inter-dimensionally through multiple sensors placed on the car and the smart band which the driver wears. It detects drunk and fatigue driving, heart rate, blood pressure, body temperature, etc. The experiments show that their system works fairly well in preventing driving hazards and in monitoring the driver's health. While providing drivers with fatigue and drunk driving reminders and alarms, their system can also monitor drivers' physical indicators and offers them easy access via the voice assistant. However, it cannot manage the driver and the vehicle and does not offer a cooperation mechanism to assist and rescue the driver. In addition, there were no details about the development environment.

In order to situate the proposed work regarding those cited above, a comparative study has been performed by the authors. Table 1 summarizes this study.

HEALTH-ASSISTANCE AS A SERVICE

In this section, the architecture and the process of the proposed service are presented.

Architecture

In order to find a solution to the problems of malaise and sleepiness behind the wheel, the possibility of integrating the Body Area Networks technology (Wikipedia, 2019) to the Vehicular Cloud Computing environment is proposed. The objective is to show how it is possible to save lives by managing persons

Table 1. Comparative study regarding related works

Research Work	The Research Objective	Development Environment	BANs Technology	Road Safety	Health Monitoring	Simulation	Metric
Wang et al., 2011	Provide personalized services	VCC	Used	Not processed	Processed	None	The financial and human costs
Abuelela & Olariu, 2010	Share and rent underutilized resources	VCC	Not used	Partially processed	Not processed	None	The financial cost
Mousannif, Khalil, & Al Moatassime, 2011	Share underutilized resources	VSN (Vehicular Sensor Network) and VCC	Not used	Not processed	Not processed	None	The financial cost
Bitam & Mellouk, 2012	Share underutilized resources	ITS and Cloud Computing	Not used	Processed	Not processed	Done	The execution time (makespan)
Bektache, 2014	Detect, predict and avoid collisions	VANET	Not used	Processed	Not processed.	Done	The number of collisions
Sam, Evangelin, & Cyril Raj, 2015	Avoid accidents between vehicles and pedestrians	VANET	Not used	Processed	Not processed	Done	The number of traffic accidents
Alshareef & Grigoras, 2018	Provide a fast and qualified medical help to a person in a medical emergency case	Cloud Computing	Not used	Not processed	Not processed	Done	The response time
Abatal, Khallouki, & Bahaj, 2018	Enable the visualization, transmission and sharing of medical records and reports between patients and their doctors	Cloud Computing and Web service	Not used	Not processed	Not processed	Done	The accessibility to the medical data
Shen et al., 2018	Detect drunk and fatigue driving	No details were mentioned	Used	Processed	Processed	Done	The capacity of facial detection
The proposed work	Monitor driver's health records, detect malaise and sleepiness behind the wheel, manage driver and vehicle and provide a fast medical help to the driver	VCC	Used	Processed	Processed	Done	The number of traffic accidents The number of lost hours, the wasted gasoline and the total financial cost

taken malaise or a falling asleep behind the wheel. Also, show how to handle the vehicles in case of loss of control and finally, how to rescue the driver taken malaise using a cooperation mechanism. To do this a new architecture of a Vehicular Cloud Computing service called HAaaS "Health-Assistance as a Service" (Figure 6) is proposed in this paper.

The proposed service is a three-layer architecture (Figure 6) which includes: the device layer, the communication layer and the Cloud layer (Benadda, Bouamrane, & Belalem, 2017).

Device Layer

The device layer represents the first layer of the proposed service which is responsible for collecting information within the vehicle in order to predict the driver's reflexes and intentions. It is divided into two subsystems (Figure 7); the first one includes the vehicles of sick people (or healthy people wanting their health to be monitored or who regularly take long trips and who are afraid of falling asleep behind the wheel) so they will be equipped, in addition to the sensors and actuators of the vehicle (Benadda, Bouamrane, & Belalem, 2017), with BAN sensors such as blood glucose or blood pressure sensors, etc.); the second subsystem includes vehicles that can provide assistance and rescue

Figure 6. The proposed service architecture

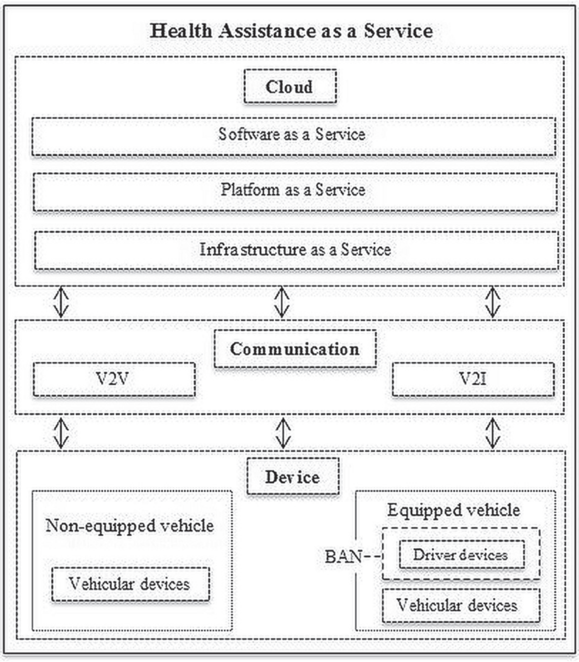
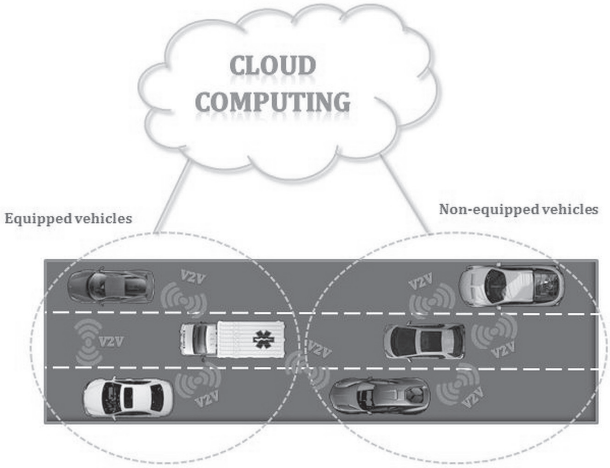


Figure 7. Device layer subsystems



driver who take malaise. In this second subsystem no BAN sensors are present in the vehicles but only automotive ones (Benadda, Bouamrane, & Belalem, 2017).

Communication Layer

The communication layer is the second layer of the proposed architecture. It is responsible for the information exchange between vehicles and the Cloud (V2I) and between the vehicles themselves (V2V) (Benadda, Bouamrane, & Belalem, 2017).

Cloud Layer

The Cloud layer is the last layer of the proposed service. It includes the three basic Cloud services: Infrastructure as a Service, Platform as a Service and Software as a service.

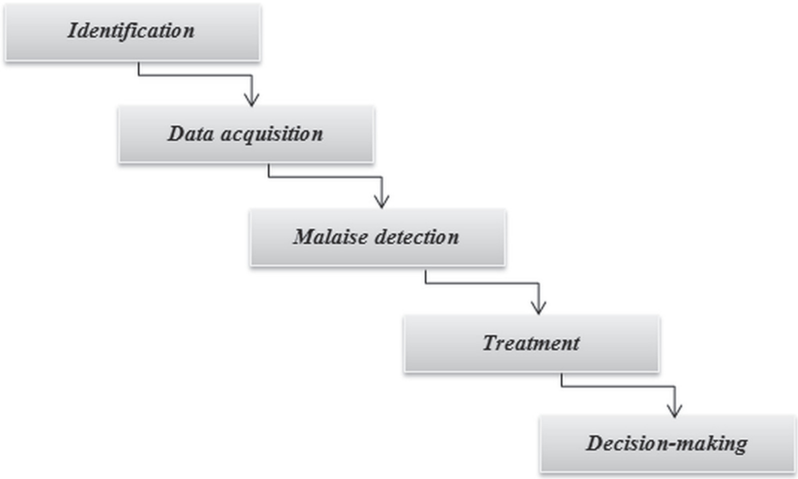
Process

The HAaaS process passes through two important and complementary phases: the first is the monitoring and recognition (exploration) phase and the second is the cooperation and management phase.

Phase One: The Monitoring and Recognition (Exploration) Phase

This first phase comprises the following five steps: Identification, Data acquisition, Malaise detection, treatment and Decision making (Figure 8).

Figure 8. The stages of the monitoring and recognition phase



Identification

The HAaaS service process begins with the identification of the driver. In this step, the most important task is the filling of a form (Figure 9). This latter contains personal information about the driver such as date of birth, sex, blood group, chronic diseases if any, etc. This operation is done by following some conditions (Benadda, Bouamrane, & Belalem, 2017). In return, the Cloud provides a unique alphanumeric identifier, composed by 19 characters, which will be used in case of vehicle changing. However, if the same driver takes the same vehicle, he will be connected directly after starting the car via his On Board Unit (OBU).

Data Acquisition

In this second step, each of the vehicle devices or the driver sensors starts collecting and transmitting data to the Cloud.

Malaise Detection

In this third step, different situations indicating that a malaise or a falling asleep (sleepiness) occurred are explained (Benadda, Bouamrane, & Belalem, 2017). Among them, an abnormal measurement

Figure 9. Example of a completed form

Birth date	09.05.1955
Gender	**F** ☐ **M** ▣
Blood group	A^+
First aid	▣
Parent number	0660.60.09.99
Relationship	Daughter
Drug allergies	▣
Drug name	Penicillin
Sick person	▣
Chronic diseases	Hypertension
Current treatment	COZAAR 100 mg
Treatment scheme	1 Pill * 2

taken by a BAN sensor, a smart camera placed inside the vehicle capturing an abnormal posture of the driver, a continuous or discontinuous line crossing not accompanied by a blinking and a non-response to the Cloud messages (Benadda, Bouamrane, & Belalem, 2017).

Treatment

In this fourth step, a set of periodic messages called *vigil-messages* is sent from the Cloud to vehicles (Benadda, Bouamrane, & Belalem, 2017). The objective is to check the driver's vigilance. In fact, the driver's response can, alone, determine the driver's situation and launch the decision-making step.

Decision-Making

In this fifth and last step, a set of actions is performed in case of detection of a malaise or a falling asleep, in order to manage the driver in malaise and his vehicle. These actions are the same in some cases. However, there is some difference in other ones. These actions concerns, in general, Figure 10, Figure 11 and Figure 12 represent the different algorithms which describe the different actions to perform. More detailed are available in (Benadda, Bouamrane, & Belalem, 2017).

Phase Two: Cooperation and Support Phase

This phase constitutes the second phase in the HAaaS process. In order to allow cooperation between the vehicles of the system, a mechanism called Help/rescue has been set up. This mechanism is used for possible assistance to a driver having taken malaise in order to provide first aid and therefore support him. In fact, each year, between 250 and 350 drivers could be saved if the witnesses of a road accident gave them first aid before the arrival of the medical teams (Alexandre, 2017). This phase consists of the following three steps: The send of the warning message, the broadcast of the help message and the send of the message to the nearest health unit.

The Send of the Warning Message

When a driver is taken malaise and the decision-making process is initiated namely the stopping of the vehicle, the Cloud sends to the vehicles behind the vehicle in question a *warning_message* to warn them that the vehicle with driver in malaise is stopping and moving to the emergency stop band (Figure 13).This information will be shared among the vehicles in order to adapt the driving behavior of the vehicles behind the vehicle in question and which represent the Vehicles in the Danger Zone (VDZ is defined as all vehicles, running a hazard in the event of a sudden stop or movement of another vehicle) because those at the front are not in danger.

Figure 10. Treatment and making-decision stages algorithm (main algorithm)

It is denoted by :
$V = \{V_1, V_2, V_3, \dots, V_n\}$, a set of n vehicles.
$GC = \{GC_1, GC_2, GC_3, \dots, GC_n\}$, a set of n geographic coordinates.
Each vehicle V_i is represented by a geographic coordinate GC_i.
Each geographic coordinate GC_i represents a pair of latitude and longitude. So $GC_i = (Lat_i, Long_i)$
Therefore, the total number of vehicles in a road section defined by departure and arrival coordinates GC_{dep} *and* GC_{arr} and denoted $N_{Total\text{-}Rs}$ is:

$$N_{Total-Rs} = \sum_{i=1}^{k} V_i \ \ where \ \ k \leq n \ , \quad Lat_{dep} < Lat_i < Lat_{arr} \ \ and \ \ Long_{dep} < Long_i < Long_{arr}$$

Main algorithm

Input:
$V = \{V_1, V_2, V_3, \dots, V_n\}$, a set of n vehicles.
t a time interval.
Vt_{vi} the speed of vehicle i.
Response1, Response2, Response3: the different responses to messages sent from the Cloud.
ind(1), ind(2), .., ind(x) a set of x health indicators (measurements taken by EEG, EGC, blood glucose and blood pressure, heartbeat sensors defined in case1 in malaise detection part).
Thr: the normal threshold for the different health indicators *ind(x)*.

Output:

```
                    / * Treatment * /
Each t send vigil_message;
Case 1 (response1, ind(i));
If response1=0 then
|    Re-send vigil_ message;
|    If response2=0 then
|    |
|    |    Send wake_ message;          // It is the sound message
|    |    Activate the help support in traffic queue
|    |    Send vigil_ message;
|    |    Case 4 (response3);
|    |    If response3=1 then
|    |    |    falling_asleep←1;       // That means that the driver is falling asleep
|    |    |    send alert_message;
|    |    |    Indicate the location of the nearest area rest;
|    |    End-If
|    End-If
End-If
```

Figure 11. Abnormal BAN measurement case algorithm

Case 1 (responsel, ind(i))

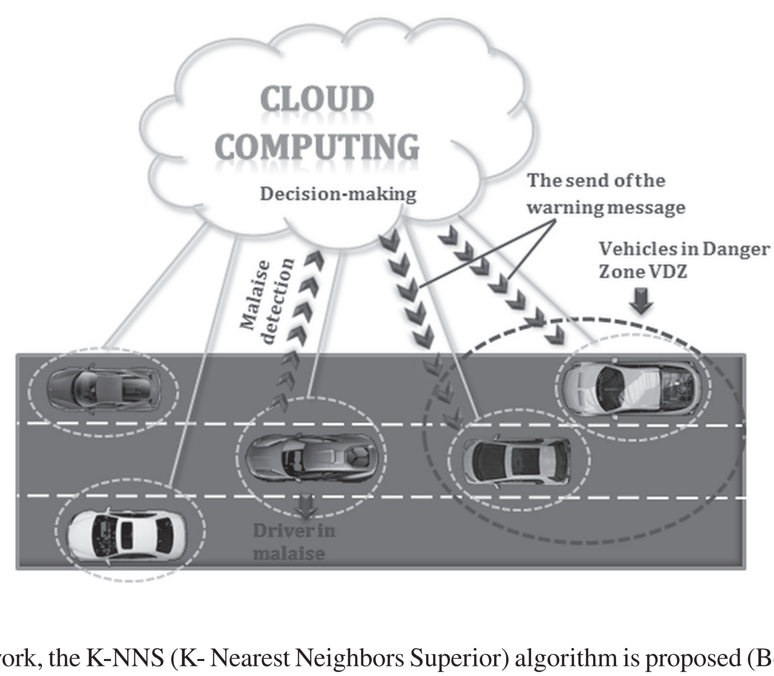
```
For i=1 to x do
|    If ind(i)≠Thr then
|        Send vigil_message;
|    |   If response1=1 then
|    |   |        / * Decision-making * /
|    |   |    Send alert_message;
|    |   |    Indicate the path of the nearest pharmacy;
|    |   End-If
|    End-If
End-For
```

Figure 12. Algorithm of the case of non-response to the Cloud messages

```
Case 4 (response3)

If response3=0 then
1:      malaise←1;                        // That means that the driver is taken malaise
                    /* Decision-making */
2:      If Vtvx is low then               //Vx is the vehicle in which the driver is in malaise
3:          For i=1 to m do               // m<n
4:              Send warning_message to Vi to avoid Vx ;  // Look Algorithm 4
5:          End-For
6:          Activate remote driving;
7:          Activate Emergency Brake Assist to stop Vx ;
8:      Else
9:          Activate AFU to Reduce Vtvx ;
10:         Repeat 3,4,5,6,7;
11:     End-If
12:         Switch on warning lights of Vx ;
13:         For i=1 to z do               // z<n
14:             Send help_message to Vi to help Vx ;   // Look Algorithm 5
15:         End-For
16:         Unlock the doors of Vx ;
17:         Send a message to emergency with the information from the form;
18: End-If
```

Figure 13. The step of sending the warning message



In this work, the K-NNS (K- Nearest Neighbors Superior) algorithm is proposed (Benadda, 2019) (Figure 14). The idea is to retrieve the GPS data from all the vehicles in the road section and, then, filter them so that only the vehicles behind the vehicle in question remain, and then apply the K-NN (K- Nearest Neighbors) algorithm.

The distance measure that is most often used in the K-NN algorithm is the Euclidean distance. However, in the proposed work, this measurement cannot be used because the authors work with

Figure 14. The K-nearest neighbors superior algorithm

```
Main algorithm

Input:
Vm: the vehicle in which the driver is in malaise
V = {V1, V2, V3,..., Vn}, a set of n vehicles.
LongVi: the longitude of vehicle i, i for 1 to n.
k the number of neighbors;

Output:
k=Cpmoy ;
i=0;
for each vehicle Vi ϵ V do
        Read Vi coordinate;
        If long (Vi)< long (Vm) than exclude Vi;
        Else
                Calculate the distance dist(Vm,Vi) // the orthodromic distance between Vm and Vi
                for the m distances obtained          // m<=n
                        Select the k vehicles which have the smallest distances
                End-For;
        End-If ;
End-for.
```

geographic data. To do this, the authors propose to take the Orthodromic Distance as the distance measure. The following formula is used:

$$arc\left(AB\right) = R * ar\cos\left[\sin\left(\varphi_A\right)\sin\left(\varphi_B\right) + \cos\left(\varphi_A\right)\cos\left(\varphi_B\right)\cos\left(\lambda_B - \lambda_A\right)\right]$$

where A (φ_A, λ_A) and B (φ_B, λ_B) two points on the earth, where φ denote latitude and λ denote longitude and R the terrestrial radius (that is 6378 kilometers, average equatorial radius).

In addition, the parameter K for the K-NN algorithm represents the number of vehicles to warn (Benadda, 2019). To find it the stopping distance of a vehicle under the conditions of the proposed system is calculated, then the average number of vehicles that can be found on this section of the highway is calculated, finally, the parameter K is defined and then the warning message can be sent.

The total stopping distance is the sum of the perception-reaction distance and the braking distance (Wikipedia, 2018a); the following formulas (Wikipedia, 2018a) are used:

$$D_{total} = D_{P-r} + D_{breaking} \tag{1}$$

$$D_{p-r} = t_{p-r} * v \ and \ DF = \frac{v^2}{2\mu g} \tag{2}$$

- t_{p-r}: time of perception-reaction (s);
- v: the speed in meters per second (m/s);
- μ: coefficient of friction between the tires and the road surface;
- g: 9,81 m/s^2 (gravity acceleration).

For the average number of vehicles that can be found on a section of road (highway) the following formula is proposed:

$$Cp_{avr} = L_{rs} \,/\, \left(Dis_{saf} + L_V\right) \times N \tag{3}$$

- L_{rs}: The length of the road section in meters (m);
- Dis_{saf}: the safety distance in a highway limited at 120 km/h;
- L_v: the average length of a vehicle (5m is proposed);
- N: The number of highway lanes.

The value of Cp_{avr} obtained is the k for the K-NN algorithm.

In the traditional K-NN algorithm the Euclidean distance is used as a metric for the proximity, in this work, it is the orthodromic distance (Wikipedia, 2018b) which is used as metric since the work is done on geographical coordinates.

The Broadcast of the Help Message

The broadcast of the help message is the second step in the cooperation and support phase of the HAaaS service process. In this phase, a help message is sent to all vehicles that are between the vehicle in which the driver is in malaise and the nearest care unit. This operation is done using geographic coordinates that the Cloud holds thanks to beacon messages that it receives periodically. The objective here is to lavish the first aid awaiting the arrival of a rescue team. In other words, it is a kind of transformation of wasted time waiting for the ambulance into a potential rescue of a human life. The correspondent algorithm is described in Figure 15.

The Send of the Message to the Nearest Health Unit

This step represents the third step of the cooperation and support phase and the last step of the HAaaS service process. In this part, the idea is to have a platform in different care units. This platform will

Figure 15. Algorithm of the broadcast of the help message

Main algorithm

Input:
Vm: the vehicle in which the driver is in malaise
$V = \{V_1, V_2, V_3, \ldots, V_n\}$, a set of n vehicles.
LongV_i: the longitude of vehicle i, i for 1 to n.
Long$_{Unit}$: the longitude of the nearest care unit.
FA_i: a boolean which is worth 1 if the driver of vehicle i knows first aids and 0 if not.

Output:

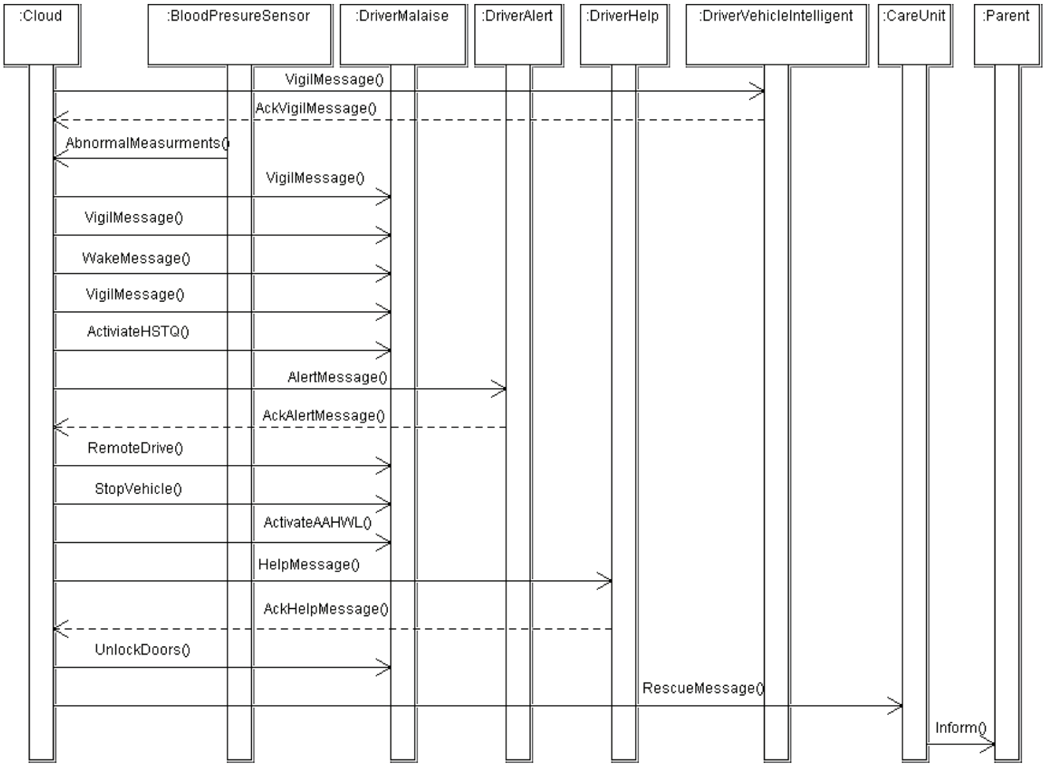
```
i=0;
for each vehicle Vi ϵ V do
    Read Vi coordinate;
    If (long (Vi)< long (Vm)) and (long (Vi)< longUnit) than exclude Vi;
    Else
        for each vehicle Vi ϵ V' do   //V' the new set of vehicles obtained
            If FAi=1 than Send Help message
            End-If;
        End-For;
    End-If ;
End-for.
```

be dedicated to the reception of messages from the Cloud concerning the vehicles whose driver is in malaise.

This message allows giving the location of the vehicle as well as a preview on the profile of the driver. The profile information is obtained thanks to the identification form that the driver filled in when he first connected to the system. The objective is to provide information that is difficult or impossible to obtain in the majority of cases of malaise and that may be indispensable in some cases, such as the prescription of drugs that cannot interact with current treatment or provoke dangerous and sometimes fatal allergies. Finally, a parent of the driver will be called through this same form.

Figure 16 shows a sequence diagram for the exchanged messages between the cloud and the vehicle with the driver in malaise and the cloud and surrounding vehicles in the case of a malaise detected by a BAN sensor and with an unconscious driver. It should be remembered that these messages exchanged with the intelligent vehicles are only representative because in reality they are multiplied by the number of vehicles in the system and are, periodically, repeated.

Figure 16. Sequence diagram representing the exchanged messages between the cloud and the vehicles

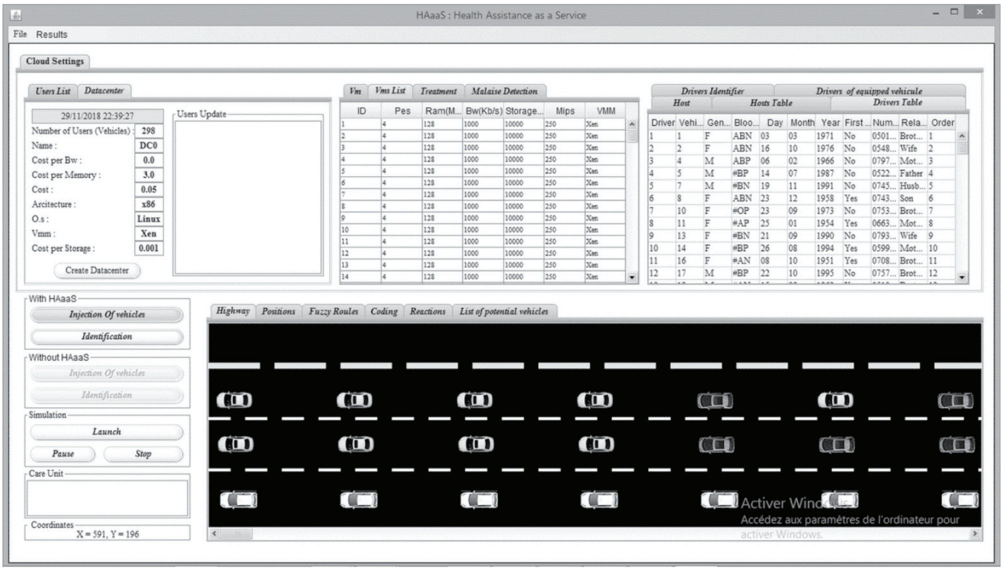


EXPERIMENTS

Simulation

In order to demonstrate and validate the efficiency of HAaaS, a simulator has been developed by the authors within the computer laboratory of Oran (LIO: Laboratoire d'Informatique d'Oran), in Algeria. The developed simulator (Figure 17) has been merged with CloudSim 3.0 simulator. The objective is to test the proposed service and measure and compare different metrics. The simulator has been developed using Netbeans 8.1 environment and the computer characteristics have been

Figure 17. Simulation of the scenario with HAaaS



as follows: Intel(R) Core(TM) 15-4200M CPU @ 2.50 GHz with 6Go RAM. Two scenarios were assumed: The first scenario simulates a malaise in a three-lane highway (Figure 17) without using the HAaaS service, the second one simulates a malaise with the same conditions as the first scenario but this time using HAaaS.

Table 2 summarizes the parameters used in the simulation. *298* represents the number of intelligent vehicles in the simulation with HAaaS service.

Table 2. The parameters used for the simulation

Parameter	Value
Road type	Highway
Road length	11.121 m
Initial number of vehicle	447
Initial number of node	298
Speed limit	120 km/h

In the first scenario, different colors are used for vehicles to differentiate their type. Green is used for intelligent vehicles equipped with BAN sensors; yellow is used for those which are not equipped with these BAN sensors and white for non-intelligent vehicles.

The highway section from El Kerma to Oued Tlélat in Oran city in Algeria was considered as the road model (Figure 18).

During simulation, the longitude, latitude, and the lane number are update for all vehicles in the system by the Cloud. If a vehicle leaves the road it will be considered as *Out of system* and *Active* if not. The parameters that were used for CloudSim configuration are shown in Table 3.

Figure 18. El Kerma -Oued Tlélat road model

Users List | *Datacenter*

Vehicle	Locality	Lane	Latitude	Longitude	Type	BAN
1	El Kerma	1	35.59727...	-0.57273611...	Intelligent	Yes
2	El Kerma	2	35.59719...	-0.57268783...	Intelligent	No
3	El Kerma	3	35.59706...	-0.57263888...	Non intellig...	
4	El Kerma	1	35.59647...	-0.57264223...	Intelligent	No
5	El Kerma	2	35.59634...	-0.57259194...	Intelligent	No
6	El Kerma	3	35.59626...	-0.57254031...	Non intellig...	
7	El Kerma	1	35.59568...	-0.57255908...	Intelligent	No
8	El Kerma	2	35.59555...	-0.57250745...	Intelligent	No
9	El Kerma	3	35.59544...	-0.57245180...	Non intellig...	
10	El Kerma	1	35.59482...	-0.57247862...	Intelligent	No
11	El Kerma	2	35.59479...	-0.57243369...	Intelligent	No
12	El Kerma	3	35.59466...	-0.57237401...	Non intellig...	
13	El Kerma	1	35.59408...	-0.57240083...	Intelligent	Yes

Table 3. The used parameters for CloudSim

	Datacenter	Users	Virtual Machines (VM)	Host
PE	/	/	1	4
Ram	/	/	128	512
Storage	/	/	10000	1000000
Configuration	01	298	298	75

Also, the Cloud calculates, updates, and displays the number of users, virtual machines, and hosts according to the number of vehicles that leave or enter the road. The global list of drivers used for simulation is shown in Figure 19.

In the two scenarios (with and without HAaaS), when a malaise happens, the vehicle becomes red, although, in the first scenario this red vehicle is managed and put out of danger, in the second

Figure 19. An overview of the global list of drivers used in the simulation

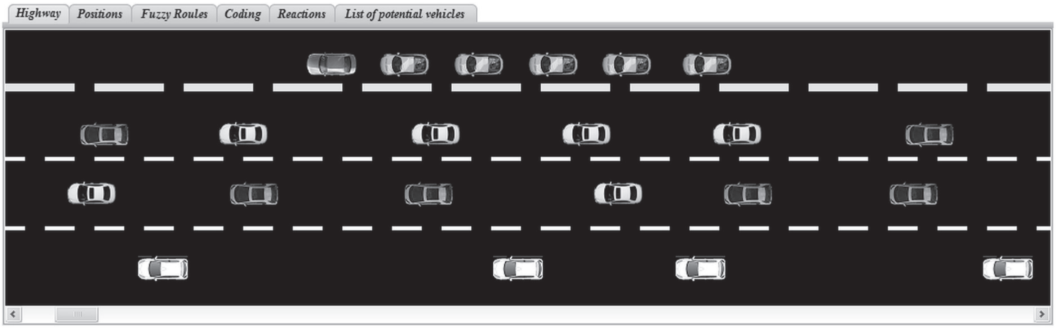
Drivers Identifier | *Drivers of equipped vehicule*
Host | *Hosts Table* | *Drivers Table*

Driver	Vehi...	Gen...	Bloo...	Day	Month	Year	First ...	Num...	Rela...	Order
1	1	F	#OP	17	01	1991	No	0774...	Daug...	1
2	2	F	#BP	03	10	1986	Yes	0702...	Son	2
3	4	F	#BP	11	08	1966	Yes	0648...	Sister	3
4	5	M	#AN	12	10	1981	No	0762...	Son	4
5	7	F	ABN	20	03	1970	No	0688...	Son	5
6	8	F	#BN	09	10	1992	Yes	0500...	Father	6
7	10	F	ABP	24	11	1958	Yes	0776...	Son	7
8	11	M	ABP	14	11	1979	No	0528...	Daug...	8
9	13	M	#AP	25	10	1964	No	0534...	Wife	9
10	14	F	ABN	21	03	1971	No	0613...	Son	10
11	16	M	#BP	20	12	1976	No	0715...	Daug...	11
12	17	M	ABN	19	08	1995	Yes	0516...	Wife	12

scenario, no control on the vehicle is performed and therefore the vehicle skids and causes at least a material damage. This latter can go from the simple collision to a dangerous pileup.

Also, after the send of the help message in the first scenario, a set of blue vehicles come to park after the red vehicle to give the diver first aid (Figure 20) however in the second scenario, no message is sent and therefore the driver in malaise, and the drivers of the eventual vehicles involved in the collisions, must wait for the arrival of the ambulance.

Figure 20. Arrival of vehicles for first aid provision



Results

In terms of reducing the number of traffic accidents; the gain that the HAaaS service would bring to the roads in general (Figure 21) and the highways especially (Figure 22) has been calculated. Indeed, this gain represents 9% for all roads combined (APR, 2016) and 27% for the highways (ASFA, 2018). Figure 21 shows that in Algeria, the number of traffic accidents was 59.902, in 2016, and that by integrating HAaaS service this number would have been 54.511, a reduction of 5.391 accidents per year. Knowing that these accidents involved, in 2016, a coefficient of 1.38 damaged vehicles so a gain of 7.440 vehicles with HAaaS.

Figure 22 shows that on the A1 highway in Algeria, the number of traffic accidents was 3,791 in 2016, and that by integrating HAaaS service this number would have been 2,767, a reduction of 1,024 accidents per year involving a gain of 7.440 vehicles with HAaaS.

Figure 21. Impact on the total number of traffic accidents

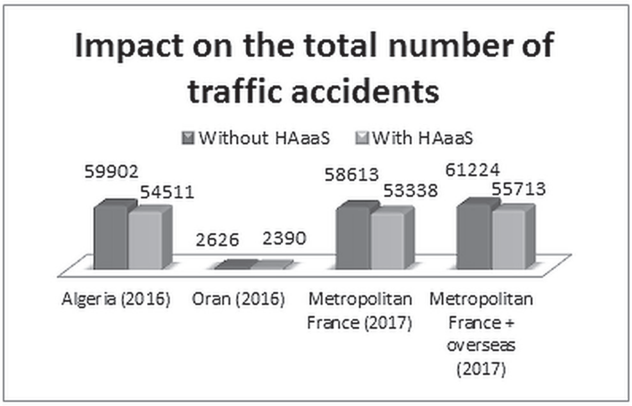
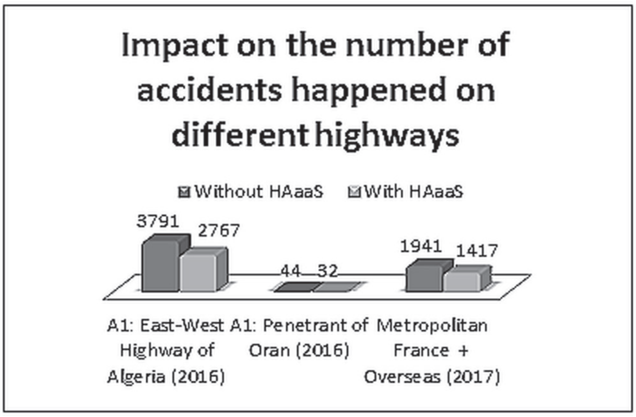


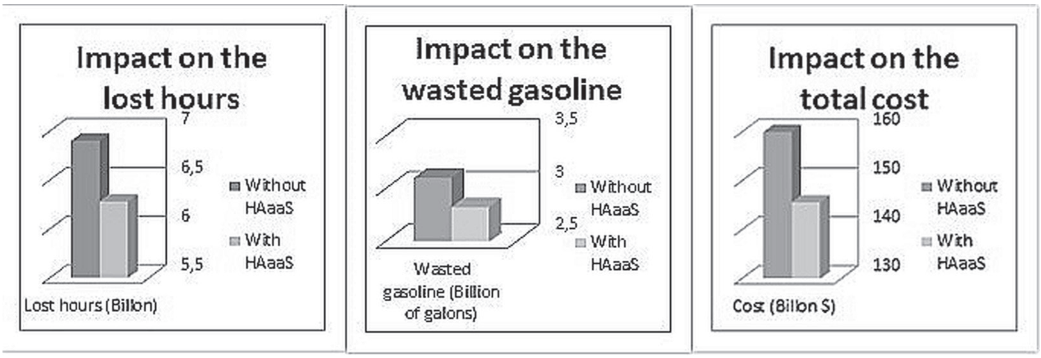
Figure 22. Impact on the number of traffic accidents happened on different highways



Also, the gain that this approach would bring to the roads in term of lost hours, gasoline wasted and costs (Schrank, Eisele, Lomax, & Bak, 2015), has been calculated (Figure 23).

Figure 23 shows that the HAaaS integration would allow a gain of a gain of 621 million hours, 279 million gallons and $ 14.4 billion per year in the United States only.

Figure 23. Impact on the lost hours, the wasted gasoline and the total cost



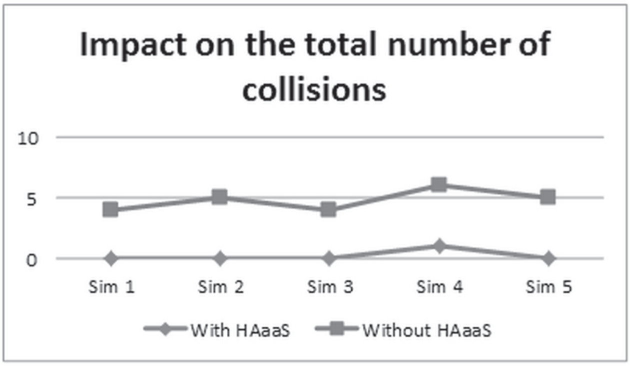
Finally, the two different situations were analyzed to determine the possibility of a collision (Figure 24). In the first situation, HAaaS has been used and no collision happened, because the vehicle with the driver in malaise was managed by the Cloud, excepted one simulation where a car was parked in the emergency stop band. In the second situation no HAaaS has been used and in each simulation, collisions happened because the drivers were not alerted for the sudden stop of the vehicle whose driver was in malaise.

The authors should note that the simulation results can be used to justify further researches.

Issues, Controversies, Problems

The research works analysis shows that very few of the analyzed works were interested in both road safety and health monitoring ((Wang et al., 2011) and (Shen et al., 2018)). For the first one, no simulation has been performed, as to the second one the detection concerns the drunk and fatigue driving and no driver's and vehicle's management or rescue approaches were proposed.

Figure 24. Impact on the total number of collisions



Also, it has been noticed that BAN technology has been used in a Vehicular Cloud Computing environment only by (Wang et al., 2011) and that none of the works that deal with road safety was interested in people with chronic diseases that drive vehicles. The main difference between the proposed work and previous works is that the detection of the malaise behind the wheel is the objective of this work unlike the analyzed works. Also, the cooperation aspect has been dealt with from the point of view of data sharing (Mousannif, Khalil, & Al Moatassime, 2011) but not for the rescue of drivers. Finally, for all the works cited in this paper, only one metric, at a time, has been supported for the evaluation of the proposed approaches. Consequently and unlike the other research works, in this paper, the contribution was provided in the road safety and health monitoring areas, the phenomenon of malaise behind the wheel has been the research subject of this study and the two technologies of Body Area Networks and Vehicular Cloud Computing were combined. Also, the cooperation aspect has been dealt with and many metrics have been considered in the simulation of the different scenarios, namely: collisions, time, cost, and human and material damages. Although the proposed service proved to be efficient, several issues remain to be addressed such as; the priority problem, when many malaises are detected by the Cloud at the same time, and the load balancing problem and finally the security and privacy of the collected data problem.

CONCLUSION

Through this paper, the authors presented a new service to deal with the problem of malaise behind the wheel. The authors proposed a Vehicular Cloud Computing service for the support and the management of the driver's malaise and vehicle. This service that they have called HAaaS "Health-Assistance as a Service" is able to detect a driver's malaise or sleepiness while driving the vehicle. Also, this service provides a cooperation mechanism between vehicles to help the driver taken malaise as soon as possible. From their point of view, the authors consider that HAaaS can both be considered as:

- A system for detecting malaise (or a falling asleep);
- A vehicle management support;
- A support system for drivers who are victims of driving malaise;
- A cooperation mechanism for driver assistance;
- An information mechanism for care units for the dispatch of ambulances.

As results, the authors explain that their contributions would make it possible to:

- Reduce the total number of traffic accidents by reducing the number of accidents caused by malaise;
- Reduce the number of deaths by avoiding the skidding of vehicles of drivers in malaise and the occurrence of collisions;
- Reduce material damage by avoiding collisions thanks to the vehicle management system and warning messages;
- Reduce the time and fuel lost in traffic jams caused by the skidding of vehicles of drivers in malaise;
- Reduce the number of unnecessary exits from care unit ambulances thanks to the information mechanism;
- Avoid blocking ambulances in traffic thanks to the vehicle management system which makes it possible to place the vehicle of the driver in malaise on the emergency stop strip and thus avoid traffic jams;
- Have exact statistics on the number of malaise or falling asleep on the road, the ages of the drivers involved and their health state. These statistics can be used to study the phenomenon of malaise driving and its various causes;

In addition, the authors explain that a study is being carried out to measure the produced errors rate and their influences on the results.

Perspectives

As prospects, the authors plan to use the simulation results to perform further researches. As short-term work they plan to:

- Study the complexity of the proposed algorithms;
- Introduce the concepts of Fog computing and Edge computing to minimize the occurrence of errors when transmitting data between vehicles and the cloud, and when detecting malaise;
- Consider other criteria for the detection of malaise or a falling asleep, for example the use of the Face Reading Technologies (FRTs) (Ding et al., 2019);
- Use a VANet simulator such as SUMO while using a simulator Cloud such as CloudSim in order to improve the implementation of their service;
- Equip all vehicles with BAN sensors for a total coverage of all possible malaises.

A long-term work will consist on the extension of the work field to other types of roads, such as expressways, national roads or agglomerations and the consideration of a completely smart road.

ACKNOWLEDGMENT

We thank Lieutenant Bellala Abdelkader responsible of communication for the Directorate of Civil Protection of Oran wilaya for providing us with the necessary data concerning the traffic accidents in Oran and in Algeria and those concerning the different units of civil protection in Oran Wilaya.

REFERENCES

Abatal, A., Khallouki, H., & Bahaj, M. (2018). A Smart Interconnected Healthcare System using Cloud Computing. *Proceedings of LOPAL '18: the International Conference on Learning and Optimization Algorithms: Theory and Applications*. 1-5. doi:10.1145/3230905.3230936

Abuelela, M., & Olariu, S. (2010). Taking VANET to the Clouds. *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia (MoMM2010).*

Alexandre, S. (2017, November 23). Permis de conduire: test obligatoire des gestes de premiers secours. *LeFIGARO*. Retrieved from http://leparticulier.lefigaro.fr/

Alshareef, H., & Grigoras, D. (2018, January). Swift personal emergency help facilitated by the mobile cloud. *International Journal of High Performance Computing and Networking*, *12*(1), 1–12. doi:10.1504/IJHPCN.2018.093837

APR. (2016, April 24,). Retrieved from https://www.preventionroutiere.asso.fr/Nos-publications/Statistiques-d-accidents/Principaux-facteurs-d-accidents

ASFA. (2018, August 2). *Key figures 2017*. Retrieved from https://www.autoroutes.fr/FCKeditor/UserFiles/File/ASFA_cles17.pdf

Bektache, D. (2014). *Application et Modélisation d'un protocole de communication pour la sécurité routière* (PhD thesis). Retrieved from http://biblio.univ-annaba.dz/wp-content/uploads/2015/10/these-Bektache-Djamel-.pdf

Benadda, M. (2019, March 7). *Le Cloud Computing pour une Gestion Intelligente du Trafic Urbain: Proposition d'un service de prise en charge et de coopération pour la gestion de personnes victimes de malaise pendant la conduite* (PhD thesis). Retrieved from https://hal.archives-ouvertes.fr/tel-02519803

Benadda, M., Bouamrane, K., & Belalem, G. (2017, April-June). How to manage persons taken malaise at the steering wheel using HAaaS in a Vehicular Cloud Computing environment. *International Journal of Ambient Computing and Intelligence*, *8*(2), 70–87. doi:10.4018/IJACI.2017040105

Bitam, S., & Mellouk, A. (2012, December). ITS_Cloud: Cloud Computing for Intelligent Transportation System. *Proceeding of Global Communications Conference (GLOBECOM).*

Ding, X., Jiang, Y., Qin, X., Chen, Y., Zhang, W., & Qi, L. (2019). Reading Face, Reading Health: Exploring Face Reading Technologies for Everyday Health. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*. Glasgow, UK: ACM. doi:10.1145/3290605.3300435

Ding, X., Jiang, Y., Qin, X., Chen, Y., Zhang, W. & Qi, L. (2019). Reading Face, Reading Health: Exploring Face Reading Technologies for Everyday Health. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1-13.

Mousannif, H., Khalil, I., & Al Moatassime, H. (2011). Cooperation as a Service in VANETs. *The Journal of Universal Computer Science J. UCS*, *17*(8), 1202–1218.

Sam, D., Evangelin, E., & Cyril Raj, V. (2015, June). Improving road safety for pedestrians in black spots using a hybrid VANET of vehicular sensors and Pedestrian body unit. *Journal of Engineering and Applied Sciences (Asian Research Publishing Network)*, *10*(10), 4639–4644.

Schrank, D., Eisele, B., Lomax, T., & Bak, J. (2015, August). *2015 Urban Mobility Scorecard.* Texas Transportation Institute, Texas A&M University System. College Station, TX.

Schrank, D., Eisele, B., Lomax, T., & Bak, J. (2019, August). *2019 Urban Mobility Scorecard.* Texas Transportation Institute, Texas A&M University System. College Station, TX.

Shen, X., Chen, J.-Y., Lei, X.-C., Huang, L., Wu, L.-S., & Yin, W.-Q. (2018, October). Smart Safety Monitoring System for Vehicles. *Proceedings of BDIOT 2018: the 2018 2nd International Conference on Big Data and Internet of Things*, 84–87. doi:10.1145/3289430.3289467

Wang, J., Cho, J., Lee, S., & Ma, T. (2011). Real Time Services for Future Cloud Computing Enabled Vehicle Networks. *Proceeding of International Conference on Wireless Communications and Signa Processing (WCSP)*, 1-5. doi:10.1109/WCSP.2011.6096957

Whaiduzzaman, Md., Sookhak, M., Gani, A., & Buyya, R. (2014, April). A survey on vehicular cloud computing. *Journal of Network and Computer Applications, 40*, 325-344.

WHO. (2020, March 21). Retrieved from https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

Wikipedia. (2018a, December 21). *Braking distance*. Retrieved from https://en.wikipedia.org/wiki/Braking_distance

Wikipedia. (2018b, November 15). *Orthodromie*. Retrieved from https://fr.wikipedia.org/wiki/Orthodromie

Wikipedia. (2019, January 12). *Body Area Network*. Retrieved from https://fr.wikipedia.org/wiki/Body_Area_Network

Meriem Benadda graduated from the Department of computer science, Faculty of exact and applied sciences, University Oran1 Ahmed Ben Bella, Algeria, where she received her PhD degree in computer science in 2019. Her current research interests are Cloud Computing, Body Area Networks, malaise behind the wheel, health monitoring, Wireless Sensors, Intelligent Transportation Systems, and VANet.

Ghalem Belalem graduated from Department of computer science, Faculty of exact and applied sciences, University Oran1 Ahmed Ben Bella, Algeria, where he received PhD degree in computer science in 2007. His current research interests are distributed system; grid computing, cloud computing, replication, consistency, fault tolerance, resource management, economic models, energy consumption, Big data, IoT, mobile environment, images processing, Supply chain optimization, Decision support systems, High Performance Computing.

International Journal of Software Science and Computational Intelligence
Volume 12 • Issue 4 • October-December 2020

Information Processing Systems in UAV Based on Bayesian Filtering in Conditions of Uncertainty

Rinat Galiautdinov, Independent Researcher, Italy
https://orcid.org/0000-0001-9557-5250

ABSTRACT

In this article, the author considers the possibility of applying modern IT technologies to implement information processing algorithms in UAV motion control system. Filtration of coordinates and motion parameters of objects under a priori uncertainty is carried out using nonlinear adaptive filters: Kalman and Bayesian filters. The author considers numerical methods for digital implementation of nonlinear filters based on the convolution of functions, the possibilities of neural networks and fuzzy logic for solving the problems of tracking UAV objects (or missiles), the math model of dynamics, the features of the practical implementation of state estimation algorithms in the frame of added additional degrees of freedom. The considered algorithms are oriented on solving the problems in real time using parallel and cloud computing.

KEYWORDS

AI, Artificial Intelligence, Bayesian Filter, Drone, Nonlinear Filtering, UAV

INTRODUCTION

The growth and development of the IT sphere and the power of the computers allows to apply more advanced methods of information processing in the sphere of UAV and air planes. The major algorithms for information processing in such the spheres are filtering and extrapolation of the parameters of the trajectories of objects upon their detection and tracking according to measurement data.

The greatest interest is represented by the implementation of numerical methods for nonlinear filtering of dynamic processes under conditions of uncertainty and insufficiency of a priori information since the quality of the state vector estimate directly affects the operation of a closed control system (Bain, A. & Crisan, D., 2009). To provide feedback in such a system, the control object is equipped with meters, with which you can determine the components of the state vector, while the measurements have limited accuracy. To improve the quality of state assessment, special algorithms are used that reduce the measurement noise, taking into account its parameters and the dynamics of the control object. In the case of nonlinear systems, for example, the methods of nonlinear Kalmanovskaya and Bayesian filtering, which are successfully used to solve problems of assessing the state of aircraft and other objects, are most popular.

DOI: 10.4018/IJSSCI.2020100103

Until recently, their practical value was small, since many algorithms were unrealizable due to the stringent requirements of real-time processing. Therefore, analytical approximations in the form of linear Kalman – Bucy filters, which make it possible to solve problems in real time, are most widely used in practice (Bergemann, K. & Reich, S., 2012).

The development of adaptive filtering methods for nonlinear dynamic processes described by a not fully known and time-varying mathematical model under the action of random perturbations with a changing family of probability densities is extremely important. One of the effective ways to solve the problem is the Bayesian algorithm. Since information on external influences is inaccurate for adaptive nonlinear filtering problems, such artificial intelligence methods as neural networks and fuzzy logic can serve as an excellent complement to traditional methods of statistical synthesis of adaptive nonlinear filtering systems. In this regard, the possibilities of using neural networks, fuzzy logic and knowledge bases when implementing numerical algorithms for adaptive nonlinear filtering to increase their efficiency are considered. Although an extended Kalman filter will be briefly considered here, we take the Bayesian approach to the synthesis of discrete nonlinear filtering algorithms, which allows us to completely solve the problem of both linear and nonlinear filtering in discrete time.

Advanced Kalman Filter

The extended Kalman filter uses the model of a stochastic continuous system:

$$x^{\cdot} = (x, t) + w \tag{1}$$

and discrete measurements:

$$z_k = h_k(x(t_k)) + y_k \tag{2}$$

where x is the state vector of the system, w is the noise of the system, y_k is the measurement noise.

The task of filtering is to find a measurement function z_k an estimate of the state vector of the x'_k system that minimizes the variance of the error x'_k-x (t_k).

Let an estimate of the state vector x'_{k-1} be obtained at time t_{k-1}. Based on this estimate, the forecast of the state vector estimate $x'_{k'}$ (a priori estimate) is built, then z_k measurements and correction of the estimate a priori based on $x'_{k'}$ 'measurement results (a posteriori estimate) are carried out (Wilson, R. C. & Finkel, L., 2009). An estimate a priori of the state vector x'_k 'is calculated by integrating the model Equation (3):

$$\frac{dx'}{dt} = f\left(x', t\right) \tag{3}$$

with initial conditions $x'(t) = x'_{k-1}$, $x'(0) = x_0$.

The a priori estimate for the covariance error matrix for the linearized equations in P'_k increments is calculated as (4), (5), (6):

$$P_{k'} = \varphi P_{k''-1} \varphi^T + Q \tag{4}$$

$$F = \frac{df\left(x, t\right)}{dx} \mid x = x'_{k-1} \tag{5}$$

$$\varphi = E + F\delta t \tag{6}$$

with the initial conditions *P'(t) = Pk''−1, P(0) = P0*. Where, *Q* is the covariance matrix of the noise of the system (Doucet, A. & Johansen, A., 2009). A posteriori estimate for the state vector and covariance matrix errors are constructed as follows:

$$x'_{k''} = x'_{k'} + K_k(Z_k - H_k x'_{k'}) \tag{7}$$

$$Pk'' = (I - K_k H_k)\, Pk' \tag{8}$$

H_k –linearized sensitivity matrix (9):

$$H_k = \frac{\partial h\left(x,t\right)}{\partial x} \mid x = x'_{k-1} \tag{9}$$

K_k –feedback correction matrix (10):

$$K_k = Pk'\, H^T_k [Hk\, Pk'\, H^T_k + R]^{-1} \tag{10}$$

where:

R –covariance matrix of noise measurements

Bayesian Approach to Solving Problems of Adaptive Nonlinear Filtering Under a Priori Uncertainty

For discrete observations, we write down nonlinear models of the processes of dynamics and measurements in the form of the following three difference vector equations (Doya, K., et al., 2007):

$$x_k = f_k\left(x_{k-1}, a_{k-1}\right) + g_k\left(x_{k-1}\right)\xi_{k-1} \tag{11}$$

$$z_k = h_k\left(x_k, a_k\right) + u_k \tag{12}$$

$$a_k = \varphi_k\left(a_{k-1}\right) + \sigma_{k-1} \tag{13}$$

where:

- k is the moment in time;
- X_k- is a stochastic n-dimensional vector;

- a_k - is the vector of accompanying parameters of dimension r;
- $f_k(x_{k-1}, a_{k-1})$ - in the general case, a vector column function of dimension n non-linear with respect to its arguments;
- $g_k(x_{k-1})$ - is a matrix function of dimension n * l;
- ξ_{k-1} - is a sequence of independent l-dimensional random vectors of forming noise with density $p(\xi_{k-1})$;
- z_k is the m-dimensional observation vector;
- $h_k(x_k,a_k)$ - in the general case, a vector column function of dimension m non-linear with respect to its arguments;
- u_k - is a sequence of independent m-dimensional random measurement noise vectors with density $p(u_k)$;
- $\varphi_k(a_{k-1})$ - in the general case, a column vector function of dimension r non-linear with respect to its argument;
- σ_{k-1} - is a column of vector noise of the drift of parameters, which are purely random processes, has the dimension r and is described by the probability density $p(\sigma_{k-1})$.

The posterior probability density $p(x_0, a_0)$ at the zero instant of time coincides with the a priori probability density $p_{pr}(x_0, a_0)$ of the vectors x and a.

The formulation of the adaptive estimation problem: it is necessary to obtain optimal estimates of the stochastic vector dynamic process x_k from the available vector observations z_k with a lack of a priori information about the vector of accompanying parameters a_k and the characteristics of the noise of the measurement channel v_k and the forming noise σ_k. Noises may depend on x_k and a_k. The vector a_k, which is to be identified, takes into account a priori uncertainty in the specification of the model of the process being evaluated and perturbations (Galiautdinov, R., 2020).

Bayesian Methodology for Solving the Problem

Assume that the posterior probability density $p(x_{k-1}, a_{k-1} \mid z_0^{k-1})$ for the time t_{k-1} is found. Here z_0^{k-1} denotes a sequence of observations for time instants: $t_0, t_1, \ldots t_{k-1}$. It is necessary to find the posterior probability density $p(x_k, a_k \mid z_0^{k-1})$ of the extrapolated values of x_k and a_k for the next time moment t_k in the absence of the observation z_k, the posterior probability density $p(x_k, a_k \mid z_0^{k})$ for the next time t_k after obtaining the reference observations z_k and a posterior probability density $p(x_{k+1}, a_{k+1} \mid z_0^{k})$ depending on $p(x_k, a_k \mid z_0^{k-1})$.

Accordingly, for the adaptive estimation problem, the following formulas were obtained for the recursive calculation of the conditional posterior probability density of the state vector of the system (14), (15), (16):

$$p(x_k, a_k \mid z_0^{k-1}) = \int_{-\infty}^{\infty} p\left[x_{pk} - f_{pk}\left(x_{pk-1}\right)\right] p(x_{k-1}, a_{k-1} \mid z_0^{k-1}) dx_{k-1} da_{k-1} \tag{14}$$

$$\begin{aligned} &p(x_k, a_k \mid z_0^{k}) \\ &= c^{-1}\left(k\right) p\left[z_k - h_k\left(x_k, a_k\right)\right] \left[\int_{-\infty}^{\infty} p\left[x_{pk} - f_{pk}\left(x_{pk-1}\right)\right] p(x_{k-1}, a_{k-1} \mid z_0^{k-1}) dx_{k-1} da_{k-1}\right] \end{aligned} \tag{15}$$

$$\begin{aligned} &p(x_{k+1}, a_{k+1} \mid z_0^{k}) \\ &= c^{-1}\left(k\right) \int_{-\infty}^{\infty} p\left[x_{pk+1} - f_{pk+1}\left(x_{pk}\right)\right] p\left[z_k - h_k\left(x_k, a_k\right)\right] p(x_k, a_k \mid z_0^{k-1}) dx_k da_k \end{aligned} \tag{16}$$

Here we use the following notation for the extended state vector (17):

$$x_{pk} = \begin{bmatrix} x_k \\ a_k \end{bmatrix}, f_{pk} = \begin{bmatrix} f_k\left(x_{k-1}, a_{k-1}\right) \\ \varphi_k\left(a_{k-1}\right) \end{bmatrix}, \xi_{pk} = \begin{bmatrix} \xi_k \\ \delta_k \end{bmatrix} \tag{17}$$

Knowing the posterior probability density allows you to find an estimate of the state vector of the system by any criterion, including the criterion of the minimum mean square error (Greaves-Tunnell, A., 2015). Formulas, in principle, completely solve the problem of both linear and nolinear filtering in discrete time, however, the solution of practical problems requires consideration of numerical algorithms in each specific case to obtain estimates and their corresponding covariance matrices (Huang, Y. & Rao, R. P. N., 2016). Since without considering specific practical examples, little can be said about the features of nonlinear filtering algorithms, in the future we will consider as an example the estimation of the motion parameters of maneuvering air objects (Kording, K. P. & Wolpert, D. M., 2004).

To solve the problem of tracking a maneuvering target, we take as a basis a variant of an adaptive algorithm for filtering the parameters of the trajectory of a maneuvering target based on the Bayesian approach (Sokoloski, S., 2015). A linear dynamic system is considered as a model of the target trajectory and linear measurements are used, and a linear Kalman filter is used to estimate the state vector (Moreno-Bote R., et al., 2011). Taking into account expressions (14) - (17), we generalize the result to a nonlinear dynamical system with nonlinear measurements and a nonlinear discrete optimal filter in order to obtain the maximum gain in accuracy in comparison with linear adaptive and non-adaptive models. As a model of indefinite perturbation, we choose a model of the semi-Markov process (Cappe, O., 2011).

The scalar perturbation *a* - is represented as a random process, the average value of which $M[a_k]$ changes stepwise, taking a number of fixed values (states) in the range from $-a_{max}$ to a_{max}.

Transitions of the jump-like process from state *i* to state occur with probability π_{ij} determined by a priori data. The residence time of the process in state *i* until the transition to state *j* is a random variable with an arbitrary distribution density $p(t_i)$. As a special case of the semi-Markov process, we can consider a pulsed random process, which is a sequence of rectangular video pulses with random amplitudes having a density *p(a)*, with random durations and intervals between them having an arbitrary distribution density (Galiautdinov, R. et al., 2019A). The Bayesian approach to constructing an adaptive filtering algorithm will be considered initially for the case of continuous perturbation *a*.

The problem of optimal estimation of the vector parameter x'_k for the quadratic loss function reduces to a weighted averaging of the estimates $x'_k\ (a_k)$, which are the solution to the filtering problem for fixed values of a_k (18):

$$\begin{aligned} x'_k &= \int_{(x)} x_k p(x_k \mid z_0^{\ k}) dx_k \\ &= \int_{(x)} x_k \int_{(a)} p(x_k, a \mid z_0^{\ k}) da dx_k \\ &= \int_{(x)} x_k \int_{(a)} p(x_k, a \mid z_0^{\ k}) p(a \mid z_0^{\ k}) da dx_k \\ &= \int_{(a)} x'\left(a\right) p(a \mid z_0^{\ k}) da_k \end{aligned} \tag{18}$$

where:

- *(X)* is the space of possible values of the estimated parameter;
- *a* is the range of possible values of the perturbing parameter,

- b is the $p(x_k \mid z_0^k)$ posterior probability density of the vector x_k according to the data of the $k + 1$-dimensional measurement sequence z_0^k;
- c is $p(x_k, a \mid z_0^k)$ - the posterior probability density of the vector x_k in the presence of the perturbing parameter a.

Estimates $x'_k(a)$ can be obtained in any way that minimizes the standard error of the mean square error, including using a recurrent linear Kalman filter (Galiautdinov, R. et al., 2019B).

The problem of optimal adaptive filtering will be solved if at each step the a posteriori probability density p of the perturbing parameter $p(a_k, z_0^k)$ is calculated. The calculation of this density from the sample of measurements z_0^k and its use for obtaining weighted estimates is the main feature of the adaptive filtering method, which we will use. In the case when the perturbing parameter takes only fixed values a_j $(j = -\mu/2, ..., -1, 0, 1, ..., \mu/2, \mu$ *is even),* instead of Expression (18), we can write (19):

$$x'_k = \sum_{j=-\mu/2}^{\mu/2} x'_k \left(a_{j,k}\right) P\left(a_{j,k} \mid z^k{}_0\right) \tag{19}$$

where $P(a_{j,k} \mid z_0^k)$ is the posterior probability of the event $-a_{j,k} = a_j$ according to the measurements of z_0^k.

To calculate the posterior probability $P(a_{j,k}|z_0^k)$ – we use the Bayes rule and consequently get (20):

$$P(a_{j,k} \mid z^k 0) = P_{k,j} = \frac{P(a_{j,k} \mid z^{k-1}{}_0)p(z_k \mid a_{j,k-1})}{\sum_{j=-\mu/2}^{\mu/2} P(a_{j,k} \mid z^{k-1}{}_0)p(z_k \mid a_{j,k-1})} \tag{20}$$

Here, according to the general scheme for applying the Bayes formula, the $k + 1$-dimensional sequence of measurements z_0^k is considered as event B.

In this expression, $p(x_k \mid z_0^{k-1})$ is the a priori probability of the parameter a_j at the k-step, obtained from the results of $k-1$ measurements and calculated by the Formula (21):

$$P(a_{j,k} \mid z^{k-1}{}_0) = \sum_{j=-\mu/2}^{\mu/2} \pi_{ij} P\left(a_{i,k-1} \mid z^{k-1}{}_0\right) \tag{21}$$

where $\pi_{ij} = P(a_k = a_j \mid a_{k-1} = a_i)$ is the conditional probability of the perturbing process transitioning from state i to $(k-1)^{\text{th}}$ step in state j at the k-step; $p(z_k \mid a_{j,k-1})$ is the conditional probability density of the measured value of the coordinate z_k, if the perturbing parameter at the previous step $k-1$ had the value a_j.

In view of (11), we obtain the following expression for the posterior probability $P_{k,j}$ (22):

$$P_{k,j} = \frac{\sum_{i=-\mu/2}^{\mu/2} \pi_{ij} P(a_{i,k-1} \mid z^{k-1}{}_0)p(z_k \mid a_{j,k-1})}{\sum_{j=-\mu/2}^{\mu/2} \sum_{i=-\mu/2}^{\mu/2} \pi_{ij} P(a_{i,k-1} \mid z^{k-1}{}_0)p(z_k \mid a_{j,k-1})} \tag{22}$$

The values of $P_{k,j}$ for each j are weighting coefficients when averaging the estimates of the filtered parameter.

In contrast to (13), to obtain the estimates $x'_k(a_k)$, we use the optimal discrete nonlinear filter (14) - (17).

In Equations (14) - (17), the paratrometer a plays the role of a "fixed state" of an indefinite disturbance; f_k, g_k, h_k are known.

We use Formulas (14) - (17) to calculate estimates of the state of a maneuvering target.

It is believed that the posterior probability density $p[x_{k-1}(a_{j,k-1}) \mid z_0^{k-1}]$ for the time t_{k-1} is found. It is necessary to find the posterior probability density $p[x_k(a_{j,k}) \mid z_0^k]$ of the extrapolated values of x_k for the following time instants t_k in the absence of an observation count z_k, the posterior probability density $p[x_k(a_{j,k}) \mid z_0^k]$ for the next time t_k after obtaining the observation z_k.

The sought posterior probability density of the extrapolated values of x_k in the absence of an observation z_k in accordance with (4) has the following form (23):

$$p\left[x_k\left(a_{j,k}\right)\middle|z_0^{\,k-1}\right]=\int_{-\infty}^{\infty}p\left\{x_k\left[a_{j,k}\right]-f_k\left[x_{k-1}\left(a_{j,k-1}\right)\right]\right\}p\left[x_{k-1}\left(a_{j,k-1}\right)\middle|z_0^{\,k-1}\right]dx_{k-1} \tag{23}$$

The sought posterior probability density for the next time moment t_k after obtaining the observation count z_k in accordance with (5) has the form (24):

$$\begin{aligned}&p\left[x_k\left(a_{j,k}\right)\middle|z_0^{\,k}\right]\\&=c^{-1}\left(k\right)p\left\{z_k-h_k\left[x_k\left(a_{j,k}\right)\right]\right\}\int_{-\infty}^{\infty}p\left\{x_k\left[a_{j,k}\right]-f_k\left[x_{k-1}\left(a_{j,k-1}\right)\right]\right\}\\&p\left[x_{k-1}\left(a_{j,k}\right)\middle|z_0^{\,k-1}\right]dx_{k-1}\end{aligned} \tag{24}$$

The expression of the posterior density $p[x_{k+1}(a_{j,k+1}) \mid z_0^k]$ depending on $p[x_k(a_{j,k}) \mid z_0^{k-1}]$ according to (6) has the form (25):

$$\begin{aligned}&p\left[x_{k+1}\left(a_{j,k+1}\right)\middle|z_0^{\,k}\right]\\&=c^{-1}\left(k\right)\int_{-\infty}^{\infty}p\left\{x_{k+1}\left(a_{j,k+1}\right)-f_{k+1}\left[x_k\left(a_{j,k}\right)\right]\right\}p\left\{z_k-h_k\left[x_k\left(a_{j,k}\right)\right]\right\}\\&p\left[x_k\left(a_{j,k}\right)\middle|z_0^{\,k-1}\right]dx_k\end{aligned} \tag{25}$$

In Formulas (24) - (25), c is the normalized constant.

To calculate the vector of estimates $x'_k(a_{j,k})$, we must use Formula (10), as a result we obtain (26):

$$x'_k\left(a_{j,k}\right)=\int x_k\left(a_{j,k}\right)p[x_k\left(a_{j,k}\right)\mid z^k{}_0]dx_k \tag{26}$$

NUMERICAL METHODS FOR IMPLEMENTING ADAPTIVE NONLINEAR FILTERING ALGORITHMS

The main problems in the implementation of nonlinear filters is the integral (global) approximation of the conditional probability density and numerical integration (Bain & Crisan 2009; Berkes et al., 2011). There are various numerical methods for representing probability density: replacing a continuous density function by a set of discrete points, approximation using Hermite polynomials, spline functions, Fourier series, and the use of poly-Gaussian distributions. We focus on numerical methods that are convenient for implementation on a computer. These include the convolution of functions and the grid method.

Using Convolution Functions and Fast Fourier Transform

An analysis of Bayesian recurrence Formulas (4) - (6), (13) - (15) to calculate the posterior probability density shows that they are a convolution of two functions of the vector argument x (Churchland et al., 2010). In accordance with the convolution theorem (Churchland et al., 2010), the classical convolution of two functions *f (x)* and *g(x)* from the interval *(a, b)* is defined as (27):

$$\psi\left(y\right) = \int_{a}^{b} f\left(y - x\right) d\left(x\right) dx = \int_{a}^{b} g\left(y - x\right) f\left(x\right) dx \tag{27}$$

Function (27) is denoted by $f * g$.

For the numerical implementation of nonlinear filters based on Expression (27), two discrete calculation schemes for nonlinear convolution can be proposed for discrete measurements to obtain optimal estimates. For convenience, we consider them for one-dimensional convolution and the discrete Fourier transform.

The First Scheme

The convolution can be quickly executed based on the Borel theorem and a pair of Fourier transforms (Churchland et al., 2010). Let the Fourier transforms under consideration exist. Direct Fourier Transform (28):

$$c\left(\nu\right) = F\left|f\left(x\right)\right| \equiv \int_{-\infty}^{\infty} f\left(x\right) e^{-2\pi i \nu x} dx = \sqrt{2\pi} C\left(2\pi\nu\right) = 2\pi\Omega\left(\omega\right) \tag{28}$$

Note that the Fourier transform of the function f (x) is not only called c c(ν), but also C(ω) or Ω (ω) (Churchland et al. 2010) (29):

$$C\left(\omega\right) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f\left(x\right) e^{-2i\omega x} dx \equiv \frac{1}{\sqrt{2\pi}} c\left(\frac{\omega}{2\pi}\right) \tag{29}$$

$$\Omega\left(\omega\right) \equiv \int_{-\infty}^{\infty} f\left(x\right) e^{-i\omega x} dx \equiv c\left(\frac{\omega}{2\pi}\right) \tag{30}$$

where $\omega = 2\pi\nu$ is the circular frequency, ν - is the cyclic frequency.

The function *f(x)*, having the Fourier transform c(ν), is the inverse Fourier transform $F^{-1}| c(\nu)$ | of the function c(ν) (30):

$$f\left(x\right) = \int_{-\infty}^{\infty} c\left(\nu\right) e^{2\pi i \nu x} d\nu = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} C\left(\omega\right) e^{i\nu x} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Omega\left(\omega\right) e^{i\nu x} d\omega$$

Further, the Borel convolution theorem is used:

$$F[f_1(x)]\ F[f_2(x)] = F[f_1(x) * f_2(x)] \quad (31)$$

$$F[f_1(x) f_2(x)] = F[f_1(x)] * F[f_2(x)] \quad (32)$$

The dimension of the Fourier image is equal to the dimension *f(x)* times the dimension *x*. Integrals (28) - (30) exist (are converging to a finite value) only for the so-called "two-sided decaying processes". In numerical calculations, this contradiction is smoothed out, since in this case we are dealing only with processes of a limited duration, and the process itself in a given range must be set by its values in a limited number of points (Deneve et al., 2007). In this case, the integration is replaced by summation, and instead of calculating the integral (29), they are limited to calculating the sum (33):

$$c\left[\left(k-1\right)\cdot\Delta\nu\right] = \Delta x\cdot\sum_{m=1}^{n} x\left[\left(m-1\right)\cdot\Delta x\right]\cdot e^{-i2\pi\cdot\left(k-1\right)\cdot\left(m-1\right)\cdot\Delta\nu\cdot\Delta x} \quad (33)$$

Here, in comparison with the integral (29), the following changes are made:

- The continuous integral is approximately replaced by a limited sum of the areas of the rectangles, one of the sides of which is equal to the discrete variable Δx with which the process values are presented, and the second to the instantaneous value of the process at the corresponding time;
- The continuous parameter x is replaced by its discrete values $(m-1)\Delta x$, where m is the point number from the beginning of the process;
- Continuous values of the frequency ν are replaced by its discrete values $(k-1)\Delta\nu$, where k is the number of the frequency value, and the frequency discrete is $\Delta\nu = 1 / T$, where, T is the interval at which the process is specified;
- The differential dx is replaced by a limited increment of the parameter Δx. In solving practical problems, a finite number of samples N of the analog function is used, and in this case, a pair of Fourier transforms (19), (20) takes the form of the so-called discrete Fourier transform (DFT) (Doya et al., 2007) (34) (35):

$$c\left(k\right) = \sum_{m=0}^{n-1} f\left(m\right)\cdot e^{-i\left(\frac{2\pi}{n}\right)\cdot mk} \quad (34)$$

where:

k=0,1,…,n-1

$$f\left(m\right) = \frac{1}{n}\sum_{k=0}^{n-1} c\left(k\right)\cdot e^{i\left(\frac{2\pi}{n}\right)\cdot\left(m-1\right)\left(k-1\right)} \quad (35)$$

where:

m=0,1,…,n-1

Expression (34) defines the direct discrete Fourier transform, and Expression (35) defines the inverse discrete Fourier transform. The discrete Fourier transform algorithm provides a large amount of computation. Elimination of redundant multiplication operations leads to the so called fast Fourier transform (FFT) (Doya et al., 2007).

So, in accordance with the Borel theorem, a convolution can be performed quickly, providing for the following sequence of actions:

1. Using FFT, the calculation of the spectra *c(k)* and *G(k)* involved in the convolution of the functions *f(x)* and *g(x)*;
2. The calculation of the product of the spectra Z(k) = c(k)G(k);
3. Using FFT, the inverse DFT is calculated from Z(k) = c(k)G(k), which is the desired result of the convolution y(k).

The Second Scheme

It is based on the convolution operation of two periodic sequences *f(k)* and *g(k)* (Doya et al., 2007), which is expressed by the Formula (36) (37):

$$y\left(n\right) = \sum_{m=0}^{N} f\left(m\right) g\left(n-m\right), n = 0,1,\ldots,N-1 \tag{36}$$

or:

$$y\left(n\right) = \sum_{m=0}^{N} f\left(n-m\right) g\left(m\right), n = 0,1,\ldots,N-1 \tag{37}$$

Since the original functions *f(k)* and *g(k)* are not periodic, it is necessary to supplement the sequence of their samples with so many zero values so that when calculating the convolution, the values of the original functions would be taken only from the main period.

Let *f(k)* be an aperiodic sequence of length N_1, *g(k)* an aperiodic sequence of length N_2 samples. In this case, sequences of samples $f_1(k)$ and $g_1(k)$ are formed, each of the length $N_1 + N_2 - 1$ samples by including additional zero values (38) (39):

$$f_1\left(n\right) = \begin{cases} f\left(n\right), where\ n = 0,1,\ldots,N_1 - 1 \\ 0, where\ n = N_1,\ldots,N_1 + N_2 - 1 \end{cases} \tag{38}$$

$$g_1\left(n\right) = \begin{cases} g\left(n\right), where\ n = 0,1,\ldots,N_2 - 1 \\ 0,\ where\ n = N_2,\ldots,N_1 + N_2 - 1 \end{cases} \tag{39}$$

Moreover, the convolution of the sequences *f(k)* and *g(k)* is determined by *(N1 + N2-1)* - point convolution (40) (41):

$$y\left(n\right) = \sum_{m=0}^{N_1+N_2-1} f_1\left(m\right) g_1\left(n-m\right), where\ n = 0,1,\ldots,N_1 + N_2 - 2 \tag{40}$$

or:

$$y\left(n\right) = \sum_{m=0}^{N_1+N_2-1} f_1\left(n-m\right) g_1\left(m\right), where\ n = 0,1,\ldots,N_1 + N_2 - 2 \tag{41}$$

Computing such a convolution using FFT will require an $N_1 + N_2 - 2$ -point DFT. Despite the fact that the first method of convolution calculation involves triple DFT calculation, it is more economical than direct convolution calculation using Formulas (36) and (37). Similarly, to expressions for one-dimensional convolution and FFT, expressions for multidimensional transformations can be written.

UAV CONTROLLED DYNAMICS WITH ROTARY ROTORS

The UAV state vector with rotary rotors includes the position of the center of mass, speed, the quaternion of the orientation of the UAV body, and angular velocity (42):

$$x = (r\ v\ q\ \Omega)^{\mathrm{T}} \tag{42}$$

The movement of the apparatus is described by Equations (43) (44) (45) (46):

$$r' = \nu \tag{43}$$

$$\nu' = q \circ \left(\sum_{i=1}^{4} qBR_i \circ \left(-1\right)^{i+1} k\hat{I}_i \left|\hat{I}_i\right| \circ q'BR_i \right) \circ q' \tag{44}$$

$$q' = \frac{1}{2} q \circ \Omega \tag{45}$$

$$\Omega' = J_B^{-1} \begin{bmatrix} -\Omega \times J_B \Omega + \sum_{i=1}^{4} r_i^B \times \left(-1\right)^{i+1} kp_i \left|p_i\right| e_{zi}^{I} - \sum_{i=1}^{4} qBR_i \circ bp_i \left|p_i\right| e_{ri}^{Ri} \\ \circ q'BR_i + \sum_{i=1}^{4} qBR_i \circ \left(J_{Ri} p_i^{Ri} + p_i^{Ri} + JR_i \omega_i^{Ri}\right) \circ q'BR_i \end{bmatrix} \tag{46}$$

where:

- M - is the total mass of the apparatus;
- g – is the acceleration of gravity;
- e^{I}_{zi} – unit vector along the axis of symmetry of the i[th] rotor;
- k and b – is aerodynamic propeller coefficients;
- ω'_i – is the rotation speed of the i-th propeller;
- JB — is the inertia tensor of the body in the main axes of the body;
- r^{B}_{i} – is the radius-vector, drawn from the center of mass of the i-th UAV towards the rotor;
- $e^{B}r_i$ – is the unit vector of r^{B}_{i};
- qBR_i – is a quaternion that determines the orientation of the i-th rotor with respect to the body of the apparatus;
- JRi – is the inertia tensor of the i-th rotor with a propeller written in its own main axes;
- $\omega\, i^{Ri}$ – is the total angular velocity of the i-th rotor with a propeller recorded in its own main axes.

To ensure independent control by position and orientation, regulators of the form (47) are used:

$$r''_d\left(t\right) = r''^{0}\left(t\right) + K_{r1}\left(r'^{0}\left(t\right) - r'\left(t\right)\right) + K_{r2}\delta r$$

$$\Omega'_d\left(t\right) = \Omega'^{0}\left(t\right) + K_{\Omega 1}\left(\Omega^{0}\left(t\right) - \Omega\left(t\right)\right) + K_{\Omega 2}\delta q \tag{47}$$

where:

- δq – is a vector part of quaternion mismatch;
- δr – is a mismatch of the current position;
- $r''0$, $'0$, $\Omega''0$, Ω'_0 – they are: target acceleration, velocity, angular acceleration and angular velocity;
- Kr_1, Kr_2, $K\Omega_1$, $K\Omega_2$ – is a diagonal matrixes of coefficients.

Maintaining the values of the revolutions ω_i and the orientation $q\mathrm{BR}_i$ of the engines satisfying the outputs of the regulators (10) according to the model (12) ensures the achievement of the target motion parameters.

In this paper, we do not consider the solution of the problem of the inverse dynamics of the UAV and the synthesis of the control loop, but it is believed that the method for determining the control parameters is known.

Experimental Part

To determine the motion path, the UAV model (9) is numerically integrated, while the control parameters ω'_i, qBR_i are some functions of the current state and the target path. The integration of the model is carried out using the MATLAB package of the 4th order Runge – Kutt method with a step of $\delta = 10^{-2}$ s. The measurement vector is composed in a manner similar to the state vector (42) (48):

$$z = (r\ v\ q\ \Omega) \tag{48}$$

The measurements are modeled by adding to the results of integration of the equations of motion (9) white Gaussian noise, the parameters of which are selected in such a way as to correspond to the parameters of existing devices (Table 1).

Table 1. The standard deviations of the noise measurements of horizontal position components, vertical position components, velocity components, pitch angles, roll, yaw and angular velocity components

σ_{rx}	1 m
σ_{rz}	2 m
σ_{v}	0.5 m/s
σ_{α}	$0.5°$
σ_{β}	$0.5°$
σ_{γ}	$1.5°$
σ_{Ω}	$0.6°$/s

Measurement Noise Parameters

As a model of aircraft dynamics in nonlinear filtering algorithms, a simplified model of UAV motion is used, which does not take into account the inertia of rotary rotors with propellers. Then the expression for angular acceleration (43) can be written as (49):

$$\Omega' = J_B^{-1}\left[-\Omega \times J_B\Omega + \sum r_i^B \times \underset{i=1}{\overset{4}{\left(-1\right)^{i+1}}} k\hat{I}_i \left|\hat{I}_i\right| e_{zi}^I - \sum qBR_i \circ b\hat{I}_i \underset{i=1}{\overset{4}{\left|\hat{I}_i\right|}} e_{ri}^{Ri} \circ q' BR_i\right] \tag{49}$$

This simplification is often resorted to in practice, since when designing UAVs with rotary rotors, even approximate identification of the main parameters of the dynamics of the executive organs of the control system is a laborious process and requires special measurements. Thus, the expressions associated with the dynamics of the executive bodies of the control system form the noise vector w of the system from the equation of the stochastic continuous system (1).

As the parameter determining the filter performance, the standard deviation of the components of the state estimation vector from the results of integration of the equations of motion was selected. To exclude the influence of the target trajectory on the experimental results, the standard deviation is averaged over 100 spans over the characteristic trajectories.

The duration of each flight mission is 90s, during this time the UAV moves along a curved path in space, and its body unfolds according to the target orientation parameters at the current moment. The maximum permissible speed of the UAV is limited to 5 m/s, and the angular speed is 3m/s.

All three filters use the same measurements and work on the same set of paths.

The performance of each of the filtering algorithms is studied for various observation intervals that are multiples of the system integration step, that is, measurements and an assessment of the state are performed at intervals $TN = N\delta$.

The matrices Q covariance of the system noise and R covariance of the measurement noise used in all three algorithms were chosen taking into account the knowledge about the parameters of the measurement noise and are adjusted so that the algorithms show high performance for the measurement interval $T1 = \delta$ (50):

$$Q = 10^{-6} diag(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 50, 50, 50)$$

$$R = diag(1, 1, 1, 0.5, 0.5, 0.5, 0.003, 0.003, 0.003, 0.006, 0.003, 0.003, 0.003) \tag{50}$$

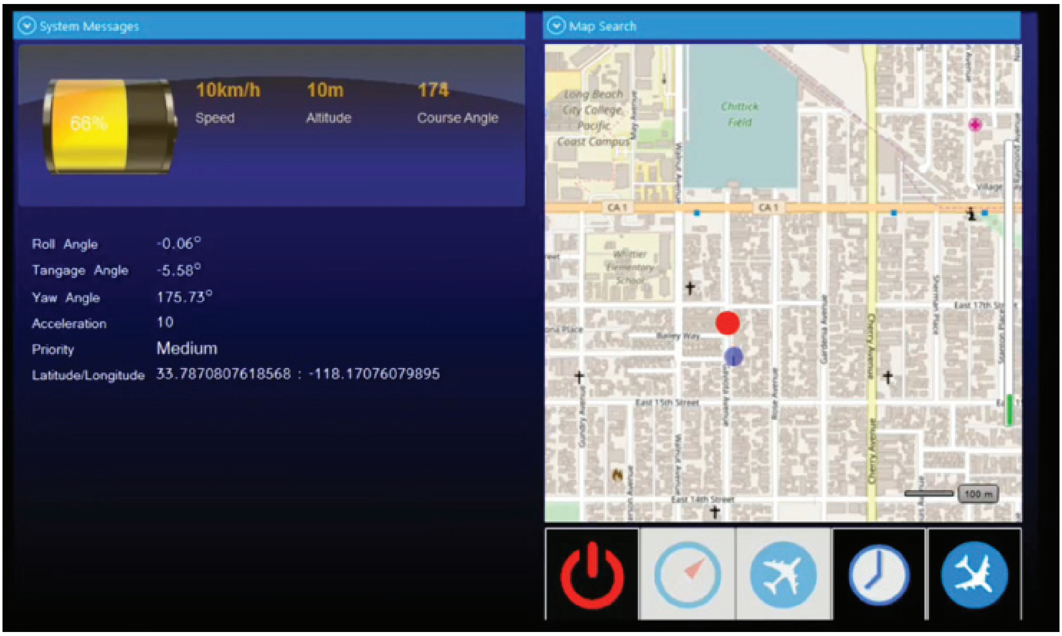
The experiments showed that the quality of the estimation of the state vector correlates with the interval of operation of the filters — a decrease in the measurement frequency leads to a deterioration of the estimation parameters. The extended Kalman filter shows the lowest performance at large measurement intervals: even for $N > 2$, state estimation errors in some cases are beyond the acceptable range, and for $N > 5$, the position estimate becomes incorrect.

The performance of the Kalman sigma-point and cubature filters also decreases with increasing N, but not so noticeably, while the failures in the operation of these algorithms are much smaller. A comparative analysis of the UKF and CKF results showed that the CKF algorithm makes a more accurate state assessment in most of the cases under consideration and is more resistant to increasing the measurement interval.

The best result was shown by an algorithm based on Bayesian filters.

As a practical implementation, the described above approach was applied in the AI of the Drone Management System developed by Rinat Galiautdinov, where the additional details are described at http://airhighway.online (Figure 1).

Figure 1. The demo of the flight



CONCLUSION

Here the author considered the possibility of applying modern IT technologies to implement information processing algorithms in UAV motion control system. Filtration of coordinates and motion parameters of objects under a priori uncertainty is carried out using nonlinear adaptive filters: Kalman and Bayesian filters. The author considers numerical methods for digital implementation of nonlinear filters based on the convolution of functions, the possibilities of neural networks and fuzzy logic for solving the problems of tracking UAV objects (or missiles), the math model of dynamics, the features of the practical implementation of state estimation algorithms in the frame of added additional degrees of freedom. The considered algorithms are oriented on solving the problems in real time using parallel and cloud computing.

A comparative analysis of the performance of the extended Kalman filter and the Bayesian filter for various measurement intervals was carried out (Galiautdinov, R., 2019). As a quality criterion, the standard deviation of the state estimate averaged over 100 experiments from the trajectory of the model for each of the components of the state vector is used. It is shown that extended Kalman filter has significant sensitivity to measurement intervals, its increase leads to a rapid increase in the error of state estimation and, subsequently, to completely incorrect estimation of some components of the state vector.

The extended Kalman filter will be briefly considered here, mostly focusing on the Bayesian approach to the synthesis of discrete nonlinear filtering algorithms, which allows us to completely solve the problem of both linear and nonlinear filtering in discrete time.

The Bayesian approach was applied and this allowed to obtain a general solution to the nonlinear filtering problem in discrete time. Based on the Bayesian methodology, formulas are obtained for the recursive calculation of the conditional posterior probability density of the extended state vector, which includes the state vector of the dynamic system and the vector of accompanying parameters, which takes into account the uncertainty in the statistical characteristics of the dynamic process and real perturbations, the inaccuracy in setting the mathematical parameters models. For the numerical implementation of algorithms using the grid method and convolution of functions, it is proposed to use Kalman filter estimates or estimates generated by a neural network as a reference solution. The problem of estimating the motion parameters of maneuvering objects based on neural networks and the use of knowledge bases and fuzzy logic to increase the efficiency of filtering algorithms are considered. With the practical implementation of parallel and pipelined computing, neurocomputers, it becomes possible to increase the accuracy and reliability of evaluations in real-time implementation of optimal nonlinear filtering algorithms. They provide developers of devices and systems with greater accuracy in estimating the parameters of dynamic processes. The numerical methods of Bayesian adaptive filtering algorithms considered in the paper allow almost unlimited parallelization and pipelining of computational processes. For the numerical implementation of nonlinear filtering methods, effective methods of parallel programming can be applied here. In addition, to increase the speed of information processing during programming, it is advisable to use dynamic data structures such as lists, trees, table functions, hashing, high-speed sorting algorithms and an arithmetic machine.

REFERENCES

Bain, A., & Crisan, D. (2009). *Fundamentals of Stochastic Filtering*. Springer. doi:10.1007/978-0-387-76896-0

Beck, J. M., & Pouget, A. (2007). Exact inferences in a neural implementation of a hidden Markov model. *Neural Computation*, *19*(5), 1344–1361. doi:10.1162/neco.2007.19.5.1344 PMID:17381269

Bergemann, K., & Reich, S. (2012). An ensemble Kalman-Bucy filter for continuous data assimilation. *Meteorologische Zeitschrift (Berlin)*, *21*(3), 213–219. doi:10.1127/0941-2948/2012/0307

Berkes, P., Orban, G., Lengyel, M., & Fiser, J. (2011). Spontaneous Cortical Activity Reveals Hallmarks of an Optimal Internal Model of the Environment. *Science*, *331*(6013), 83–87. doi:10.1126/science.1195870 PMID:21212356

Beskos, A., Crisan, D., & Jasra, A. (2014). On the Stability of Sequential Monte Carlo Methods in High Dimensions. *PLoS Computational Biology*, *46*.

Beskos, A., Crisan, D., Jasra, A., Kamatani, K., & Zhou, Y. (2014). *A Stable Particle Filter in High-Dimensions*. ArXiv:1412.350

Cappe, O. (2011). Online EM Algorithm for Hidden Markov Models. *Journal of Computational and Graphical Statistics*, *20*(3), 1–20. doi:10.1198/jcgs.2011.09109

Churchland, A. K., Kiani, R., Chaudhuri, R., Wang, X.-J., Pouget, A., & Shadlen, M. N. (2011). Variance as a signature of neural computations during decision making. *Neuron*, *69*(4), 818–831. doi:10.1016/j.neuron.2010.12.037 PMID:21338889

Churchland, M. M., Yu, B. M., Cunningham, J. P., Sugrue, L. P., Cohen, M. R., Corrado, G. S., Newsome, W. T., Clark, A. M., Hosseini, P., Scott, B. B., Bradley, D. C., Smith, M. A., Kohn, A., Movshon, J. A., Armstrong, K. M., Moore, T., Chang, S. W., Snyder, L. H., Lisberger, S. G., & Shenoy, K. V. et al. (2010). Stimulus onset quenches neural variability: A widespread cortical phenomenon. *Nature Neuroscience*, *13*(3), 369–378. doi:10.1038/nn.2501 PMID:20173745

Crisan, D., & Xiong, J. (2010). Approximate McKean–Vlasov representations for a class of SPDEs. *Stochastics*, *82*(1), 53–68. doi:10.1080/17442500902723575

Deneve, S., Duhamel, J.-R., & Pouget, A. (2007). Optimal sensorimotor integration in recurrent cortical networks: A neural implementation of Kalman filters. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, *27*(21), 5744–5756. doi:10.1523/JNEUROSCI.3985-06.2007 PMID:17522318

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*(3), 197–208. doi:10.1023/A:1008935410038

Doucet, A., & Johansen, A. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. Handbook of Nonlinear Filtering, 4–6.

Doya, K. (2007). *Bayesian Brain: Probabilistic Approaches to Neural Coding*. The MIT Press.

Ernst, M. O., & Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, *415*(6870), 429–433. doi:10.1038/415429a PMID:11807554

Fiser, A., Mahringer, D., Oyibo, H. K., Petersen, A. V., Leinweber, M., & Keller, G. B. (2016). Experience-dependent spatial expectations in insect visual cortex. *Nature Neuroscience*, *19*(12), 1658–1664. doi:10.1038/nn.4385 PMID:27618309

Fiser, J., Berkes, P., Orban, G., & Lengyel, M. (2010). Statistically optimal perception and learning: From behavior to neural representations. *Trends in Cognitive Sciences*, *14*(3), 119–130. doi:10.1016/j.tics.2010.01.003 PMID:20153683

Galiautdinov, R. (2019). Advanced method of planning the trajectory of swarm's drones possessing Artificial Intelligence. *International Journal of Scientific Research (Ahmedabad, India)*. Advance online publication. doi:10.21275/ART20203362

Galiautdinov, R. (2020). Brain machine interface: The accurate interpretation of neurotransmitters' signals targeting the muscles. *International Journal of Applied Research in Bioinformatics*, *0102*. Advance online publication. doi:10.4018/IJARB.2020

Galiautdinov, R., & Mkrttchian, V. (2019A). Math model of neuron and nervous system research, based on AI constructor creating virtual neural circuits: Theoretical and Methodological Aspects. In V. Mkrttchian, E. Aleshina, & L. Gamidullaeva (Eds.), *Avatar-Based Control, Estimation, Communications, and Development of Neuron Multi-Functional Technology Platforms* (pp. 320–344). IGI Global. doi:10.4018/978-1-7998-1581-5.ch015

Galiautdinov, R., & Mkrttchian, V. (2019B). Brain machine interface – for Avatar Control & Estimation in Educational purposes Based on Neural AI plugs: Theoretical and Methodological Aspects. In V. Mkrttchian, E. Aleshina, & L. Gamidullaeva (Eds.), *Avatar-Based Control, Estimation, Communications, and Development of Neuron Multi-Functional Technology Platforms* (pp. 345–360). IGI Global. doi:10.4018/978-1-7998-1581-5.ch016

Greaves-Tunnell, A. (2015). *An optimization perspective on approximate neural filtering*. Master thesis.

Hennequin, G., Aitchison, L., & Lengyel, M. (2014). Fast Sampling-Based Inference in Balanced Neuronal Networks. *Advances in Neural Information Processing Systems*.

Huang, Y. & Rao, R. (2014),.Neurons as Monte Carlo Samplers: Bayesian Inference and Learning in Spiking Networks. *Neural Information Processing Systems*, 1–9.

Huang, Y., & Rao, R. P. N. (2016). Bayesian Inference and Online Learning in Poisson Neuronal Networks. *Neural Computation*, *28*(8), 1503–1526. doi:10.1162/NECO_a_00851 PMID:27348304

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., & Chopin, N. (2015). On Particle Methods for Parameter Estimation in State-Space Models. *Statistical Science*, *30*(3), 328–351. doi:10.1214/14-STS511

Kappel, D., Habenschuss, S., Legenstein, R., & Maass, W. (2015). Network Plasticity as Bayesian Inference. *PLoS Computational Biology*, *11*(11), 1–31. doi:10.1371/journal.pcbi.1004485 PMID:26545099

Knill, D. C., & Pouget, A. (2004). The Bayesian brain: The role of uncertainty in neural coding and computation. *Trends in Neurosciences*, *27*(12), 712–719. doi:10.1016/j.tins.2004.10.007 PMID:15541511

Kording, K., Tenenbaum, J. B., & Shadmehr, R. (2007). The dynamics of memory as a consequence of optimal adaptation to a changing body. *Nature Neuroscience*, *10*(6), 779–786. doi:10.1038/nn1901 PMID:17496891

Kording, K. P., & Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. *Nature*, *427*(6971), 244–247. doi:10.1038/nature02169 PMID:14724638

Legenstein, R., & Maass, W. (2014). Ensembles of Spiking Neurons with Noise Support Optimal Probabilistic Inference in a Dynamically Changing Environment. *PLoS Computational Biology*, *10*(10), e1003859. doi:10.1371/journal.pcbi.1003859 PMID:25340749

Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature Neuroscience*, *9*(11), 1432–1438. doi:10.1038/nn1790 PMID:17057707

MacKay, D. J. (2005). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Makin, J. G., Dichter, B. K., & Sabes, P. N. (2015). Learning to Estimate Dynamical State with Probabilistic Population Codes. *PLoS Computational Biology*, *11*(11), 1–28. doi:10.1371/journal.pcbi.1004554 PMID:26540152

Mkrttchian, V., Gamidullaeva, L., & Galiautdinov, R. (2019). Design of Nano-scale Electrodes and Development of Avatar-Based Control System for Energy-Efficient Power Engineering: Application of an Internet of Things and People (IOTAP) Research Center. *International Journal of Applied Nanotechnology Research*, *4*(1), 41–48. Advance online publication. doi:10.4018/IJANR.2019010104

Mongillo, G., & Deneve, S. (2008). Online learning with hidden markov models. *Neural Computation*, *20*(7), 1706–1716. doi:10.1162/neco.2008.10-06-351 PMID:18254694

Moreno-Bote, R., Knill, D. C., & Pouget, A. (2011). Bayesian sampling in visual perception. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(30), 12491–12496. doi:10.1073/pnas.1101430108 PMID:21742982

Movellan, J. R., Mineiro, P., & Williams, R. J. (2002). A Monte Carlo EM approach for partially observable diffusion processes: Theory and applications to neural networks. *Neural Computation*, *14*(7), 1507–1544. doi:10.1162/08997660260028593 PMID:12079544

Orban, G., Berkes, P., Fiser, J., & Lengyel, M. (2016). Neural Variability and Sampling-Based Probabilistic Representations in the Visual Cortex. *Neuron*, *92*(2), 530–543. doi:10.1016/j.neuron.2016.09.038 PMID:27764674

Pecevski, D., Buesing, L., & Maass, W. (2011). Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons. *PLoS Computational Biology*, *7*(12), 7. doi:10.1371/journal.pcbi.1002294 PMID:22219717

Poyiadjis, G., Doucet, A., & Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, *98*(1), 65–80. doi:10.1093/biomet/asq062

Rebeschini, P., & Van Handel, R. (2015). Can local particle filters beat the curse of dimensionality? *Annals of Applied Probability*, *25*(5), 2809–2866. doi:10.1214/14-AAP1061

Sokoloski, S. (2015). *Implementing a Bayes Filter in a Neural Circuit: The Case of Unknown Stimulus Dynamics*. ArXiv, ArXiv:1512.07839

Surace, S. C., & Pfister, J.-P. (2016). *Online Maximum Likelihood Estimation of the Parameters of Partially Observed Diffusion Processes*. ArXiv:1611.00170

Welling, M., & Teh, Y. (2011), Bayesian Learning via Stochastic Gradient Langevin Dynamics. *Proceedings of the 28th International Conference on Machine Learning*.

Wilson, R. C., & Finkel, L. (2009). A neural implementation of the Kalman filter. *Advances in Neural Information Processing Systems*, 22.

Yang, T., Laugesen, R. S., Mehta, P. G., & Meyn, S. P. (2016). Multivariable feedback particle filter. *Automatica*, *71*, 10–23. doi:10.1016/j.automatica.2016.04.019

Yang, T., Mehta, P. G., & Meyn, S. P. (2013). Feedback particle filter. *IEEE Transactions on Automatic Control*, *58*(10), 2465–2480. doi:10.1109/TAC.2013.2258825

Rinat Galiautdinov is a Principal Software Developer and Software Architect having the expertise in Information Technology and Computer Science. Mr. Galiautdinov is also an expert in Banking/Financial industry as well as in Neurobiological sphere. Mr. Rinat Galiautdinov works on the number of highly important researches (including such the spheres as: Drones, Financial Systems and Private Money, Brain interface and neuro-biology) as an independent researcher and projects holder and developer.

International Journal of Software Science and Computational Intelligence
Volume 12 • Issue 4 • October-December 2020

Experience-Based Approach for Cognitive Vehicle Research

Hironori Hiraishi, Ashikaga University, Japan
https://orcid.org/0000-0002-0447-9114

ABSTRACT

This study discusses an experience-based approach to cognitive vehicle research. The author recruited six subjects who have different individual driving experiences and skills. This paper made them drive their own cars on the same test course. This research collected the drivers' mental status and operations as the driving data using a simple brainwave sensor and the acceleration sensor of a smart phone. The authors analyzed the data and developed models using the cognitive qualitative analysis and modeling tool (QCAM) that the authors have developed. The authors experimentally verified how different the results of the analysis and modeling were. The experiment results identified a parameter that indicated the degree of experience and skill of each driver. Moreover, the models derived the rules that explain the experience of each driver. The experience-based approach enables an understanding of unconscious operations and situated cognition based on the drivers' experiences, and it also allows novice or senior drivers to record their driving experiences and watch them later to maintain their driving skills.

KEYWORDS

Acceleration Data, Brainwave Sensor, Cognitive Model, Cognitive Vehicle, Cognitive-Experiential Theory, Qualitative Induction

1. INTRODUCTION

In the area of cognitive vehicle research, cognitive computing has been used to analyze and model driving operations and situations for driving supports and ensuring safety of self-driving vehicles. A vehicle simulator had been developed to simulate human driving behavior (Marvin, Jonathan, Christine & Craig, 2017). Further, some experiments in actual driving that use biological data have been reported. A highly accurate cognitive model that expresses mental workloads of drivers was successfully constructed using an approach based on eye movement analysis (Sega, Iwasaki, Hiraishi & Mizoguchi, 2011). It was reported in (Yoshizama, Nishiyama, Iwasaki & Mizoguchi, 2017) that analyzing eye movement is effective in determining the driver's cognitive status such as attention and distraction. Additionally, a pressure sensor placed on a seat can be used to determine drivers' or passengers' fatigue level (Hiraishi, 2018). Similarly, drivers' face can also be used to identify the extent of fatigue (Zuojin, Jun, Liukui, Ying & Jinliang, 2017). A simple brainwave sensor is now available to measure drivers' attention and relaxation levels (Hiraishi, 2019). A qualitative approach is

DOI: 10.4018/IJSSCI.2020100104

advantageous in analyzing such biological data (Sega, Iwasaki, Hiraishi & Mizoguchi, 2011; Hiraishi, 2017). Because qualitative analysis does not deal with the quantity of each data and only focuses on the qualitative change, such as whether it is an increase, a decrease, or a constant, it is effective for factors that have individual differences such as mental status and cognitive situation. Therefore, we have developed the Qualitative Cognitive Analysis and Modeling tool (QCAM) (Hiraishi, 2019), which performs a qualitative analysis and a cognitive modeling of the data obtained by a biological sensor.

QCAM can analyze data from a driving recorder, simple brainwave sensor, acceleration sensor of a smart phone, etc. In addition, it can make models that represent the relationship between the driver's operation and his/her mental status such as attention or relaxation. Moreover, the scene-based analysis function of QCAM can make each model cognizant of the road type such as whether it is a narrow road or a two-lane road. For example, we can create a model that shows an increase in the attention level if the driver accelerates on a narrow road or an increase in the relaxation level in the case of a wide road. Here, we expect the output of such a model to differ from driver to driver depending on his/her experiences and skills. In the case of novice drivers, they tend to feel tense when it comes to driving itself. Therefore, QCAM may generate a model indicating that the driver's attention increases when driving not only on a narrow road but also on a wide road. Thus, even with the same driving operations and road situations, drivers feel different mental workloads individually according to their experiences and skills.

In this paper, we discuss an experience-based approach for analyzing and modeling cognitive driving data. We recruited six subjects who have different individual driving experiences and skills. We made them drive their own cars on the same test course. We collected the drivers' mental status and operations as the driving data using a simple brainwave sensor and the acceleration sensor of a smart phone. We analyzed the data and made models by using QCAM and then experimentally verified how different the results of the analysis and modeling were.

The rest of the paper is organized as follows. Section 2 provides a simple introduction to QCAM. Section 3 explains our experiment conditions and discusses the test course, the collected driving data, and the subjects' profiles. Section 4 shows the results of the analysis and modeling done by QCAM. Section 5 discusses the experience-based approach. Finally, we conclude the paper in Section 6.

2. THE COGNITVE QUALITATIVE ANALYSIS AND MODELING TOOL

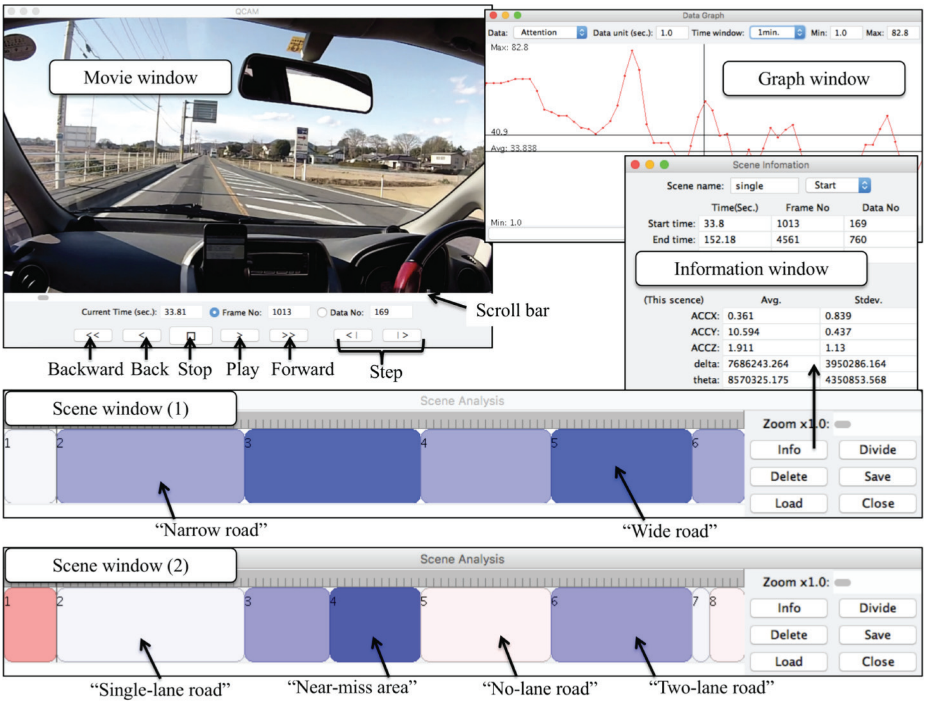
The input data to QCAM are movie data and time sequence sensor data. After QCAM starts, the movie window appears, as shown in Figure 1 (upper left). We can play the movie by using the seven buttons at the bottom of the window, and we can seek through the movie by using the scroll bars. We can check the current time, frame number of the movie, and data number of the sensor data on the movie window.

The data of each sensor are shown on a graph window (Figure 1, upper right). This graph window is synchronized with the movie play. It is scrolled automatically so that its center becomes the current time of the movie. Moreover, QCAM allows us to view several graphs simultaneously. Therefore, we can check the change in the data of each sensor visually while watching the movie.

The most important feature of QCAM is the scene analysis function. The belt-shaped scene window of QCAM is shown in Figure 1 (lower part). In the first stage, there is only one belt. QCAM treats one belt as one scene. Therefore, in the beginning, it treats all data as one scene. If we press the "Divide" button after moving the current position, we can divide the scene at that position. If we select a scene and press the "Delete" button, we can delete the scene. Performing these operations repeatedly will enable us to separate scenes with a narrow road and a wide road like in scene window (1) or to separate scenes with a no-lane road, single-lane road, two-lane road, etc., like in scene window (2).

The "Zoom" bar allows us to zoom the scene belt up to 10×. Thus, we can divide scenes very specifically. A scene with the same color means a scene with the same name. We can save a sequence of scenes with any filename by using the "Save" button and load a file by using the "Load" button.

Figure 1. Screenshot of QCAM



If we press the "Info" button while a scene is selected, QCAM shows an information window (Figure 1, right center). We can name the scene and confirm the time information, average (Avg.), and standard deviation (Stdev.) of each data item in the window.

Finally, QCAM provides functions for performing a qualitative analysis and modeling. We can analyze the qualitative change in the data of each sensor, such as whether it is an increase, a decrease, or a constant. QCAM generates the model as a qualitative tree (Šuc, Vladusic & Bratko, 2004); we can extract rules that express the characteristics of each scene. Moreover, we can design a qualitative model in the Qualitative Simulation (QSIM) description format, and this model can be verified using the QSIM tool, where we can check the model's state transitions.

In some cognitive-modeling approaches, the model is designed first and then its correctness is verified by applying some data (Sega, Iwasaki, Hiraishi & Mizoguchi, 2011). In contrast, QCAMs uses a data-driven approach; i.e., it first visualizes and analyzes real-time data and then designs the model on the basis of the analysis results.

3. EXPERIMENT SETTING

This section explains our experiment setting, such as the test course, driving data, and profiles of the six subjects.

3.1. Test Course

Figure 2 shows a picture of our test course, which is in Gunma Prefecture in Japan. This course contains different types of roads, such as no-lane roads, single-lane roads, and two-lane roads. Notably, this course contains a near-miss incident area, which is registered on the near-miss-incident map of Gunma

Figure 2. Picture of our test course

Prefecture. This area is a two-lane downhill road. Therefore, car speed becomes relatively fast. There is a junction merged from the left lane and an intersection that has a traffic signal after the junction.

Each subject drove along the road indicated by the red arrows from the start point to the goal point. The distance is approximately 4.9 km, and it takes about 10 minutes to traverse it. Table 1 summarizes the features of the test course. The course consists of a narrow road (2.5 km) and a wide road (2.4 km) basically. The narrow road contains a no-lane road (1.0 km) and a single-lane road (1.5 km) In contrast, the wide road contains only a two-lane road, although it includes the near-miss incident area (0.6 km).

Table 1. Features of the test course

Road Type	Distance (km)	Road Infrastructure	Distance (km)
Narrow road	2.5	No-lane road	1.0
		Single-lane road	1.5
Wide road	2.4	Two-lane road	1.8
		Near-miss incident area	0.6

3.2. Analysis Data

Figure 3 shows the different types of equipment used for collecting the driving data. The driving and road situations were recorded with a small camera (Driftinovation.com HD Ghost), shown in the upper right of Figure 3. Acceleration data were collected using a smart phone (Google Nexus 5X) mounted on the car dashboard, as shown on the left side of Figure 2. For the smart phone, the positive directions are backward for the Z-axis, rightward for the X-axis, and upward for the Y-axis. Thus, when a person is driving, acceleration produces a negative Z value, whereas slowing down produces a positive Z value. Turning left makes X negative and turning right makes it positive. The acceleration data can be obtained every 200 ms.

Figure 3. Types of equipment used for collecting the driving data

We measured the driver's brainwave data by using a simple brainwave sensor (B3 Band, B-Bridge International, Inc.), shown in the lower right of Figure 3. The data can be obtained every second. With the brainwave sensor, we can obtain the following eight commonly recognized types of brainwaves: delta, theta, low alpha, high alpha, low beta, high beta, low gamma, and high gamma. This sensor uses the brainwave measurement module by NeuroSky, Inc., and measures the "attention" and "meditation" levels using a proprietary algorithm called eSence. "Attention" and "meditation" are related to the attention and relaxation levels, respectively. The levels range from 0 (lowest) to 100 (highest). The algorithm details for calculating each value are not publicly available. However, they are defined in such a way that "attention" has more emphasis on the beta wave and "meditation" has more emphasis on the alpha wave.

Table 2 shows the first part of the input data for QCAM. The data names are on the first row, and the data are located below that. The first three columns indicate the acceleration data. The eight types of brainwaves are on the 4th to the 11th column. The 12th and 13th columns are the attention and meditation levels, respectively. The frequency of the acceleration sensor is 5 Hz and that of the brainwave sensor is 1 Hz. Therefore, we converted the brainwave data to 5 Hz. The interval value kept it linear.

Table 2. Sensor data format

ACCX	ACCY	ACCZ	Delta	…	Attention	Meditation
0.2011	10.5668	2.0549	6,256,898		44	20
0.1772	10.5524	2.0980	5,169,307		46	18.8
…						

We observed the driving and road situations from the movie data; driving operations, e.g., acceleration, braking, and steering, were used as the acceleration data. Furthermore, we observed the attention and relaxation levels from the brainwave sensor and analyzed how the operations, situations, and activities affected the driver's cognitive status and process.

3.3. Profile of Each Subject

We recruited six subjects (the author and five students), who have different individual driving experiences and skills.

The characteristics of each subject are as follows:

Subject A:
- Age and sex:22-year-old male student
- Driving career:1 month
- Driving frequency: seldom
- Experience: none
- Near-miss incident area:unknown

Subject B:
- Age and sex:22-year-old male student
- Driving career:4 years
- Driving frequency: only drives to attend school
- Experience: wide road
- Near-miss incident area:unknown

Subject C:
- Age and sex:22-year-old male student
- Driving career:4 years
- Driving frequency: everyday
- Experience: wide road
- Near-miss incident area:unknown

Subject D:
- Age and sex:25-year-old male student
- Driving career:7 years
- Driving frequency: everyday
- Experience: narrow road
- Near-miss incident area:unknown

Subject E:
- Age and sex:47-year-old male
- Driving career:28 years
- Driving frequency: everyday
- Experience: various types of road
- Near-miss incident area:known

Subject F:
- Age and sex:22-year-old male student
- Driving career:4 months
- Driving frequency: sometimes
- Experience: narrow road
- Near-miss incident area:known

Subject A was quite a beginner driver, who has just obtained his driver's license and does not have his own car. Therefore, he does not usually drive at all. Subject B uses his car only for attending school. He usually comes to school through a wide road. Therefore, he is used to driving on a wide road. Subjects C and D drive every day. Subject C drives to school through a wide road, whereas subject D does it through a narrow road. Subject E is the author and drives every day. He is quite used to driving on both a narrow road and a wide road.

Subject F is experienced in driving, but rarely drove his car during our experiment. After that, he went to school by his car. Therefore, a second experiment was performed with subject F to determine the change in his experience.

We did not tell the subjects about the near-miss incident area, except for subjects E and F. The subjects drove the test course with their own cars except for subject A.

4. EXPERIMENTAL RESULTS

This section shows the result of the comparison of the mental workloads of the subjects and also the models of a qualitative tree for each subject.

4.1. Comparison of Mental Workloads

Table 3 shows the experimental results of subjects A to E. We made two types of scene: narrow road (narrow) and wide road (wide), like those shown in scene window (1) in Figure 1. “Meditation” means the average of the relaxation level, and “attention” means the average of the attention level in each scene. “M-A” is the difference between these two values. A bigger value of this parameter indicates that the driver was more relaxed. On the contrary, a smaller value means that the driver felt more tense in a scene. “Difference” is the absolute difference between the “M-A” values of two scenes, and “average” is the average of two “M-A” values.

Table 3. Experimental results of subjects A-E

Subject	Scene	Meditation	Attention	M-A	Difference	Average
A	Narrow	59.1	62.2	-3.1	6.7	0.3
	Wide	57.1	53.5	3.6		
B	Narrow	58.2	58.5	-0.3	13.9	6.7
	Wide	64.3	50.7	13.7		
C	Narrow	62.5	56.8	5.8	5.0	8.3
	Wide	69.1	58.4	10.8		
D	Narrow	57.2	44.3	13.0	4.3	10.9
	Wide	59.1	50.4	8.7		
E	Narrow	49.6	27.3	22.3	6.2	19.2
	Wide	54.6	38.6	16.1		

First, we focus on subjects B, C, and D. We can see that the “M-A” value of the scene that each subject is used to is bigger. Because subject B comes to school through a wide road, the value of “wide” became bigger. As for subject C, this value became bigger, as well as for B. Subject D usually drives on a narrow road. Therefore, the value of “narrow” became bigger than that of “wide”. Thus, we can consider that “M-A” indicates the level of experience or skills of the driver for each road type.

In the case of subject B, the value of “difference” was extremely bigger than those of other subjects. He does not usually drive very much, and he just drives on a wide road when attending school. Because he is quite used to driving on a wide road, the “M-A” value of “wide” became big. Moreover, the “M-A” value of “narrow” became extremely small because he has little experience driving on a narrow road, which also explains why the value of “difference” became extremely bigger.

Subject A is quite a beginner driver, who has just obtained his driver's license. Because he is not used to driving, we can assume that he is not used to driving on a narrow road or on a wide road either. However, the "M-A" value of "wide" was bigger than that of "narrow". Therefore, we can assume that he felt it was easier to drive on a wide road than to drive on a narrow road.

Because subject E is an experienced driver, he is used to driving on both a narrow road and a wide road. However, the "M-A" value of "wide" was smaller than that of "narrow". This means that he paid more attention to the wide road. We can attribute this to his prior knowledge of the near-miss incident area. Because of this knowledge, his attention level increased when driving on the wide road.

Finally, we can see that the value of "average" became bigger in the order from subjects A to E. We can regard this value as a parameter that indicates the level of experience or skill of a driver.

As for subject F, we conducted the experiment with him twice to check the change in his experience. The second experiment was done after about half a year. Table 4 shows the results of subject F. He was not experienced in driving. Because he always paid attention to the driving itself, the value of "M-A" became a negative value. Moreover, the value of "difference" was small. This indicates that he was not used to driving either on a narrow road or on a wide road. However, the value of "difference" in the second experiment became bigger. The value of "narrow" was bigger than that of "wide". We can regard this as an improvement in his driving experience on a narrow road. We can attribute this to his prior knowledge of the near-miss incident area; moreover, his attention level increased when driving on the wide road. The value of "average" in the second experiment became bigger than that in the first one. This means that subject F gained more experience after about half a year.

Table 4. Experimental results of subject F

No.	Road	Meditation	Attention	M-A	Difference	Average
1	Narrow	42.3	54.2	-11.9	3.4	-13.6
	Wide	42.7	58.0	-15.3		
2	Narrow	61.8	59.4	2.4	8.1	-1.6
	Wide	56.5	62.2	-5.7		

4.2. Qualitative Model

We can generate a qualitative tree by using the qualitative induction function of QCAM. A qualitative tree is a model that represents the relationships between the qualitative changes of each parameter as a tree. We can interpret one leaf from the root as one rule. Here, we made four types of scene: no-lane road, single-lane road, two-lane road, and near-miss incident area, like those shown in scene window (2) in Figure 1.

The representative rules for each subject are as follows:

Subject A:
[Rule 1] Scene=no, ACCZ=DEC, Attention=DEC
[Rule 2] Scene=single, ACCZ=DEC, Attention=INC
[Rule 3] Scene=two, ACCZ=DEC, Attention=INC
Subject B:
[Rule 4] Scene=no, ACCZ=INC, Meditation=DEC, Attention=STD
[Rule 5] Scene=single, ACCZ=DEC, Attention=INC
[Rule 6] Scene=two, ACCZ=INC, Meditation=INC

Subject C:
[Rule 7] Scene=no, ACCZ=INC, Meditation=DEC, Attention=STD
[Rule 8] Scene=single, ACCZ=INC, Meditation=STD, Attention=DEC
[Rule 9] Scene=two, ACCZ=DEC, Meditation=STD
[Rule 10] Scene=two, ACCZ=INC, Meditation=STD, Attention=STD
[Rule 11] Scene=near, ACCZ=INC, Meditation=INC
Subject D:
[Rule 12] Scene=no, ACCZ=DEC, Attention=INC
[Rule 13] Scene=single, ACCZ=INC, Meditation=DEC
[Rule 14] Scene=two, ACCZ=INC, Meditation=INC
Subject E:
[Rule 15] Scene=no, ACCZ=INC, Meditation=DEC
[Rule 16] Scene=single, ACCZ=DEC, Meditation=DEC
[Rule 17] Scene=two, ACCZ=INC, Meditation=INC
[Rule 18] Scene=near, ACCZ=INC, Attention=DEC
Subject F (Second experiment):
[Rule 19] Scene=no, ACCZ=INC, Attention=DEC, Meditation=STD
[Rule 20] Scene=single, ACCZ=INC, Attention=STD
[Rule 21] Scene=two, ACCZ=INC, Attention=DEC

Here, “Scene” indicates each scene; “no,” the no-lane road; “single,” the single-lane road; “two,” the two-lane road; and “near,” the near-miss incident area. “ACCZ” means the acceleration back and forth. “Meditation” is the relaxation level and “attention” is the attention level. “INC” means increase, “DEC” means decrease, and “STD” means steady (constant) change.

As for subject A, he showed increased attention level by slowing down on both the single-lane road and the two-lane road (rules 2 and 3). As for subject F, he showed decreased attention level by speeding up on both the no-lane road and the two-lane road (rules 19 and 21). These rules have the same meaning. Because subjects A and F were not very much experienced in driving, they tended to show increased attention level by slowing down and to show decreased attention level by speeding up regardless of road type.

Subjects B and C are used to driving on a wide road. Therefore, subject B showed increased relaxation level by speeding up on the wide road (rule 6). However, because he is not used to driving on a narrow road, he showed decreased relaxation level by speeding up on the no-lane road (rule 4) and tended to show increased attention level by slowing down on the single-lane road (rule 5). Similar rules were also generated in the case of subject C. However, the rule of the near-miss incident area was applied (rule 11). Subject C showed increased relaxation level by speeding up despite having to drive through the near-miss incident area. We can consider that this rule was applied because he did not know about this area.

Subjects D and B are experienced drivers. In the case of the narrow road, they tended to show increased attention level by slowing down (rule 12) and showed decreased relaxation level by speeding up (rules 13 and 15). In the case of the relatively wide road, their relaxation level rose by speeding up (or dropped by slowing down) (rules 14, 16, and 17). Because subject E knew about the near-miss incident area, the rule concerning the area was applied (rule 18). The rule means that he showed decreased attention level by speeding up (or increased attention level by slowing down).

Thus, we confirmed the rules that explain the experiences and skills of each driver.

5. DISCUSSION OF THE EXPERIENCE-BASED APPROACH

We analyzed and modeled the driving data of six subjects with different individual driving experiences and skills by using the scene analysis function of QCAM. From the experiment result, we found a parameter that indicates the degree of experience and skill of each driver. Moreover, from the models, we derived the rules that explain the experience and skill of each driver.

The cognitive-experiential theory (Epstein, 2014) exists as a cognitive theory about experience. This theory states that human personality consists of a "rational system" and an "experiential System". The rational system is a conscious reasoning system that solves problems according to people's understanding of logical principles and to the evaluation of evidence. On the other hand, the experiential system is an unconscious reaction system that solves problems in adaptation by automatically reacting according to an organism's reinforcement history. When we are learning something, we behave and think consciously in the early stage. We practice repeatedly, and then, eventually, we act unconsciously. Driving is also similar. When we are a novice driver, we operate a car and think rationally while driving. However, as we get used to driving gradually, we eventually do it unconsciously. Therefore, we can think that the factor of driving transfers from the rational system to the experiential system.

Here, for the safety of driving, we need to point out that understanding of "common sense" between drivers and pedestrian is very important (Mizoguchi, Yoshizawa & Iwasaki, 2017). For example, a driver may think that a pedestrian must stop in certain cases. Moreover, we need to point out that "affordance" is key in a narrow road crossing (Mizoguchi, Yoshizawa & Iwasaki, 2018). Affordance is the factor that causes a certain action to be performed according to the situation. For instance, there are roads where everyone tends to drive fast. "Common sense" and "affordance" come about by experience unconsciously, and they are factors that are processed passively by the surroundings.

As defined in the cognitive-experiential theory, experience is unconscious and passive. Our experience-based analysis and modeling enable an understanding of unconscious operation and situated cognition (Clancey, 1997) based on the drivers' experiences.

6. CONCLUSION

In this paper, we discussed an experience-based approach to cognitive vehicle research. We recruited six subjects who have different individual driving experiences and skills. We made them drive their cars on the same test course. We collected the drivers' mental status and operations as the driving data. We analyzed the data and made models by using QCAM. We experimentally verified how different the results of the analysis and modeling were. From the experiment result, we found a parameter that indicates the degree of experience and skill of each driver. Moreover, from the models, we derived the rules that explain the experience and skill of each driver.

Using our experience-based approach, we recorded the drivers' experiences as numeric values and saved these experiences as models or rules. Recently, accidents involving elderly drivers have become a big social problem in Japan. Our approach allows such drivers to record their driving experiences and replay the recordings so that they know what to maintain or improve in their driving skills. The development of a driving simulator based on actual driving experiences will be one of the industrial applications of our approach.

We considered road types such as a narrow road or a wide road as static road conditions. We treated road conditions such as traffic volume, traffic signal, and pedestrian as dynamic road conditions. Therefore, we extracted such dynamic information, using object detection techniques by deep learning (Elmisery, Sertovic & Gupta, 2018), from the video recorded by a driving recorder. One of our further works is to clarify the effects of dynamic road conditions.

ACKNOWLEDGMENT

This research is supported by JSPS KAKENHI Grant Number JP17K01709. I would like to thank Editage (www.editage.com) for English language editing.

REFERENCES

Clancey, W. J. (1997). *Situated Cognition*. Cambridge University Press.

Elmisery, A. M., Sertovic, M., & Gupta, B. B. (2018). Cognitive Privacy Middleware for Deep Learning Mashup in Environmental IoT. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 8029–8041. doi:10.1109/ACCESS.2017.2787422

Epstein, S. (2014). *Cognitive-Experiential Theory: An Integrative Theory of Personality*. Oxford University Press. doi:10.1093/acprof:osobl/9780199927555.001.0001

Hiraishi, H. (2017). Qualitative Analysis of Concentration Level in Throwing Using Simple Brain-Wave Sensor. *International Journal of Cognitive Informatics and Natural Intelligence*, *11*(3), 17–30. doi:10.4018/IJCINI.2017070102

Hiraishi, H. (2018). Passenger Condition Based Route-Planning for Cognitive Vehicle System. *International Journal of Software Science and Computational Intelligence*, *10*(2), 25–35. doi:10.4018/IJSSCI.2018040102

Hiraishi, H. (2019). Qualitative and Cognitive Analysis and Modeling Tool for Biological Data. *International Journal of Cognitive Informatics and Natural Intelligence*, *13*(2), 30–47. doi:10.4018/IJCINI.2019040103

Marvin, T. C., Jonathan, T. C., Christine, C., & Craig, G. (2017). An Artificial Intelligence-Based Vehicular System Simulator. *International Journal of Software Science and Computational Intelligence*, *9*(1), 54–67.

Mizoguchi, F., Yoshizawa, A., & Iwasaki, H. (2017). Common-Sense Approach to Avoid Near-Miss Incidents of Pedestrians Suddenly Crossing Narrow Roads. *The 16th International conference on Cognitive informatics & Cognitive Computing, ICCI*CC'17*, 335-340.

Mizoguchi, F., Yoshizawa, A., & Iwasaki, H. (2018). Study of Car Driver's Judgment Affordance on a Narrow Crossing Road. *The 17th International conference on Cognitive informatics & Cognitive Computing, ICCI*CC'18*, 366-372. doi:10.1109/ICCI-CC.2018.8482056

Sega, S., Iwasaki, H., Hiraishi, H., & Mizoguchi, F. (2011). Qualitative Reasoning Approach to a Driver's Cognitive Mental Load. *International Journal of Software Science and Computational Intelligence*, *3*(4), 18–32. doi:10.4018/jssci.2011100102

Šuc, D., Vladusic, D., & Bratko, I. (2004). Qualitatively faithful quantitative prediction. *Artificial Intelligence*, *158*(2), 189–214. doi:10.1016/j.artint.2004.05.002

Yoshizama, A., Nishiyama, H., Iwasaki, H., & Mizoguchi, F. (2017). Estimation of Cognitive Distraction from Driver Gazing. *International Journal of Software Science and Computational Intelligence*, *9*(2), 50–60. doi:10.4018/IJSSCI.2017040104

Zuojin, L., Jun, P., Liukui, C., Ying, W., & Jinliang, S. (2017). A Method for Identifying Fatigue State of Driver's Face Based on Improved AAM Algorithm. *International Journal of Software Science and Computational Intelligence*, *9*(2), 31–49. doi:10.4018/IJSSCI.2017040103

*Hironori Hiraishi received BS degree in 1993, his MS degree in 1996, and a Ph.D. in 1999 in information technology from Tokyo University of Science. He was a research associate of Information Media Center in Tokyo University of Science from 2000 to 2005, chief technological officer of WisdomTex, Inc. from 2001 to 2009, an associate professor of department of electrical and computer engineering in National Institute of Technology, Akita College from 2010 to 2016, and an associate professor of faculty of engineering in Ashikaga Institute of Technology from 2017 to 2019. He is a professor of faculty of engineering in Ashikaga University from 2020. He received the winning award on the Innovative Application conference (IAAI2003), and the best paper award twice on the Cognitive Informatics and Cognitive Computing conference (ICCI*CC2011 and ICCI*CC2017). His recent research interests have focused on the applications integrating with human cognitive aspects and AI technologies.*

International Journal of Software Science and Computational Intelligence
Volume 12 • Issue 4 • October-December 2020

Evaluation of NoSQL Databases:
MongoDB, Cassandra, HBase, Redis, Couchbase, OrientDB

Houcine Matallah, Department of Computer Science, University of Tlemcen, Algeria
Ghalem Belalem, University of Oran 1, Algeria
https://orcid.org/0000-0002-9694-7586
Karim Bouamrane, University of Oran 1, Algeria
https://orcid.org/0000-0002-6953-0339

ABSTRACT

The explosion of the data quantities, which reflects the scaling of volumes, numbers, and types, has resulted in the development of new locations techniques and access to data. The final steps in this evolution have emerged new technologies: cloud computing and big data. The new requirements and the difficulties encountered in the management of data classified "big data" have emerged NoSQL and NewSQL systems. This paper develops a comparative study about the performance of six solutions NoSQL, employed by the important companies in the IT sector: MongoDB, Cassandra, HBase, Redis, Couchbase, and OrientDB. To compare the performance of these NoSQL systems, the authors will use a very powerful tool called YCSB: Yahoo! Cloud Serving Benchmark. The contribution is to provide some answers to choose the appropriate NoSQL system for the type of data used and the type of processing performed on that data.

KEYWORDS

Big Data, Cassandra, Couchbase, HBase, MongoDB, NoSQL, OrientDB, Redis, YCSB

INTRODUCTION

The plethora of sources to create digital data and extension of computer science in different sectors and areas (Astrology, Meteorology E-Commerce, E-Government, Multimedia, etc.) exploded amounts of data, which reflects the scaling volumes and types. It is extremely difficult to estimate the quantities of digital data produced every day in the world of business, government and individuals, whether photographs, videos, texts, tweets, or emails.

Computer designs since the nineties used data warehouses (Inmon, 2005), which are usually centralized in servers connected to storage arrays. These architectures poorly scalable (addition of power on demand). Indeed, the growing volume of data, the wide heterogeneous data, and the data velocity, traditional DBMS and even Data Warehouses have struggled to adapt.

This scientific revolution that invading the world of IT has imposed new issues that have led to the development of new technologies to contain and process these large volumes of data. The goal is

DOI: 10.4018/IJSSCI.2020100105

to discover new orders of magnitude to capture, search, share, store, analyze, and present data. This new IT era has led to replacing traditional databases limited by ACID constraints with new solutions that respond to these imposed changes. These new requirements have led to the emergence of the movement NoSQL (Cattell, 2011; Oussous, Benjelloun, Lahcen, & Belfkih, 2017) and NewSQL movement (Aslett, 2011; Piekos, 2015).

Several open-source and proprietary NoSQL solutions have been designed, developed and deployed by the big companies of the sector, to manage large volumes of data manipulated. However, the lack of standardization and the panoply of solutions proposed in the market complicates the choice of the model appropriate to the operating environment, which poses a real problem on the best NoSQL solution to adopt compared to the user needs.

The contribution presented in this paper is to provide indicators which can help interested actors to decide on the solutions adopted by their companies, by developing a comparative study on a set of NoSQL solutions widely deployed on the market. This study compares the performance of NoSQL Databases from the experimental point of view. Note that the current work is an extension of a first work in which the performances of MongoDB and HBase were compared and which was the subject of a paper published (Matallah, Belalem, & Bouamrane, 2017a).

Our study focuses on six data management solutions characterized by the implementation in their kernels of the same algorithm “MapReduce” (Lattanzi, Moseley, Suri, & Vassilvitskii, 2011), these are the MongoDB (Chodorow, 2013; Membrey, Plugge, & Hawkins, 2011), Cassandra (Lakshman & Malik, 2010), HBase (George, 2011), Redis (Macedo & Oliveira, 2011), Couchbase (Brown, 2012), and OrientDB (Tesoriero, 2013) models. To evaluate and compare the available NoSQL solutions, several benchmarks have been designed, the most commonly used is the YCSB (Cooper, Silberstein, Tam, Ramakrishnan, & Sears, 2010).

This paper will be organized as follows: In the first Section of the manuscript, we expose the limitations of relational DBMS in large scale distributed environments which led to the emergence of NoSQL. In the second Section, we will present the NoSQL data management systems designed to meet the new needs required for scaling up. In in the third Section, we will focus on the six NoSQL solutions compared and the benchmark used. After assessing the performance of each database, the different experimental results of this comparative study will be synthesized and analyzed in the fourth Section. The paper is concluded with a summary and some perspectives for our future works.

LIMITS OF RELATIONAL SYSTEMS AND APPEARANCE OF NOSQL

The relational model was created by Edgar Codd (Codd, 1970) in the 1970s and has been used by the big companies. It was, and still is, widely used today because of its simplicity and strong mathematical foundations. In this model, the data is stored in a table, all of these tables constitute a relational database with a schema. The schema is a structural representation of the database contents, that defines the tables, the fields of the tables, and the relations between the two. The data is stored as tables where each column corresponds to a specific attribute and each line corresponds to a record or a tuple. The data in each table can be read, deleted, and updated using the standard language for relational databases: Structured Query Language (SQL). The management, storage, updating, sharing, consistency, and security of data are provided by a Relational Data Base Management System (RDBMS). This model is based on simple concepts that have both strengths and weaknesses. Indeed, relational database systems respond well to the transactional need thanks to the ACID properties (Atomicity, Coherence, Isolation, Durability). Likewise, the extensions based on this model (such as cubes and star schema) designed to meet the need for analysis in large consolidated databases (Data Warehouse) (Inmon, 2005). However, these relational systems cannot be deployed in a large-scale Cloud environment, while maintaining the same performance.

Mainly, the problems of the relational model related to this scaling up are (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012):

1. Limited data types;
2. Applying ACID properties in a distributed environment;
3. Non-optimal join request;
4. Management of heterogeneous objects.

Faced with the masses of heterogeneous data exchanged and the scaling up, the standard RDBMS have shown great weaknesses in terms of scalability, mainly because of the respect of ACID properties. The important providers of web services such as Google, Yahoo, Facebook, Twitter, Amazon, were quickly limited by traditional systems. Not finding a solution in the market that addresses these issues, then, they decided to solve their problems by designing new architectures and developing each one internally and in parallel with their main activities, their own DBMS. So, the giant Google has developed Bigtable, to manage its Big Data, used by many Google services, such as search, Maps and Gmail. Facebook used Cassandra then HBase for the search function in the mailbox. Amazon developed DynamoDB for its e-commerce. LinkedIn Voldemort designed to manage updates online from various features of intensive writes on its website. Yahoo has built PNUTS to store, user data that can be read or written on any web page, data lists for Yahoo's shopping pages as well as data to serve its social networking applications. These new solutions, developed separately, are known as the NoSQL. Indeed, it was in 2009, at a meeting of the non-relational DBMS developer community that the term NoSQL was adopted to name all non-relational DBMS (Rith, Lehmayr, & Meyer-Wegener, 2014). Since that, NoSQL databases have begun to emerge, to meet the new needs required by the Big Data and Cloud Computing environments.

This new NoSQL movement will be presented in the following Section.

NoSQL

"NoSQL" is a combination of two words: No and SQL. The name may seem like an opposition to SQL databases and could be misinterpreted as the end of the SQL language which should not be used anymore. In fact, the "No" is an acronym that means "Not only", so it's a way of saying that there is something other than relational databases and SQL (Cattell, 2011; Oussous et al., 2017).

The aim of the movement not to replace the relational systems called SQL, but to propose alternatives or to complete the functionalities of these models to handle large masses of distributed data and adapt to new trends and architectures of the moment, including Cloud Computing (Rith, Lehmayr, & Meyer-Wegener, 2014).

Unlike relational systems based on ACID properties, NoSQL systems prioritize scalability and high availability, to the detriment of consistency.

The founders of NoSQL such as Google, Amazon, Facebook, and others whose availability needs take precedence over data consistency have decided to privilege the availability to the detriment of consistency. The opposition of these two properties was presented by Eric Brewer in his famous CAP theorem (Gilbert & Lynch, 2002). This theorem demonstrates that the database in a distributed system can only guarantee two of these constraints at the same time, but not all three:

- **Consistency:** All nodes (servers) in the system exactly view the same data at the same time;
- **Availability:** Guarantee that any query receives a reply;
- **Partition Tolerance:** The system must be able to respond correctly to all queries in all times except in a general network failure. In the case of a partitioning into subnets, each of these subnets must be able to operate autonomously.

Therefore, only the CA systems (Consistent and highly Available, but not Tolerant), the CP systems (Consistent and Tolerant, but not Available) and PA systems (very Available and Tolerant, but sacrificing Consistency) are possible.

Most NoSQL databases are the PA type (Gilbert & Lynch, 2002), which means they focus more on the resistance to partitioning, to ensure availability at all times and therefore renounce consistency.

Interests

The main interests of NoSQL systems are:

- Proposition of alternatives to relational databases to resolve the difficulties of managing Big Data;
- Abandon of the tabular representation of the information and the language SQL;
- Manipulation of large amounts of distributed data;
- Emphasis on Performance (Latency, Throughput), Extensibility, Availability;
- Adoption of Parallel File Systems.

Strengths and Weaknesses

Here below some strengths and weaknesses of these new architectures: (Han, Haihong, Le, & Du, 2011; McCreary & Kelly, 2014):

1. More than rows in tables: Supports mass storage by multiplying the number of records in tables (A table can contain millions of rows);
2. Management of heterogeneous data from different data sources;
3. Free of join: It not obliged to apply joins between tables. The data are generally integrated in the same container. This form of storage can be considered as already executed joins which allows a fast reading of data;
4. Provision of fast response times because of the simplicity of schemas and the lack of joins;
5. Support of Scale-out: NoSQL systems are designed to increase workload by automatically distributing data between several nodes and forming a cluster to maintain highly scalable linear performance. A NoSQL database can be spread on multiple servers without the help of the application. In addition, servers can be added or removed easily. Logically, a NoSQL system should never have to restart;
6. Support of parallel processing models, in particular MapReduce, which is often embedded in NoSQL solutions. This programming model improves and simplifies the parallel computations in distributed architectures;
7. Support of distributed queries: The sharding of a database can in some cases counteract the execution of a complex query. This problem is solved in NoSQL (even in some relational DBMS like SQL Server and Oracle) where we can keep all the power of the language itself to recover data spread over multiple servers;
8. It is designed to work on multiple processors: Break down the problem into multiple threads with many CPUs working together;
9. Adoption of shared-nothing architecture nodes: Each node of the cluster has its own CPU, RAM and disk;
10. Low costs of managing, operating and storing large volumes of data.

However, most NoSQL solutions, have some deficiencies as:

1. Consistency of data relatively abandoned in favor of high availability;
2. Non-existence of standardized query languages such as SQL;
3. Maturity and stability: Relational bases have a head start on this point. Users are familiar with their operation and trust them;

4. Data security and lack of other insufficiently developed additional features: NoSQL systems are based on the application for data protection;
5. Less documentation and tools are available for users.

Types of NoSQL Databases

There are a variety of architectures and technologies, called NoSQL, which are distinguished by their manners of representing data. These different systems can be classified into four main categories (Cattell, 2011; Indrawan-Santiago, 2012):

Columns-Oriented

In terms of architecture, column-oriented databases are closer to relational databases. The principle of a column database consists in their storage by column and not by row. They constitute the stern of the NoSQL movement because of their recurrence in the news with products like HBase or Cassandra (Oussous et al., 2017). They were designed by the web giants to cope with the management and processing of large volumes of rapidly growing data. The main NoSQL solutions Columns-Oriented are: Cassandra (Apache), HBase (Apache), Bigtable (Google), Accumulo (Apache), Hypertable, etc. (Cattell, 2011; Grolinger et al., 2013; Hecht & Jablonski, 2011).

Key-Value Oriented

This architecture is the simplest of the others. The principle employed is very simple: Each stored data item corresponds to a single key. All queries on the data will use this key. The most known solutions having adopted the Key-value oriented system are: Redis, Memcached, Amazon DynamoDB, Riak, Voldemort Ehcache, Hazelcast (used by LinkedIn), OrientDB, Berkeley DB, Oracle NoSQL, etc. (Cattell, 2011; Grolinger et al., 2013; Hecht & Jablonski, 2011).

Document-Oriented

Document-oriented databases are an alternative to column-oriented databases. They therefore operate on the same key-value associative principle (here key-document), but with a feature addition that considers the structure of data stored as a document. In these models, the keys are no longer associated with values in the form of a binary block but with a document of non-imposed format (Han et al., 2011; Indrawan-Santiago, 2012). The main Document-Oriented NoSQL databases are: MongoDB (10gen), CouchDB, CoucHBase, Amazon DynamoDB, MarkLogic, RavenDB, Cloudant, OrientDB, GemFire, RethinkDB, Datameer, Microsoft Azure DocumentDB, ArangoDB, PouchDB, etc. (Cattell, 2011; Grolinger et al., 2013; Hecht & Jablonski, 2011).

Graph-Oriented

This model is not designed to solve performance problems, but rather to overcome the very complex problems of connectivity links between data that a relational database would be unable to do. The graph approach is therefore inevitable for applications such as social networks (Han et al., 2011; Indrawan-Santiago, 2012). The most known solutions that have adopted the Graph-Oriented NoSQL are: Neo4j, Orient DB, Titan, ArangoDB, Giraph, InfiniteGraph, Sqrrl, Sparksee, InfoGrid, HyperGraphDB, FlockDB, VelocityGraph, GlobalsDB, GraphDB, etc. (Cattell, 2011; Grolinger et al., 2013; Hecht & Jablonski, 2011).

Multi-Model

Apart from the four recognized models, there are other hybrid solutions that have been designed to meet specific needs like: Amazon DynamoDB, Microsoft Azure Cosmos DB, Datastax Enterprise, Ignite, ArangoDB, Apache Drill, Virtuoso, etc. (Cattell, 2011; Grolinger et al., 2013; Hecht & Jablonski, 2011). Multi-model databases can support different models within the engine or through different layers above the engine. Document, graphic, key-value and relational models are examples of data models that can be supported by a multi-model database. This unique database is made up of several sub-databases of different architectures dedicated to different applications (Example: relational database for enterprise customers and document-oriented database for the general public).

The NoSQL solutions compared in this study and the used comparison tool will be the subject of the next Section.

COMPARATIVE STUDY

The different NoSQL models existing in the market adopt different architectures since they were designed, developed, and deployed in different sectors to meet different needs. There are currently in the IT market more than 305 NoSQL solutions of different architectures.

In this context, this comparative study consists of evaluating the performance of six NoSQL solutions, in order to choose the system appropriate to the type of data used and to the nature of the processing executed on this data. To evaluate and compare NoSQL solutions, several benchmarks have been designed, the most commonly used is Yahoo's YCSB (Cooper et al., 2010). Several comparative studies in the same context using YCSB have been developed (Abramova & Bernardino, 2013; Abramova et al., 2014; Abramova et al., 2015; Ahamed, 2016; Cattell, 2011; Gandini, Gribaudo, Knottenbelt, Osman, & Piazzolla, 2014; Kumar & Mary, 2017; Lungu & Tudorica, 2013; Martins, Bezerra, Gomes, Albuquerque, & Costa, 2015; Park, 2016; Singh, 2019; Swaminathan & Elmasri, 2016; Tang & Fan, 2016). Among these works, the researchers cite their previous work (Matallah et al., 2017a), where MongoDB was opposed to HBase.

Popularity Ranking NoSQL Systems

In the comparative study, the authors targeted popular solutions, according to Solid IT (Solid, 2019). This DB-Engines Ranking is an updated list of 350 database management systems of different models, sorted by popularity. They measure the popularity of a system using the following parameters:

1. Number of system citations in websites;
2. General interest in the system;
3. Relevance in social networks;
4. Frequency of technical discussions on the system;
5. Number of jobs offers in which the system is mentioned;
6. Number of profiles in the professional networks in which the system is mentioned.

The Top 50 ranking of July 2019 is shown in Table 1.

The 17 NoSQL systems in the Top 50 are of different architectures and are deployed in various projects in different sectors. The solutions chosen in this study, are recalled in Table 2.

The six solutions selected in this comparative study are of different types, two for each type (MongoDB and Couchbase as document-oriented, Cassandra and HBase as column-oriented, Redis and OrientDB as a key-value). The first five solutions are well placed at the top of the NoSQL database popularity ranking. The only exception was made for OrientBD ranked 50th, but chosen for its portability and its multi-model architecture (key-value, document-oriented, graph-oriented). Moreover, note that these solutions are widely used and compared in several works and studies (Abramova et al., 2013; Abramova et al., 2014; Abramova et al., 2015; Ahamed, 2016; Cattell, 2011; Gupta, 2018; Kumar & Mary, 2017; Lungu & Tudorica, 2013; Martins et al., 2015; Matallah et al., 2017a; Park, 2016; Singh, 2019; Swaminathan & Elmasri, 2016; Tang & Fan, 2016).

Table 1. Top 50 of DBMS (Solid, 2019)

Rank Jul 2018	Rank Jun 2018	Rank Jul 2017	DBMS	Database Model	Score Jul 2018	Score Jun 2018	Score Jul 2017
1.	1.	1.	Oracle	Relational DBMS	1277.79	-33.47	-97.09
2.	2.	2.	MySQL	Relational DBMS	1196.07	-37.62	-153.04
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1053.41	-34.32	-172.59
4.	4.	4.	PostgreSQL	Relational DBMS	405.81	-4.86	+36.37
5.	5.	5.	MongoDB	Document store	350.33	+6.54	+17.56
6.	6.	6.	DB2	Relational DBMS	186.20	+0.56	-5.05
7.	7.	9.	Redis	Key-value store	139.91	+3.61	+18.40
8.	8.	10.	Elasticsearch	Search engine	136.22	+5.18	+20.25
9.	9.	7.	Microsoft Access	Relational DBMS	132.58	+1.59	+6.45
10.	10.	8.	Cassandra	Wide column store	121.06	+1.84	-3.07
11.	11.	11.	SQLite	Relational DBMS	115.28	+1.02	+1.41
12.	12.	12.	Teradata	Relational DBMS	78.22	+2.45	-0.14
13.	14.	16.	Splunk	Search engine	69.24	+3.46	+8.94
14.	13.	18.	MariaDB	Relational DBMS	67.51	+1.67	+13.15
15.	16.	13.	SAP Adaptive Server	Relational DBMS	62.12	+0.64	-4.79
16.	15.	14.	Solr	Search engine	61.52	-0.55	-4.51
17.	17.	15.	HBase	Wide column store	60.77	+1.07	-2.85
18.	18.	20.	Hive	Relational DBMS	57.63	+0.30	+11.42
19.	19.	17.	FileMaker	Relational DBMS	56.39	+0.21	-2.26
20.	20.	19.	SAP HANA	Relational DBMS	51.60	+2.25	+3.65
21.	21.	22.	Amazon DynamoDB	Multi-model	49.63	+3.84	+13.17
22.	22.	21.	Neo4j	Graph DBMS	41.88	-0.09	+3.36
23.	23.	24.	Memcached	Key-value store	33.88	+0.07	+5.35
24.	24.	23.	Couchbase	Document store	33.07	+0.62	+0.06
25.	26.	26.	Microsoft Azure SQL Database	Relational DBMS	26.84	+0.55	+4.55
26.	25.	25.	Informix	Relational DBMS	26.59	+0.03	-1.08
27.	27.	28.	Vertica	Relational DBMS	20.82	-0.34	-0.97
28.	28.	30.	Firebird	Relational DBMS	20.66	+0.27	+1.67
29.	29.	27.	CouchDB	Document store	19.50	-0.70	-2.65
30.	30.	41.	Microsoft Azure Cosmos DB	Multi-model	19.45	+0.25	+11.74
31.	31.	29.	Netezza	Relational DBMS	16.90	-0.20	-2.96
32.	32.	32.	Amazon Redshift	Relational DBMS	14.72	+0.35	+1.88
33.	33.	35.	Google BigQuery	Relational DBMS	13.34	+0.08	+2.04
34.	34.	31.	Impala	Relational DBMS	13.29	+0.26	-0.02
35.	35.	37.	Spark SQL	Relational DBMS	12.73	+0.81	+2.08
36.	36.	40.	InfluxDB	Time Series DBMS	11.70	+0.37	+3.59
37.	37.	33.	MarkLogic	Multi-model	11.31	+0.34	-1.12
38.	39.	34.	Greenplum	Relational DBMS	10.80	+0.19	-0.82
39.	38.	38.	dBASE	Relational DBMS	10.79	-0.01	+0.30
40.	40.	36.	Oracle Essbase	Relational DBMS	9.13	-0.06	-1.98
41.	41.	39.	Hazelcast	Key-value store	8.76	-0.33	-0.15
42.	44.	58.	Firebase Realtime Database	Document store	7.54	+0.91	+3.47
43.	42.		Datastax Enterprise	Multi-model	7.36	+0.10	
44.	48.		Microsoft Azure SQL Data Warehouse	Relational DBMS	7.09	+0.87	
45.	45.	45.	Sphinx	Search engine	6.80	+0.42	+0.37
46.	43.	44.	Ehcache	Key-value store	6.42	-0.35	-0.73
47.	46.	43.	Interbase	Relational DBMS	6.33	+0.06	-0.96
48.	47.	42.	Riak KV	Key-value store	6.28	+0.02	-1.08
49.	50.	60.	Realm	Relational DBMS	5.57	+0.23	+1.54
50.	49.	46.	OrientDB	Multi-model	5.39	+0.04	-0.18

Table 2. Ranking studied NoSQL solutions (Solid, 2019)

System	Ranking Among NoSQL	General Ranking
MongoDB	1	5
Redis	2	7
Cassandra	3	10
HBase	4	17
Couchbase	8	24
OrientDB	17	50

MongoDB

MongoDB is an open-source document-oriented database, distributed under the free-license (AGPL), providing high performance, high availability and automatic scalability (Chodorow, 2013; Membrey et al., 2011). MongoDB has been developed in C ++ since 2007 by the 10gen company that was working on a widely distributed Cloud Computing system, similar to Google's AppEngine service. The first version appeared in 2009, but it was in 2010 that version 1.4 was considered industrially viable. MongoDB is scalable, very powerful, easy to use, and well-designed to manage storage for large-scale applications, and it can execute in a distributed and multi-platform environment. As a result, MongoDB has been adopted by several major companies such as Foursquare, SAP, and GitHub. MongoDB allows manipulating structured objects (documents) in BSON (Binary JSON) format, a more performance-oriented binary JSON derivative that has been designed to facilitate data scanning. Functioning as a distributed-centralized architecture, it replicates data on multiple servers with the master-slave or primary / secondary principle, allowing for greater fault-tolerance (failover) and workload distribution between nodes (load balancing). The distributed architecture of MongoDB is based on two principles: data distribution (sharding) and replication (replica set). The distribution and duplication of documents is done so that the most requested documents are on the same server and that it is duplicated a sufficient number of times. The performance of MongoDB in a distributed environment has also been used to implement new data replication strategies for balancing the workload of the nodes (Tabet, Mokadem, & Laouar, 2018; Tabet, Mokadem, & Laouar, 2019). Because of its ease of use from a customer development perspective, as well as its outstanding performance, MongoDB is the most widely used document-oriented database (Membrey et al., 2011; Solid, 2019).

Redis

Redis is a key-value type NoSQL database, an acronym for REmote DIctionary Server, written with the C programming language and distributed under BSD license. It is part of the NoSQL movement whose fundamental aim is to have the highest possible performance. Redis allows manipulating simple data types: strings, associative arrays, sets, ordered sets, and lists. After having experienced a big growth in 2010, Redis continues its way quietly with the addition of the "sentinel failover" which is in charge of monitoring, notifying, and automatically switching instances in case of problems. The integration of Lua language, because everyone uses JavaScript, will further extend the possibilities of this key-value-oriented database. Currently, it is used by companies like The Guardian, GitHub, Stack Overflow, and Craigslist or Blizzard Entertainment (Macedo & Oliveira, 2011).

The two of the most popular NoSQL databases (MongoDB, Redis) are used for a number of applications in the real world especially social networks (Gupta & Gugulothu, 2018).

Cassandra

Cassandra is a column-oriented NoSQL database. It has been designed to overcome the performance problem of relational databases, as well as the problems of managing large volumes of data. In addition, it addresses the complexity of deployment and the maintenance of integrity during cluster deployment and in different datacenters. Cassandra is designed to handle massive amounts of data spread across multiple servers (Cluster), with particular emphasis on maximizing data availability and eliminating individual points of failure (Lakshman & Malik, 2010).

Initially, it was developed by Facebook, to meet needs concerning its courier service, then it was released in Open-source and has been adopted by other major Web companies such as Digg.com or Twitter. It is today one of the main projects of the Apache Foundation.

Considerable efforts are currently being devoted to Cassandra to add new features as and when the improvement of its requests. However, the discussions do not stop with the fact that the configuration of a Cassandra cluster is not a part of pleasure. The most notable point remains DataStax, the company behind Cassandra, which is striving to integrate Data Mining capabilities. Since the beginning

Cassandra was not considered as a powerful query system, but recently and with the addition of new features, it becomes possible to exploit it in a context of data analysis.

HBase

HBase is a column-oriented NoSQL database, open source, written in Java inspired by Google's publications on BigTable (George, 2011), with structured storage in Big Data environments.

HBase is a Hadoop subproject licensed by Apache, developed as part of the Apache Hadoop project of Apache Software Foundation. The American company Cloudera distributes an edition of Hadoop and HBase in Cloudera Enterprise support. The major objective of this project is the hosting of very large tables of billions of lines and millions of columns. HBase is based on Hadoop Distributed File System HDFS, which is characterized by high fault tolerance and massive parallelism (Jehangiri, Yahyapour, Yaqub, & Wieder, 2018; Matallah, Belalem, & Bouamrane, 2017b). HBase can manage random read/write access for real-time applications by keeping a large number of files open at the same time (George, 2011).

Facebook announced in 2010 that it would now use HBase to replace Cassandra, to store all messages exchanged in the social network, which represented more than 150 terabytes of new data per month (Liu, Gao, Chu, & Lu, 2017).

Couchbase

Since its birth in 2011 via the fusion of Membase and CouchOne, Couchbase has become an important element in the NoSQL environment with its Couchbase Server product (Brown, 2012). Couchbase Server belongs to the NoSQL document-oriented database class designed for interactive web applications. It has a flexible, easily scalable data model with consistent high performance that can serve application data with 100% availability. It is a persistent database that exploits an integrated RAM caching layer (Ostrovsky, Rodenski, & Haji, 2015).

The latest features of Couchbase are as follows (Brown, 2012):

- **Change From a Key-Value Orientation to a Document Orientation:** Native JSON support that allows obtaining documents with different structures;
- **Query Enhancement and Indexing:** Documents can be indexed via views. All indexes are automatically distributed on the nodes of a cluster. Queries now support searches by key, filter and aggregates. Queries are also processed during topology changes (failover etc.);
- **XDCR Replication (Cross Data-Centers):** Replication of data on several clusters distributed in several data centers in order to anticipate natural disasters or to bring data closer to their users for better performance.

OrientDB

OrientDB is an open source NoSQL database, written in Java, that can be deployed on any platform. This model is fully transactional, supporting ACID transactions that ensure the reliable processing of all database transactions. As a result, all pending documents, in the event of an incident, are recovered and committed. It is possible to execute it in distributed mode, in integrated mode or in memory, extremely useful for development, tests or small autonomous applications (Tesoriero, 2013).

OrientDB is a multi-model system that can be used as a document-oriented, graph-oriented or key-value database.

YCSB

YCSB: Yahoo! Cloud Serving Benchmark is a standard reference tool for NoSQL systems (Cooper et al., 2010). This tool is multi-platform and supports the majority of NoSQL systems and easily adapts to these solutions. It is often used to compare the relative performance of NoSQL systems in the cloud, this is confirmed by the multitude of studies that have used it (Abramova & Bernardino, 2013; Abramova et al., 2014; Abramova et al., 2015; Ahamed, 2016; Cattell, 2011; Gandini et al., 2014; Kumar & Mary, 2017; Lungu & Tudorica, 2013; Martins et al., 2015; Park, 2016; Singh, 2019; Swaminathan & Elmasri, 2016; Tang & Fan, 2016). In addition to these works, the authors quote their previous work (Matallah et al., 2017a), where MongoDB was opposed to HBase. YCSB consists of two components: a data generator and a set of performance tests to evaluate insertion, update, and delete operations. Each test is a workload, where you can configure the number of records to load, the number of operations to execute, and the proportion of read and write. Supported operations include: inserting, updating (modifying one of the fields), reading (a random field or all the fields of a record) as well as the scan (reading the recordings in the order of starting from a key in a randomly selected record). Each workload uses different reference parameters, which can be modified and customized according to the type of expected results.

All the results obtained in the experimental phase in the form of graphs are illustrated and commented in the next Section.

EXPERIMENTAL RESULTS

The experimental analysis was done in a physical machine: Core i5, 8GB RAM and HD 1 TB. The software environment used is Ubuntu 16.04, 64-bit, MongoDB 3.2.8, CouchBase 2.5.2, Cassandra 2.2.5, HBase 0.94.8, Redis 3.0.2, and OrientDB 2.1.3.

The execution times chosen as indicators in our study were collected and compared. Also note that to diminish the effect of variations in CPU and I / O performance, the same tests were performed on three different days to keep only the average execution time obtained.

The version of the benchmark used is YCSB 0.8.0. The benchmark package provides a set of default workloads that may be executed, as follows:

1. **Workload A (Update Heavy):** Constituted of a ratio of 50% Read and 50% Update;
2. **Workload B (Read Mostly):** Constituted of a ratio of 95% Read and 5% Update;
3. **Workload C (Read Only):** 100% Read;
4. **Workload D (Read-Modify-Write):** Constituted of a ratio of 50% Read, 50% Read-Modify-Write;
5. **Workload E (Update Mostly):** Constituted of a ratio of 5% Read and 95% Update;
6. **Workload F (Update Only):** 100% Update.

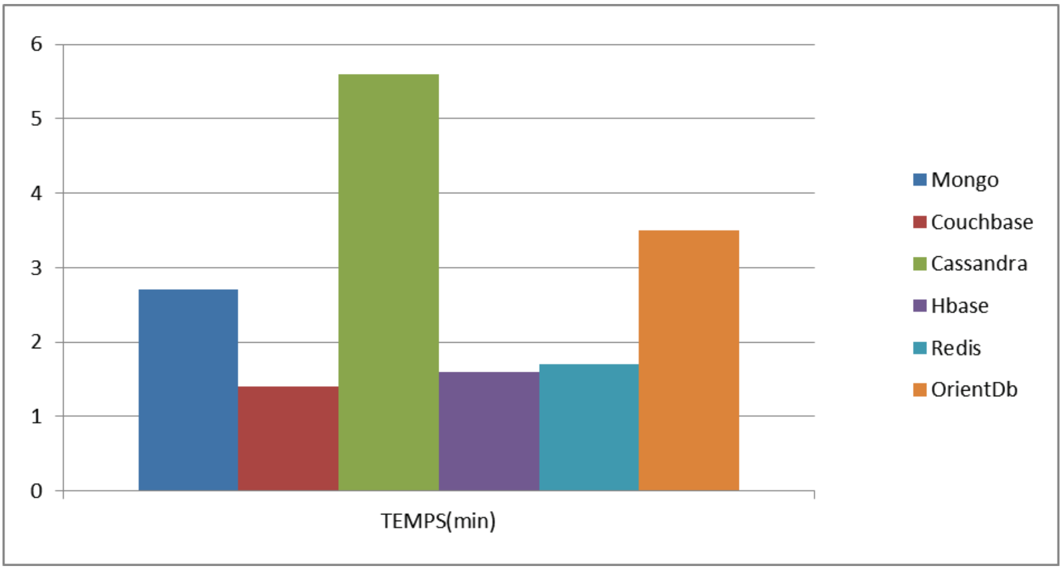
In this Section, the authors will expose the load results of 600 000 records generated with YCSB, each of which consists of 10 randomly generated fields of 100 bytes on the register identification key, which gives approximately 1 kb per recording. These same data sets have been used in the works (Abramova et al., 2014; Ahamed, 2016; Gandini et al., 2014; Kumar & Mary, 2017; Lungu & Tudorica, 2013; Martins et al., 2015; Park, 2016; Singh, 2019; Swaminathan & Elmasri, 2016; Tang & Fan, 2016; Matallah et al., 2017a). The authors also present the execution time of the workloads obtained during read, write and update operations. All workloads execute 1 000 operations and measure the average execution time, which means that there were 1 000 test queries each time querying the database. Subsequently, an overall assessment of the performance of all workloads (A-B-C-D-E-F) will be performed. Before concluding with a summary of the results obtained, we will evaluate the databases, on both aspects read and write separately.

Data Loading

Figure 1 shows the loading times of 600,000 records for each of the six databases.

During the loading operation, document-oriented databases (Couchbase and MongoDB) performed better on average. The best time was obtained by Couchbase with only 1 minute

Figure 1. Loading time (Minutes: Seconds)



and 4 seconds against 2 minutes and 7 seconds by MongoDB (Couchbase was twice as fast as MongoDB). HBase also proved its effectiveness in the initial loading operations with only 1 minute and 6 seconds, unlike Cassandra who was far behind with 5 minutes and 6 seconds. Although Redis was significantly faster than MongoDB, the key-value databases (Redis and OrientDB) showed average performance below those of the documents, since OrientDB was twice as slow as Redis and slower than the basics document-oriented (Couchbase and MongoDB). For the 600 000 records, the two column-oriented databases performed less well, on average, than the others. This is confirmed in several works (Abramova et al., 2014; Matallah et al., 2017a; Park, 2016; Singh, 2019; Tang & Fan, 2016).

Workload A (50% Read - 50% Update)

Figure 2 shows the results obtained after executing workload A.

In this test, the good performances are presented first by the document-oriented category where MongoDB was the fastest followed by Couchbase. The column-oriented category is in second position, Cassandra with 20 seconds and HBase with 28 seconds. The key-value databases are the least efficient relative to the previous ones. Nevertheless, to favor a certain solution for the Read or Update operations or both at the same time, it is also necessary to refer to the other results and in particular those of the workload C (100% Read) and workload H (100% Update). This is confirmed in several works (Abramova et al., 2013; Matallah et al., 2017a; Singh, 2019; Tang & Fan, 2016).

Workload B (95% Read - 5% Update)

The results of Figure 3 prove once again that document-oriented models are used for loads composed mainly of read operations. On the other hand, those of the key-value proved their performance compared to the column-oriented models. Recall also that Couchbase has imposed itself another time on others with a duration of 8 seconds. This is confirmed in these studies (Abramova et al., 2014; Park, 2016; Tang & Fan, 2016).

Figure 2. Executing time for workload A (Seconds)

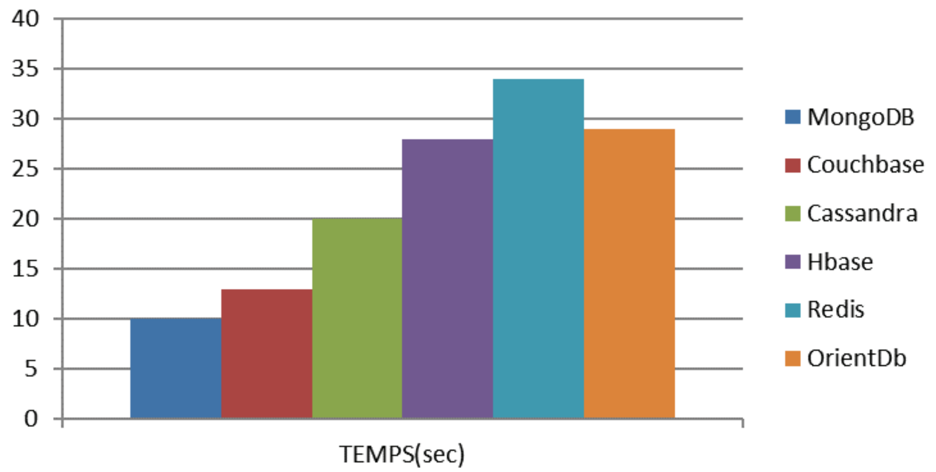
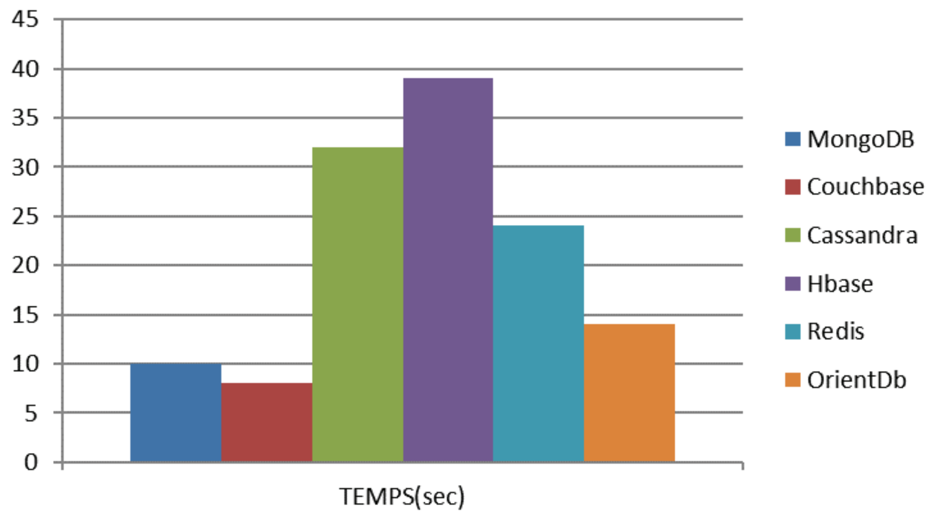


Figure 3. Executing time for workload B (Seconds)



Workload C (100% Read)

For purely read operations, the results obtained confirm the previous ranking during the execution of the load B. The first position for the document-oriented, the second position for the key-value and the third position for the column-oriented. This is confirmed in this study (Tang & Fan, 2016) (Figure 4).

Workload D (50% Read - 50% Read-Modify-Write)

Figure 5 shows the results obtained after executing workload D with 50% read operations in the first part. For the other 50%: the records are read first, modified, and the updates are written in the database permanently. We can reconfirm again the underperformance of the column-oriented systems namely HBase and Cassandra because of their constraints in reading operations compared to updating

Figure 4. Executing time for workload C (Seconds)

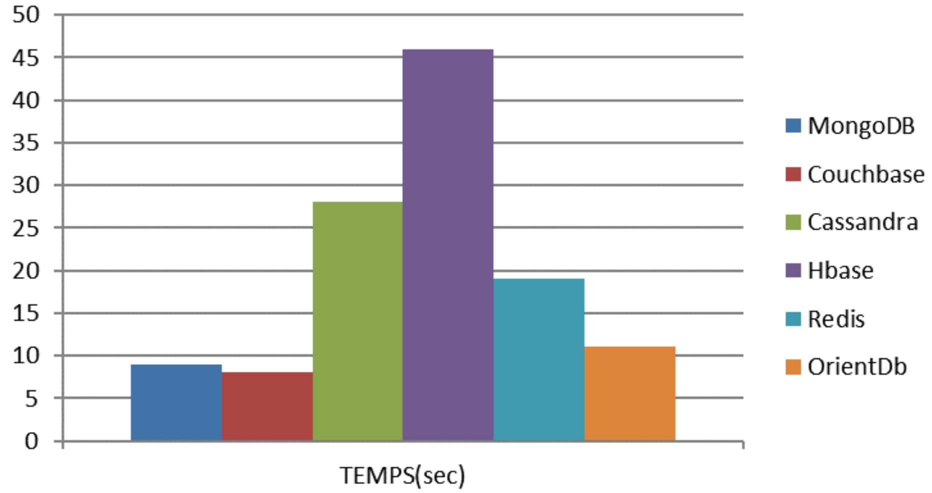
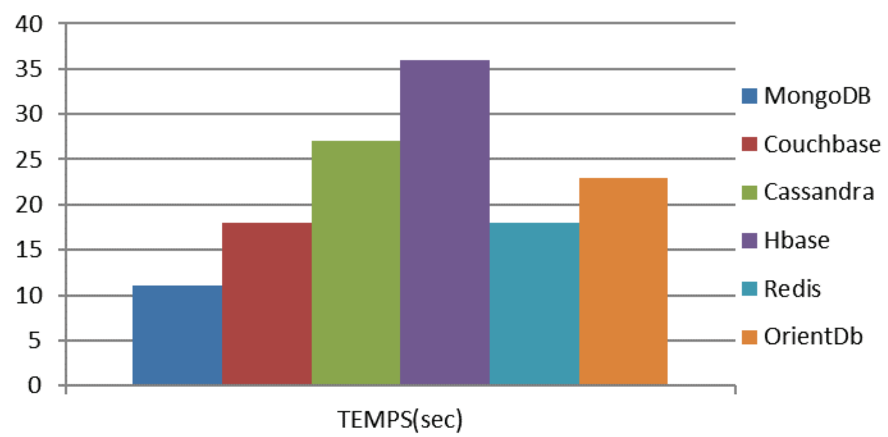


Figure 5. Executing time for workload D (Seconds)

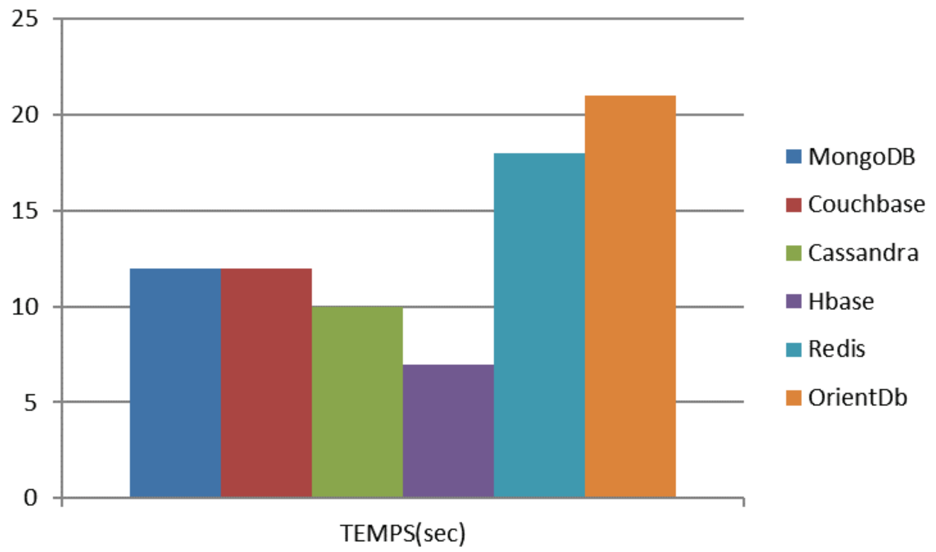


operations. In addition, Redis and Couchbase have obtained good results (18 sec each). On the other hand, the best performance is that of MongoDB which is proving its effectiveness with Couchbase for reading operations. This is confirmed in these studies (Abramova et al., 2014; Tang & Fan, 2016).

Workload E (5% Read - 95% Update)

The results of the workload E reveal that for a workload composed mainly of update, the column-oriented databases are more efficient compared to the other architectures. On the other hand, those of the key-values were largely below. Also note that the two document-oriented systems Couchbase and MongoDB obtained the same results (12 seconds each). This is confirmed in several works (Abramova et al., 2013; Kumar & Mary, 2017; Matallah et al., 2017a; Singh, 2019; Tang & Fan, 2016) (Figure 6).

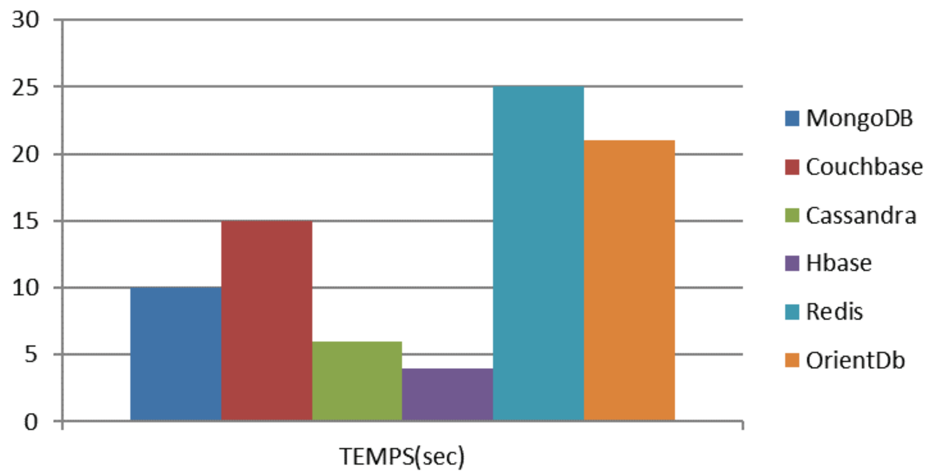
Figure 6. Executing time for workload E (Seconds)



Workload F (100% Update)

Figure 7 shows the results obtained after the execution of workload F (update operations only). For this purely updated workload, the HBase and Cassandra column-oriented databases clearly reconfirm their performance achieved when executing workload E compared to all other NoSQL systems. The Key-value systems are disappointing again for this category of queries. This is confirmed in several works (Abramova et al., 2013; Matallah et al., 2017a; Singh, 2019; Tang & Fan, 2016).

Figure 7. Executing time for workload F (Seconds)



Overall Executing Time of All Workloads

From a global point of view, document-oriented and column-oriented models are much more efficient than key-value models. The best execution time is presented by Couchbase with 2 minutes only against 3 minutes obtained by Cassandra who is in second position. MongoDB and HBase are positioned just after with 5 minutes. The two key-value systems Redis and OrientDB are placed at the last positions with 9 and 18 minutes respectively (Figure 8).

Overall Evaluation for Reading and Updating Operations

The following two Figures cover respectively the read operations and update operations. The x-axis of Figure 9 contains eight workloads ranked in ascending order of reading rates (from 0% to 100%) as

Figure 8. Overall executing time (A + B + C + D + E + F) (Seconds)

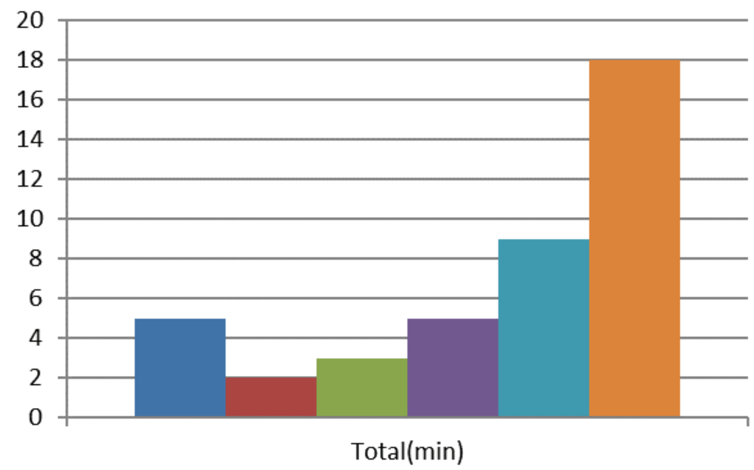
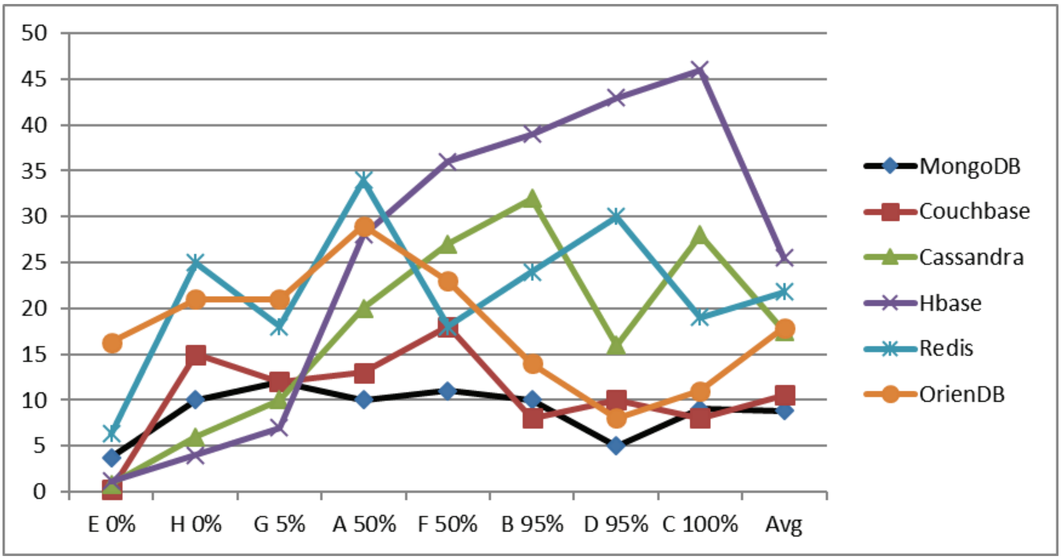


Figure 9. Overall evaluation for read operations (Seconds)

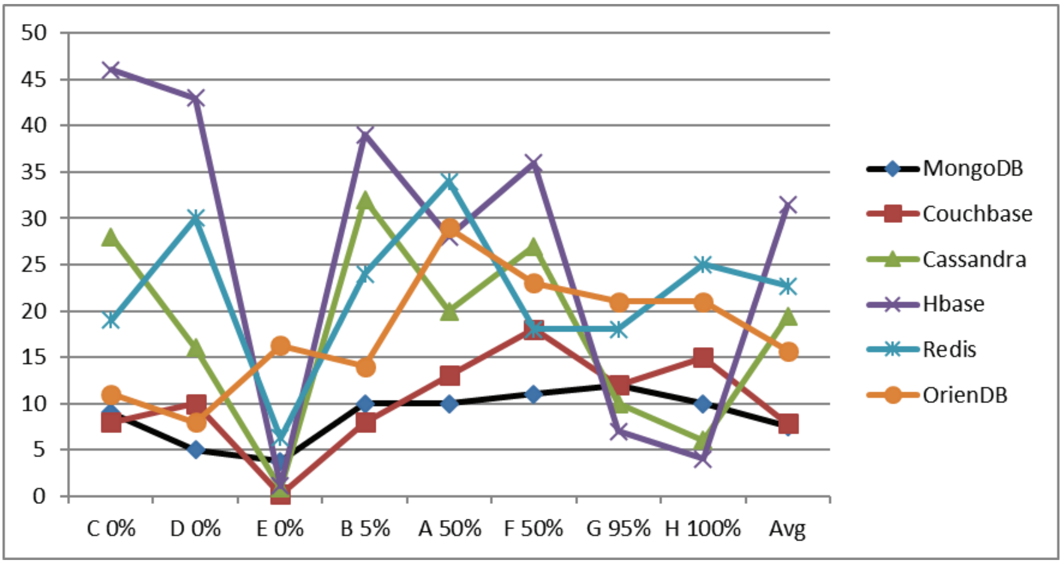


well as the average of the execution times. The x-axis of Figure 10 contains eight workloads ranked in ascending order of update rates (from 0% to 100%) as well as the average. The executing times appear on the y-axis.

In general, Figure 9 shows that the increase in read operations in workloads favors document-oriented models over other models.

In contrast to Figure 9, Figure 10 shows globally that the increase in the rate of update operations in workloads is in favor of column-oriented models compared to other models, which can be seen in the latest H workload, where HBase and Cassandra were the best performers.

Figure 10. Overall evaluation for write operations (Seconds)



After executing the 1000 operations on the 600 thousand records inserted, it can be notice that on average, a slight advantage of document-oriented systems. However, the results may change if the number of operations to be performed (1000) will increase.

A synthesis and analysis of the results obtained is described in the following.

Synthesis and Analysis of Results

Through the performance evaluations of the six NoSQL databases studied with different architectures (document-oriented, column-oriented, and key-value), several lessons can be learned:

1. For the loading phase of the 600 000 records, the best result obtained is that of Couchbase. This can be justified by the fact that the latter offers high performances because of its innovations that allow it to be the first to exploit an integrated RAM cache layer;
2. HBase has proven its performance in insertions of new records into the database. The good performance of HBase in this first test is justified by the fact that HBase has a specific memory management that optimizes its use to perform initial loading operations;
3. Couchbase was generally more efficient in reading operations compared to others. This is confirmed in this study;

4. MongoDB was also good at reading operations. This is mainly due to the memory loaded MongoDB register mapping which improves this reading performance. This is confirmed by (Matallah et al., 2017a) despite the differences between the hardware and software environments used in the previous study and this study;
5. For transactions consisting of huge reading operations, HBase was less efficient than the others. This is justified by the fact that HBase provides a fairly high level of consistency compared to the others, so to perform a read operation, HBase compares all copies and returns the most recent copy;
6. For purely read operations, document-oriented architectures namely Couchbase and MongoDB are highly recommended;
7. HBase and Cassandra proved their performance once again in the write operations, this time in the update of already existing records in the database. This is explained by the fact that HBase and Cassandra use a log and caches to save and keep track of all changes made. The use of these mechanisms considerably reduces the number of I / O operations performed on the disk, characterized by a low speed compared to the RAM. Data replication is done automatically, which speeds up the updating process, this is also confirmed by (Abramova et al., 2014; Matallah et al., 2017a; Tang & Fan, 2016)). Thus, these solutions are optimized specifically for performing update operations;
8. The update process in MongoDB is slowed down proportionally to the number of updates applied. This database uses locking mechanisms that increase the update execution time;
9. The performance of OrientDB and Redis has also degraded with the increasing number of update operations;
10. For heavy update operations, it is very interesting to adopt column-oriented architectures;
11. For the key-value solutions studied, much work remains to be done by designers to improve their performance.

In summary, the studied solutions can be classified in two categories, those optimized for readings and those optimized for updates. Thus, MongoDB, Redis, Couchbase, and OrientDB are optimized to perform read operations, while Cassandra and HBase perform better in update operations.

Also, it should be noted that some databases were less efficient because they require more system resources than the environment used in the evaluation. For the loading operations, Couchbase and HBase have were efficient because they have a specific RAM processing. MongoDB is good at reading operations, due to the memory loaded MongoDB register mapping, unlike HBase which requires excessive consistency checks. HBase and Cassandra proved their performance in the update operations because they use a log and caches to save and keep track of all changes made, Unlike MongoDB which uses locking mechanisms that increase the update execution time.

CONCLUSION

In response to the new requirements imposed by highly distributed environments, several NoSQL systems have emerged to replace relational systems. A performance evaluation is necessary for the NoSQL users to know the best suitable database for their application.

The study developed and presented in this paper, consists of comparing the performances of six NoSQL models of different architectures, widely used, namely MongoDB, Couchbase, Cassandra, HBase, Redis, and OrientDB. The aim is to evaluate the performance of these databases by inserting first 600 000 records, then by executing a set of tests as workload composed of 1 000 operations each of different natures: read, insert or update.

The tool used to arbitrate the six systems is Yahoo! Cloud Serving Benchmark, which is highly recommended for this type of study in the area of NoSQL databases.

After analyzing the experimental results obtained, it can affirm that there are very powerful databases for particular workloads, unlike others that were better in other workloads. There are solutions that are optimized for readings, and others for updates, as has been stated by (Abramova et al., 2014; Matallah et al., 2017a; Singh, 2019; Tang & Fan, 2016).

In general, NoSQL databases guarantee good performance for simple operations over possibly huge datasets. However, the authors can confirm that the criteria for choosing the appropriate solution depend, application requirements, the nature of operations performed on the data manipulated and the context of use of these last.

This work is part of a list of similar works evaluating the performance of the various NoSQL databases. In the various research works, several solutions were compared either in a single machine or in several virtual or physical machines by varying each time the number of nodes or the number of cores (Abramova et al., 2015; Ahamed, 2016; Gandini et al., 2014; Kumar & Mary, 2017; Park, 2016). The number of records inserted, the types of workloads as well as the number of operations executed have changed from one study to another (Abramova et al., 2013; Abramova et al., 2014; Matallah et al., 2017a; Singh, 2019; Tang & Fan, 2016). Several comparison metrics were used to distinguish between the different databases such as execution time, latency, throughput or scalability (Abramova et al., 2015; Ahamed, 2016; Gandini et al., 2014; Kumar & Mary, 2017; Singh, 2019; Swaminathan & Elmasri, 2016; Tang & Fan, 2016).

In this study, the various tests were carried out in a single physical machine. The researchers opted for six NoSQL solutions, they inserted 600,000 records in each, and chose six workloads of 1000 operations. They only targeted execution time as a metric to compare performance.

As perspectives, the next evaluation studies will consist to skip to higher scales with the increase in the number of operations performed to reach Workloads exceeding 200 000 operations and the number of records inserted to have a large test database hosting millions of records. Another perspective in the near future is to do experiments on real data from transport, automotive, and health applications where the data generated is qualified as Big Data generated from connected objects (Al-Qerem, Alauthman, Almomani, & Gupta, 2019; Bharathi & Selvarani, 2019; Hussain & Beg, 2019; Kaur, Singh, & Aggarwal, 2019). The researchers also want to support, in a future study, attacks in NoSQL servers, such as the DDoS attack (Alieyan et al., 2019; Alomari, Manickam, Gupta, Karuppayah, & Alfaris, 2012; Dahiya & Gupta, 2019). It should be recalled that this comparative study is based on a single-node architecture; tests and performance evaluations in distributed and parallel environments will be another challenge to raise. The authors also plan to develop further benchmarking studies comparing NoSQL systems with SQL solutions used in cloud computing environments as well as with new NewSQL systems, which may be the subject of further comparative studies. Finally, note that this evaluation was based on the executing time. The aim is to provide some answers to choose the adequate NoSQL solution compared to the nature of frequently executed operations (update / read / insert / scan). Comparisons with other metrics such as data consistency, availability, latency, reliability, and scalability are planned in future work.

REFERENCES

Abramova, V., & Bernardino, J. (2013). NoSQL databases: MongoDB vs cassandra. In *Proceedings of the international C* conference on computer science and software engineering* (pp. 14-22). ACM.

Abramova, V., Bernardino, J., & Furtado, P. (2014). Experimental evaluation of NoSQL databases. *International Journal of Database Management Systems*, *6*(3), 1. doi:10.5121/ijdms.2014.6301

Abramova, V., Bernardino, J., & Furtado, P. (2015). SQL or NoSQL? Performance and scalability evaluation. *International Journal of Business Process Integration and Management*, *7*(4), 314–321. doi:10.1504/IJBPIM.2015.073655

Ahamed, M. A. U. R. (2016). *Benchmarking Top NoSQL Databases* (Doctoral dissertation). Yüksek Lisans Tezi, Institute of Computer Science TU Clausthal.

Al-Qerem, A., Alauthman, M., Almomani, A., & Gupta, B. B. (2019). IoT transaction processing through cooperative concurrency control on fog-cloud computing environment. *Soft Computing*, 1–17.

Alieyan, K., Almomani, A., Anbar, M., Alauthman, M., Abdullah, R., & Gupta, B. B. (2019). DNS rule-based schema to botnet detection. *Enterprise Information Systems*, 1–20. doi:10.1080/17517575.2019.1644673

Alomari, E., Manickam, S., Gupta, B. B., Karuppayah, S., & Alfaris, R. (2012). *Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art*. arXiv preprint arXiv:1208.0403

Aslett, M. (2011). *How will the database incumbents respond to NoSQL and NewSQL*. Academic Press.

Bharathi, R., & Selvarani, R. (2019). Software Reliability Assessment of Safety Critical System Using Computational Intelligence. *International Journal of Software Science and Computational Intelligence*, *11*(3), 1–25. doi:10.4018/IJSSCI.2019070101

Brown, M. C. (2012). *Getting Started with Couchbase Server: Extreme Scalability at Your Fingertips*. O'Reilly Media, Inc.

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *SIGMOD Record*, *39*(4), 12–27. doi:10.1145/1978915.1978919

Chodorow, K. (2013). *MongoDB: the definitive guide: powerful and scalable data storage*. O'Reilly Media, Inc.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, *13*(6), 377–387. doi:10.1145/362384.362685

Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., & Sears, R. (2010). Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing* (pp. 143-154). ACM.

Dahiya, A., & Gupta, B. B. (2019). A PBNM and economic incentive-based defensive mechanism against DDoS attacks. *Enterprise Information Systems*, 1–21. doi:10.1080/17517575.2019.1700553

Gandini, A., Gribaudo, M., Knottenbelt, W. J., Osman, R., & Piazzolla, P. (2014). Performance evaluation of NoSQL databases. In *European Workshop on Performance Engineering* (pp. 16-29). Springer.

George, L. (2011). *HBase: the definitive guide: random access to your planet-size data*. O'Reilly Media, Inc.

Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *Acm Sigact News*, *33*(2), 51–59. doi:10.1145/564585.564601

Grolinger, K., Higashino, W. A., Tiwari, A., & Capretz, M. A. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. *Journal of Cloud Computing: Advances, Systems and Applications, 2*(1), 22.

Gupta, S. (2018). Performance Evaluation of Unstructured PBRA for Bigdata with Cassandra and MongoDB in Cloud. *International Journal of Cloud Applications and Computing*, *8*(3), 48–59. doi:10.4018/IJCAC.2018070104

Gupta, S., & Gugulothu, N. (2018). Secure NoSQL for the social networking and e-commerce based bigdata applications deployed in cloud. *International Journal of Cloud Applications and Computing*, *8*(2), 113–129. doi:10.4018/IJCAC.2018040106

Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database. In *2011 6th international conference on pervasive computing and applications* (pp. 363-366). IEEE.

Hecht, R., & Jablonski, S. (2011). NoSQL evaluation: A use case oriented survey. In *2011 International Conference on Cloud and Service Computing* (pp. 336-341). IEEE. doi:10.1109/CSC.2011.6138544

Hussain, M. M., & Beg, M. S. (2019). Using Vehicles as Fog Infrastructures for Transportation Cyber-Physical Systems (T-CPS): Fog Computing for Vehicular Networks. *International Journal of Software Science and Computational Intelligence*, *11*(1), 47–69. doi:10.4018/IJSSCI.2019010104

Indrawan-Santiago, M. (2012). Database research: Are we at a crossroad? Reflection on NoSQL. In *2012 15th International Conference on Network-Based Information Systems* (pp. 45-51). IEEE.

Inmon, W. H. (2005). *Building the data warehouse*. John Wiley & Sons.

Jatana, N., Puri, S., Ahuja, M., Kathuria, I., & Gosain, D. (2012). A survey and comparison of relational and non-relational database. *International Journal of Engineering Research & Technology (Ahmedabad)*, *1*(6), 1–5.

Jehangiri, A. I., Yahyapour, R., Yaqub, E., & Wieder, P. (2018). Distributed predictive performance anomaly detection for virtualised platforms. *International Journal of High Performance Computing and Networking*, *11*(4), 279–290. doi:10.1504/IJHPCN.2018.093229

Kaur, M., Singh, S., & Aggarwal, N. (2019). Study of NoSQL Databases for Storing and Analyzing Traffic Data. In *Proceedings of 2nd International Conference on Communication, Computing and Networking* (pp. 577-587). Springer. doi:10.1007/978-981-13-1217-5_56

Kumar, R., & Mary, R. (2017). Comparative Performance Analysis of various NoSQL Databases: MongoDB, Cassandra and HBase on Yahoo Cloud Server. *Imperial Journal of Interdisciplinary Research, 3*(4).

Lakshman, A., & Malik, P. (2010). Cassandra: A decentralized structured storage system. *Operating Systems Review*, *44*(2), 35–40. doi:10.1145/1773912.1773922

Lattanzi, S., Moseley, B., Suri, S., & Vassilvitskii, S. (2011). Filtering: a method for solving graph problems in mapreduce. In *Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures* (pp. 85-94). doi:10.1145/1989493.1989505

Liu, T., Gao, S., Chu, X., & Lu, C. (2017). Storing and querying AIS data in HBase. In *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)* (pp. 88-92). IEEE. doi:10.1109/ACIRS.2017.7986071

Lungu, I., & Tudorica, B. G. (2013). The development of a benchmark tool for nosql databases. *Database Systems Journal BOARD, 13*.

Macedo, T., & Oliveira, F. (2011). *Redis Cookbook: Practical Techniques for Fast Data Manipulation*. O'Reilly Media, Inc.

Martins, G., Bezerra, P., Gomes, R., Albuquerque, F., & Costa, A. (2015). Evaluating performance degradation in NoSQL databases generated by virtualization. In *2015 Latin American Network Operations and Management Symposium (LANOMS)* (pp. 84-91). IEEE. doi:10.1109/LANOMS.2015.7332675

Matallah, H., Belalem, G., & Bouamrane, K. (2017a). Experimental comparative study of NoSQL databases: HBASE versus MongoDB by YCSB. *Computer Systems Science and Engineering*, *32*(4), 307–317.

Matallah, H., Belalem, G., & Bouamrane, K. (2017b). Towards a new model of storage and access to data in big data and cloud computing. *International Journal of Ambient Computing and Intelligence*, *8*(4), 31–44. doi:10.4018/IJACI.2017100103

McCreary, D., & Kelly, A. (2014). *Making sense of NoSQL*. Manning.

Membrey, P., Plugge, E., & Hawkins, D. (2011). *The definitive guide to MongoDB: the noSQL database for cloud and desktop computing*. Apress.

Ostrovsky, D., Rodenski, Y., & Haji, M. (2015). *Pro couchbase server*. Apress. doi:10.1007/978-1-4842-1185-4

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2017). NoSQL databases for big data. *International Journal of Big Data Intelligence*, *4*(3), 171–185. doi:10.1504/IJBDI.2017.085537

Park, H. J. (2016). A Study about Performance Evaluation of Various NoSQL Databases. *The Journal of Korea Institute of Information, Electronics, and Communication Technology*, *9*(3), 298–305. doi:10.17661/jkiiect.2016.9.3.298

Piekos, J. (2015). *SQL vs, NoSQL vs. NewSQL: finding the right solution*. Academic Press.

Rith, J., Lehmayr, P. S., & Meyer-Wegener, K. (2014). Speaking in tongues: SQL access to NoSQL systems. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (pp. 855-857). doi:10.1145/2554850.2555099

Singh, S. (2019). *A comparative study of NoSQL databases: MongoDB and HBase*. Technical Report. DOI: .10.13140/RG.2.2.14017.61286

Solid, I. T. (2019). *DB-Engines ranking. Solid IT. DB-Engines Ranking*. http://db-engines.com/en /ranking

Swaminathan, S. N., & Elmasri, R. (2016). Quantitative analysis of scalable NoSQL databases. In *2016 IEEE International Congress on Big Data (BigData Congress)* (pp. 323-326). IEEE. doi:10.1109/BigDataCongress.2016.49

Tabet, K., Mokadem, R., & Laouar, M. R. (2018). Towards a New Data Replication Strategy in MongoDB Systems. In *Proceedings of the 4th ACM International Conference of Computing for Engineering and Sciences* (p. 2). doi:10.1145/3213187.3287609

Tabet, K., Mokadem, R., & Laouar, M. R. (2019). A data replication strategy for document-oriented NoSQL systems. *International Journal of Grid and Utility Computing*, *10*(1), 53–62. doi:10.1504/IJGUC.2019.097224

Tang, E., & Fan, Y. (2016). Performance comparison between five NoSQL databases. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)* (pp. 105-109). IEEE. doi:10.1109/CCBD.2016.030

Tesoriero, C. (2013). *Getting Started with OrientDB*. Packt Publishing Ltd.

Houcine Matallah graduated from Department of Computer Science, Faculty of Sciences, University of Tlemcen, Algeria, where he received his Ph.D degree in Computer Science in 2018. He supervised 40 master and co-supervised 5 PhDs in computer science. His research interests include: Big Data, NoSQL, Cloud Computing, Distributed System and Advanced Database.

Ghalem Belalem graduated from Department of Computer Science, Faculty of Exact and Applied Sciences, University of Oran, Algeria, where he received PhD degree in computer science in 2007. His current research interests are distributed system; grid computing, cloud computing, replication, consistency, fault tolerance, resource management, economic models, energy consumption, Big data, IoT, mobile environment, images processing, Supply chain optimization, Decision support systems, High Performance Computing https://orcid.org/0000-0002-9694-7586.

Karim Bouamrane received the PhD Degree in computer science from the « Oran University » in 2006. He is full Professor of computer Science at the same university since 2012. He is the head of computer science laboratory (LIO) since November 2017 and coordinate the team "decision support and piloting system". His current research interests deal with decision support system in transportation system, production system and eLearning. He participates in several scientific committees' international/national conferences in Algeria and other countries in the same domain and collaborate in Algerian-French scientific projects. He is co-author of more than 30 scientific publications.

International Journal of Software Science and Computational Intelligence

Volume 12 • Issue 4 • October-December 2020 • ISSN: 1942-9045 • eISSN: 1942-9037

MISSION

The **International Journal of Software Science and Computational Intelligence (IJSSCI)** is a transdisciplinary, archived, and rigorously refereed journal that publishes and disseminates cutting-edge research findings and technological developments in the emerging fields of software science and computational intelligence, as well as their engineering applications. The latest developments in computer science, theoretical software engineering, cognitive science, cognitive informatics, and intelligence science, and the crystallization of accumulated knowledge by the fertilization of these areas, have led to the emergence of a transdisciplinary and convergence field known as software and intelligence sciences.

Software Science is a discipline of enquiries that studies, theoretical framework of software as instructive and behavioral information, which can be embodied and executed by generic computers in order to create expected system behaviors and machine intelligence. The discipline of software science studies the common objects in the abstract world such as software, information, data, concepts, knowledge, instructions, executable behaviors, and their processing by natural and machine intelligence. From this view, software engineering is the engineering discipline that applies software science theories and methodologies to efficiently, economically, and reliably organize and develop large-scale software systems.

Computational Intelligence is a human enquiry of both natural and machine intelligence at the reduction levels of biology, physiology, psychology, cognition, computation, logic, and philosophy from the bottom up. Intelligence science is naturally interdisciplinary in vertical and based on reductionism and is transdisciplinary in horizontal based on holism. Intelligence science studies theories and models of the brain at all levels, and the relationship between the concrete physiological brain and the abstract soft mind. Intelligence science is a new frontier with the fertilization of biology, psychology, neuroscience, cognitive science, philosophy, information science, computer science, anthropology, and linguistics.

SUBSCRIPTION INFORMATION

The International Journal of Software Science and Computational Intelligence (IJSSCI) is available in print and electronic formats and offers individual or institution-level pricing. Full subscription information can be found at www.igi-global.com/IJSSCI.

IJSSCI is also included in IGI Global's InfoSci-Journals Database which contains all of IGI Global's peer-reviewed journals and offers unlimited simultaneous access, full-text PDF and XML viewing, with no DRM. Subscriptions to the InfoSci-Journals Database are available for institutions. For more information, please visit www.igi-global.com/infosci-journals or contact E-Resources at eresources@igi-global.com.

CORRESPONDENCE AND QUESTIONS

EDITORIAL

Brij Gupta, Editor-in-Chief • IJSSCI@igi-global.com

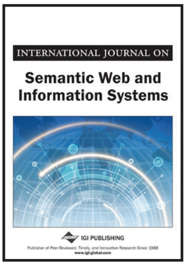
SUBSCRIBER INFO

IGI Global • Customer Service
701 East Chocolate Avenue • Hershey PA 17033-1240, USA
Telephone: 717/533-8845 x100 • **E-Mail:** cust@igi-global.com

The *International Journal of Software Science and Computational Intelligence* is indexed or listed in the following.
ACM Digital Library; Bacon's Media Directory; Cabell's Directories; DBLP; GetCited; Google Scholar; INSPEC; JournalTOCs; MediaFinder; The Standard Periodical Directory; Ulrich's Periodicals Directory

Stay Current on the Latest Emerging Research Developments

Become an IGI Global Journal Reviewer



The overall success of a refereed journal is dependent on quality and timely reviews. In this competitive age of scholarly publishing, constructive and timely feedback significantly expedites the turnaround time of manuscripts from submission to acceptance, allowing the publication and discovery of innovative research at a much more expeditious rate.

Several IGI Global journals in the area of **Computer Science and Information Technology** are currently seeking highly-qualified experts in the field to fill vacancies on their respective editorial review boards:

- Journal of Global Information Management (JGIM)
- International Journal of Web Services Research (IJWSR)
- International Journal on Semantic Web and Information Systems (IJSWIS)
- International Journal of Data Warehousing and Mining (IJDWM)
- Journal of Cases on Information Technology (JCIT)

Applications may be submitted online at:

www.igi-global.com/journals/become-a-reviewer/

Applicants must have a doctorate (or an equivalent degree), as well as publishing and reviewing experience. Reviewers are asked to write reviews in a timely, collegial, and constructive manner. All reviewers will begin their role on an ad-hoc basis for a period of one year, and upon successful completion of this term can be considered for full editorial review board status, with the potential for a subsequent promotion to Associate Editor.

Questions regarding this opportunity can be sent to:

journaleditor@igi-global.com

If you have a colleague that may be interested in this opportunity, we encourage you to share this information with them.

Free Download

InfoSci®-Dictionary App

Search

Search through 55,000+ terms.

Select

Retrieve 107,000+ definitions related to all aspects of academic research.

Share

Share the wealth of knowledge with colleagues, friends, and more.

Users can search from over 55,000 terms and more than 107,000 definitions, all supported by published research from over 50 countries and hundreds of disciplines. Providing access to cutting-edge terminology in diverse fields, the InfoSci®-Dictionary app presents immediate results on research directly relevant to your needs and interests.

Unrivaled among scholarly publishers, InfoSci®-Dictionary is continuously updated to include definitions from our latest reference book releases. The InfoSci®-Dictionary App is available on Amazon, the Google Play Store and the Apple App Store. Download it today!

IGI Global is pleased to offer the InfoSci®-Dictionary mobile application.

Available for Download Now

Available at amazon | GET IT ON Google play | Download on the App Store

For more information on IGI Global's databases, visit our InfoSci®-Databases webpage: www.igi-global.com/e-resources.
For more information about InfoSci®-Dictionary, contact IGI Global's Database team at eresources@igi-global.com.